

Novell eDirectory™

8.7.3

www.novell.com

ADMINISTRATION GUIDE

March 18, 2008



Novell®

Legal Notices

Novell, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export, or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. Please refer to www.novell.com/info/exports/ for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Copyright © 2008 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

Novell, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.novell.com/company/legal/patents/> and one or more additional patents or pending patent applications in the U.S. and in other countries.

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.

www.novell.com

Novell eDirectory 8.7.3 Administration Guide
[March 18, 2008](#)

Online Documentation: To access the online documentation for this and other Novell products, and to get updates, see www.novell.com/documentation.

Novell Trademarks

ConsoleOne is a registered trademark of Novell, Inc. in the United States and other countries.

DirXML is a registered trademark of Novell, Inc. in the United States and other countries.

eDirectory is a trademark of Novell, Inc.

Internetwork Packet Exchange and IPX are trademarks of Novell, Inc.

NetWare is a registered trademark of Novell, Inc. in the United States and other countries.

NetWare Core Protocol and NCP are trademarks of Novell, Inc.

NetWare Loadable Module and NLM are trademarks of Novell, Inc.

NetWare SMS is a trademark of Novell, Inc.

NMAS is a trademark of Novell, Inc.

Novell is a registered trademark of Novell, Inc. in the United States and other countries.

Novell Certificate Server is a trademark of Novell, Inc.

Novell Client is a trademark of Novell, Inc.

Novell Directory Services and NDS are registered trademarks of Novell, Inc. in the United States and other countries.

Novell SecretStore is a registered trademark of Novell, Inc. in the United States and other countries.

Transaction Tracking System and TTS are trademarks of Novell, Inc.

ZENworks is a registered trademark of Novell, Inc. in the United States and other countries.

Third-Party Materials

All third-party trademarks are the property of their respective owners.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org>).

Contents

About This Guide	15
1 Understanding Novell eDirectory	17
What's New	18
Novell eDirectory	19
Ease of Management through Novell iManager	19
Powerful Tree Structure	20
Web-Based Management Utility	21
Single Login and Authentication	22
Object Classes and Properties	22
List of Objects	23
Container Object Classes	25
Leaf Object Classes	28
Context and Naming	37
Distinguished Name	38
Typeful Name	38
Name Resolution	39
Current Workstation Context	39
Leading Period	39
Relative Naming	39
Trailing Periods	40
Context and Naming on UNIX	40
Schema	40
Schema Management	41
Schema Classes, Attributes, and Syntaxes	41
Understanding Mandatory and Optional Attributes	45
Sample Schema	45
Designing the Schema	46
Partitions	46
Partitions	47
Distributing Replicas for Performance	48
Partitions and WAN Links	48
Replicas	49
Replica Types	50
Filtered Replicas	53
NetWare Bindery Emulation	54
Server Synchronization in the Replica Ring	54
Access to Resources	54
eDirectory Rights	55
Trustee Assignments and Targets	55
eDirectory Rights Concepts	56
Default Rights for a New Server	60
Delegated Administration	60
Administering Rights	61
2 Designing Your Novell eDirectory Network	67

eDirectory Design Basics	67
Network Layout	67
Organizational Structure	68
Preparing for eDirectory Design	68
Designing the eDirectory Tree.	68
Creating a Naming Standards Document	68
Designing the Upper Layers of the Tree	71
Designing the Lower Layers of the Tree	73
Guidelines for Partitioning Your Tree	74
Determining Partitions for the Upper Layers of the Tree.	74
Determining Partitions for the Lower Layers of the Tree.	75
Determining Partition Size	75
Considering Network Variables	75
Guidelines for Replicating Your Tree	75
Workgroup Needs	76
Fault Tolerance	76
Determining the Number of Replicas	77
Replicating the Tree Partition	77
Replicating for Administration	77
Meeting Bindery Services Needs for NetWare.	77
Managing WAN Traffic.	78
Planning the User Environment	78
Reviewing Users' Needs.	78
Creating Accessibility Guidelines	78
Designing eDirectory for e-Business	79
Understanding the Novell Certificate Server	80
Rights Required to Perform Tasks on Novell Certificate Server	80
Ensuring Secure eDirectory Operations on Linux, Solaris, AIX, and HP-UX Systems	81
Synchronizing Network Time	84
Synchronizing Time on NetWare Servers	84
Synchronizing Time on Windows Servers	85
Synchronizing Time on Linux, Solaris, AIX, or HP-UX Systems.	85
Verifying Time Synchronization	85
3 Managing Objects	87
General Object Tasks	87
Browsing the eDirectory Tree	88
Creating an Object.	90
Modifying an Object's Properties	90
Copying Objects	90
Moving Objects	91
Deleting Objects	91
Renaming Objects	91
Managing User Accounts	92
Creating and Modifying User Accounts	92
Setting Up Optional Account Features.	93
Setting Up Login Scripts	95
Login Time Restrictions for Remote Users	96
Deleting User Accounts	97
Configuring Role-Based Services	97
Defining RBS Roles	99
Defining Custom RBS Tasks.	101
4 Managing the Schema	103
Extending the Schema	104
Creating a Class	104
6 Novell eDirectory 8.7.3 Administration Guide	

Deleting a Class	104
Creating an Attribute	105
Adding an Optional Attribute to a Class	105
Deleting an Attribute	105
Creating an Auxiliary Class	106
Extending an Object with the Properties of an Auxiliary Class	106
Modifying an Object's Auxiliary Properties	106
Deleting Auxiliary Properties from an Object	107
Viewing the Schema	107
Viewing Class Information	107
Viewing Attribute Information	107
Manually Extending the Schema	108
Extending the Schema on NetWare	108
Extending the Schema on Windows	108
Extending the Schema on Linux, Solaris, AIX, or HP-UX Systems	108
Schema Flags Added in eDirectory 8.7.	110
Using the eMBox Client to Perform Schema Operations	111
Using the DSSchema eMTool	111
DSSchema eMTool Options	112
5 Managing Partitions and Replicas	113
Creating a Partition	114
Merging a Partition	115
Moving Partitions	116
Cancelling Create or Merge Partition Operations.	117
Administering Replicas	117
Adding a Replica	117
Deleting a Replica.	118
Changing a Replica Type	119
Setting Up and Managing Filtered Replicas	120
Using the Filtered Replica Wizard.	120
Defining a Partition Scope	121
Setting Up a Server Filter	122
Viewing Partitions and Replicas	123
Viewing the Partitions on a Server	123
Viewing a Partition's Replicas.	123
Viewing Information about a Partition.	123
Viewing Partition Hierarchy	124
Viewing Information about a Replica	124
6 Novell eDirectory Management Utilities	125
Novell Import Conversion Export Utility.	125
Using the Novell iManager Import Convert Export Wizard	126
Using the Command Line Interface	129
Conversion Rules	144
LDAP Bulk Update/Replication Protocol	152
Migrating the Schema between LDAP Directories	153
Improving the Speed of LDIF Imports.	154
Index Manager	155
Creating an Index	156
Deleting an Index	156
Taking an Index Offline	157
Managing Indexes on Other Servers	157
Using the Novell Import Conversion Export Utility to Manage Indexes	158
Predicate Data	159
Managing Predicate Data	160
eDirectory Service Manager	160

Using the eMBox Client Service Manager eMTool	160
Using the Service Manager Plug-In to Novell iManager	161
7 Using Novell iMonitor 2.1	163
System Requirements	164
Platforms	164
eDirectory Versions That Can Be Monitored	165
Accessing iMonitor	165
iMonitor Architecture	165
Anatomy of an iMonitor Page	166
Modes of Operation	166
iMonitor Features Available on Every Page	168
NetWare Remote Manager Integration	168
Configuration Files	168
iMonitor Features	171
Viewing eDirectory Server Health	172
Viewing Partition Synchronization Status	172
Viewing Server Connection Information	172
Viewing Known Servers	173
Viewing Replica Information	174
Controlling and Configuring the DS Agent	174
Configuring Trace Settings	175
Viewing Process Status Information	176
Viewing Agent Activity	176
Viewing Traffic Patterns	177
Viewing Background Processes	177
Viewing eDirectory Server Errors	177
Viewing DSRepair Information	177
Viewing Agent Health Information	178
Browsing Objects in Your Tree	178
Viewing Entries for Synchronization or Purging	179
Viewing DirXML Details	179
Viewing the Synchronization Status of a Replica	179
Configuring and Viewing Reports	179
Viewing Schema, Class, and Attribute Definitions	181
Searching for Objects	181
Using the Stream Viewer	182
Clone DIB Set	182
Ensuring Secure iMonitor Operations	187
8 Merging Novell eDirectory Trees	189
Merging eDirectory Trees	189
Prerequisites	190
Target Tree Requirements	190
Schema Requirements	190
Merging the Source into the Target Tree	191
Partition Changes	191
Preparing the Source and Target Trees	192
Synchronizing Time before the Merge	192
Merging Two Trees	193
Post-Merge Tasks	194
Grafting a Single Server Tree	195
Understanding Context Name Changes	196
Preparing the Source and Target Trees	197
Grafting the Source and Target Tree	199
Renaming a Tree	199

Using the eMBox Client to Merge Trees	200
Using the DSMerge eMTool	200
DSMerge eMTool Options	201
9 Repairing the Novell eDirectory Database	203
Performing Basic Repair Operations	204
Performing an Unattended Full Repair	205
Performing a Local Database Repair	206
Checking External References	207
Repairing a Single Object	207
Deleting Unknown Leaf Objects	208
Viewing and Configuring the Repair Log File	208
Opening the Log File	208
Setting Log File Options	209
Performing a Repair in Novell iMonitor	209
Repairing Replicas	210
Repairing All Replicas	210
Repairing Selected Replicas	210
Repairing Time Stamps	211
Designating This Server As the New Master Replica	212
Destroying the Selected Replica	212
Repairing Replica Rings	212
Repairing All Replica Rings	213
Repairing the Selected Replica Ring	213
Sending All Objects to Every Server in the Ring	213
Receiving All Objects from the Master to the Selected Replica	214
Removing This Server from the Replica Ring	214
Maintaining the Schema	214
Requesting Schema from the Tree	215
Resetting the Local Schema	215
Performing a Post-NetWare 5 Schema Update	215
Performing Optional Schema Enhancements	216
Importing Remote Schema	216
Declaring a New Schema Epoch	217
Repairing Server Network Addresses	217
Repairing All Network Addresses	218
Repairing a Server's Network Addresses	218
Performing Synchronization Operations	219
Synchronizing the Selected Replica on This Server	219
Reporting the Synchronization Status on This Server	219
Reporting the Synchronization Status on All Servers	220
Performing a Time Synchronization	220
Scheduling an Immediate Synchronization	221
Advanced DSRepair Options	221
Running DSRepair on the eDirectory Server	221
DSRepair Command Line Options	222
Using Advanced DSRepair Switches	224
Using the eMBox Client to Repair a Database	225
Using the DSRepair eMTool	225
DSRepair eMTool Options	226
10 WAN Traffic Manager	229
Understanding WAN Traffic Manager	229

LAN Area Objects	231
WAN Traffic Policies	232
Limiting WAN Traffic	235
Assigning Cost Factors	236
WAN Traffic Manager Policy Groups	238
1-3am.wmg	238
7am-6pm.wmg	238
Costlt20.wmg	238
lpx.wmg	238
Ndsttyps.wmg	239
Onospoof.wmg	250
Opnspoof.wmg	250
Samearea.wmg	250
Tcpiw.wmg	251
Timecost.wmg	251
WAN Policy Structure	251
Declaration Section	252
Selector Section	254
Provider Section	254
Construction Used within Policy Sections	254
11 Understanding LDAP Services for Novell eDirectory	259
Key Terms for LDAP Services	260
Clients and Servers	260
Objects	260
Referrals	261
Understanding How LDAP Works with eDirectory	262
Connecting to eDirectory from LDAP	262
Class and Attribute Mappings	265
Enabling Nonstandard Schema Output	268
Syntax Differences	269
Supported Novell LDAP Controls and Extensions	270
Using LDAP Tools on Linux, Solaris, AIX, or HP-UX	271
LDAP Tools	271
Extensible Match Search Filter	280
12 Configuring LDAP Services for Novell eDirectory	283
Loading and Unloading LDAP Services for eDirectory	283
Verifying That the LDAP Server Is Loaded	284
Verifying That the LDAP Server Is Running	285
Scenarios	285
Verifying That The LDAP Server Is Running	286
Verifying That A Device Is Listening	287
Configuring LDAP Objects	288
Configuring LDAP Server and LDAP Group Objects on Linux, Solaris, AIX, or HP-UX Systems	289
Refreshing the LDAP Server	292
Authentication and Security	293
Requiring TLS for Simple Binds with Passwords	294
Starting and Stopping TLS	294
Configuring the Server for TLS	295
Configuring the Client for TLS	296
Exporting the Trusted Root	296
Authenticating with a Client Certificate	297
Using Certificate Authorities from Third-Party Providers	297
Creating and Using LDAP Proxy Users	298
Using SASL	299

Using the LDAP Server to Search the Directory	300
Setting Search Limits	300
Using Referrals	301
Searching Filtered Replicas	306
Configuring for Superior Referrals	307
Scenario: Superior Referrals in a Federated Tree	307
Creating a Nonauthoritative Area	308
Specifying Reference Data	309
Updating Reference Information through LDAP	310
Affected Operations	310
Discovering Support for Superior References	310
Persistent Search: Configuring for eDirectory Events	311
Managing Persistent Searches	311
Controlling Use of the Monitor Events Extended Operation	312
Getting Information about the LDAP Server	313
13 SNMP Support for Novell eDirectory	317
Definitions and Terminology for SNMP	317
Understanding SNMP Services	318
Network Management Station	318
Managed Devices	319
Subagent	319
Master Agent	319
Management Information Base	319
SNMP Network Management Protocol	319
eDirectory and SNMP	320
Benefits of SNMP Instrumentation on eDirectory	320
Understanding How SNMP Works with eDirectory	320
Installing and Configuring SNMP Services for eDirectory	323
Loading and Unloading the SNMP Server Module	323
Subagent Configuration	324
Setting Up SNMP Services for eDirectory	326
Monitoring eDirectory Using SNMP	336
Traps	336
Configuring Traps	350
Statistics	359
Troubleshooting	363
14 Backing Up and Restoring Novell eDirectory	365
Checklist for Backing Up eDirectory	366
Understanding Backup and Restore Services	368
About the eDirectory Backup eMTool	369
What's Different about Backup and Restore in eDirectory 8.7.3?	370
Overview of How the Backup eMTool Does a Restore	371
Format of the Backup File Header	372
Format of the Backup Log File	376
Using DSMASTER Servers as Part of Disaster Recovery Planning	377
Transitive Vectors and the Restore Verification Process	378
Restore Verification Is Backward Compatible Only with eDirectory 8.5 or Later	378
Preserving Rights When Restoring File System Data on NetWare	379
Using Roll-Forward Logs	380
Issues to Be Aware of When Turning On Roll-Forward Logging	381
Location of the Roll-Forward Logs	382
Backing Up and Removing Roll-Forward Logs	383
Cautionary Note: Removing eDirectory Also Removes the Roll-Forward Logs	384
Preparing for a Restore	384
Prerequisites for Restoring	384

Locating the Right Backup Files for a Restore	385
Using Novell iManager for Backup and Restore	386
Backing Up Manually with iManager	387
Configuring Roll-Forward Logs with iManager	389
Restoring from Backup Files with iManager	390
Using the eMBox Client for Backup and Restore	393
Backing Up Manually with the eMBox Client	393
Doing Unattended Backups, Using a Batch File with the eMBox Client	396
Configuring Roll-Forward Logs with the eMBox Client	398
Restoring from Backup Files with the eMBox Client	400
Backup and Restore Command Line Options	403
Using DSBK.NLM on NetWare	409
Changes to Server-Specific Information Backup (NetWare Only)	410
Recovering the Database If Restore Verification Fails	411
Cleaning Up the Replica Ring	412
Repair the Failed Server and Readd Replicas to the Server	414
Scenarios for Backup and Restore	415
Scenario: Losing a Hard Drive Containing eDirectory in a Single-Server NetWork	416
Scenario: Losing a Hard Drive Containing eDirectory in a Multiserver Environment	416
Scenario: Losing an Entire Server in a Multiple-Server Environment	419
Scenario: Losing Some Servers in a Multiple-Server Environment	419
Scenario: Losing All Servers in a Multiple-Server Environment	419
Backing Up and Restoring NCI	421
UNIX	422
NetWare	423
Windows	424
15 Maintaining Novell eDirectory	427
Improving eDirectory Performance	427
Distributing Memory between Entry and Block Caches	428
Using the Default Cache Settings	428
Tuning LDAP for eDirectory	433
Improving eDirectory Performance on Linux, Solaris, AIX, and HP-UX Systems	435
Fine-Tuning the eDirectory Server	435
Optimizing eDirectory Cache	436
Tuning the Solaris OS for Novell eDirectory	440
Advanced Referral Costing	441
Improving Server-to-Server Connection	442
Advantages of Referral Costing	443
Enabling Advanced Referral Costing	443
Monitoring Advanced Referral Costing	443
Printing the ResolveTimes Table	444
Keeping eDirectory Healthy	444
When to Perform Health Checks	444
Health Check Overview	445
Checking eDirectory Health Using iMonitor	445
For More Information	446
Resources for Monitoring	447
Upgrading Hardware or Replacing a Server	447
Planned Hardware or Storage Device Upgrade without Replacing the Server	447
Planned Replacement of a Server	451
Restoring eDirectory after a Hardware Failure	454
16 DHost iConsole Manager	455
What is DHost?	456

Running DHost iConsole	457
Running DHost iConsole on NetWare	457
Running DHost iConsole on Windows	457
Running DHost iConsole on Linux, Solaris, AIX, and HP-UX.	457
Managing eDirectory Modules	458
Loading or Unloading Modules on NetWare	458
Loading or Unloading Modules on Windows	459
Loading or Unloading Modules on Linux, Solaris, AIX, and HP-UX.	459
Querying for DHost Information.	459
Viewing the Configuration Parameters	460
Viewing Protocol Information	460
Viewing Connection Properties	460
Viewing the Thread Pools Statistics	461
Process Stack	461
Setting the SAdmin Password	462
Setting the SAdmin Password on NetWare.	462
Setting the SAdmin Password on Windows.	462
Setting the SAdmin Password on Linux, Solaris, AIX, and HP-UX	463
17 The eDirectory Management Toolbox	465
Using the eMBox Command Line Client	465
Displaying the Command Line Help.	466
Running the eMBox Command Line Client in Interactive Mode	466
Running the eMBox Command Line Client in Batch Mode	470
eMBox Command Line Client Options	472
Establishing a Secure Connection with the eMBox Client	473
Finding Out eDirectory Port Numbers.	473
Using the eMBox Logger	474
Using the eMBox Logger Command Line Client	475
Using the eMBox Logger Feature in Novell iManager	476
18 Troubleshooting Novell eDirectory	477
Resolving Error Codes	477
Naming Objects.	477
Determining the eDirectory Version Number	478
NetWare.	478
Windows	479
Linux	479
Solaris.	480
AIX	480
HP-UX.	481
Recovering from eDirectory Replica Problems	481
The eDirectory for Windows Server Won't Start	482
The Windows Server Can't Open the eDirectory Database Files	482
Restoring eDirectory on Windows after an Emergency Repair.	483
Log Files	483
modschema.log	483
dsinstall.log	483
Troubleshooting LDIF Files	484
Understanding LDIF.	484
Debugging LDIF Files.	491
Using LDIF to Extend the Schema	496
Troubleshooting eDirectory on Linux, Solaris, AIX, and HP-UX	500
Repeated eDirectory Logins	500
Novell Public Key Infrastructure Services	500
NMAS on UNIX	501

LDAP Services	501
Novell Import Convert Export Utility	502
ndsmerge Utility	503
ndstrace Utility	503
ndsbackup Utility	503
Installation and Configuration	503
Using Ndsrepair	504
Using ndstrace	511
Troubleshooting ConsoleOne	519
Unable to Browse the eDirectory Tree	519
Running Pure IP Environments on NetWare Servers	519
Obituaries	519
Examples	521
Impact of Stuck and Orphaned Obituaries	522
Prevention	522
Troubleshooting Tips	523
Accessing HTTPSTK When DS Is Not Loaded	525
Setting the SAdmin Password on NetWare	525
Setting the SAdmin Password on Windows	526
Setting the SAdmin Password on Linux, Solaris, AIX, and HP-UX	526
Troubleshooting SNMP	527
Traps Might Not Get Generated As Expected	527
SNMP on Linux	527
SNMP on HP-UX	528
eDirectory Install Fails for Container Administrators	529
Migrating the Sun ONE Schema to Novell eDirectory	530
Step 1: Perform the Schema Cache Update operation	530
Step 2: Rectify the error LDIF file to eliminate the errors	530
Step 3: Import the LDIF File	533
Migrating the Active Directory Schema to Novell eDirectory Using ICE	533
Step 1: Perform the Schema Cache Update operation	533
Step 2: Rectify the error LDIF file to eliminate the errors	533
Step 3: Import the LDIF file	534
A NMAS Considerations	535
Setting Up a Security Container As a Separate Partition	535
Merging Trees with Multiple Security Containers	535
Product-Specific Operations to Perform prior to Tree Merge	536
Performing the Tree Merge	539
Product-Specific Operations to Perform after the Tree Merge	539
B Novell eDirectory UNIX Commands and Usage	541
General Commands	541
LDAP-Specific Commands	543
C Configuring OpenSLP for eDirectory	545
Service Location Protocol	545
SLP Fundamentals	545
Novell Service Location Providers	546
User Agents	546
Service Agents	547
Configuration Parameters	547
Open Enterprise Server SLP Configuration (Linux)	548
D How Novell eDirectory Works with DNS	551

About This Guide

This guide describes how to manage and configure Novell® eDirectory™ 8.7.3. The guide is intended for network administrators, and contains the following sections:

- ◆ Chapter 1, “Understanding Novell eDirectory,” on page 17
- ◆ Chapter 2, “Designing Your Novell eDirectory Network,” on page 67
- ◆ Chapter 3, “Managing Objects,” on page 87
- ◆ Chapter 4, “Managing the Schema,” on page 103
- ◆ Chapter 5, “Managing Partitions and Replicas,” on page 113
- ◆ Chapter 6, “Novell eDirectory Management Utilities,” on page 125
- ◆ Chapter 7, “Using Novell iMonitor 2.1,” on page 163
- ◆ Chapter 8, “Merging Novell eDirectory Trees,” on page 189
- ◆ Chapter 9, “Repairing the Novell eDirectory Database,” on page 203
- ◆ Chapter 10, “WAN Traffic Manager,” on page 229
- ◆ Chapter 11, “Understanding LDAP Services for Novell eDirectory,” on page 259
- ◆ Chapter 12, “Configuring LDAP Services for Novell eDirectory,” on page 283
- ◆ Chapter 13, “SNMP Support for Novell eDirectory,” on page 317
- ◆ Chapter 14, “Backing Up and Restoring Novell eDirectory,” on page 365
- ◆ Chapter 15, “Maintaining Novell eDirectory,” on page 427
- ◆ Chapter 16, “DHost iConsole Manager,” on page 455
- ◆ Chapter 17, “The eDirectory Management Toolbox,” on page 465
- ◆ Chapter 18, “Troubleshooting Novell eDirectory,” on page 477
- ◆ Appendix A, “NMAS Considerations,” on page 535
- ◆ Appendix B, “Novell eDirectory UNIX Commands and Usage,” on page 541
- ◆ Appendix C, “Configuring OpenSLP for eDirectory,” on page 545
- ◆ Appendix D, “How Novell eDirectory Works with DNS,” on page 551

Additional Documentation

For eDirectory installation instructions, see the *Novell eDirectory 8.7.3 Installation Guide* (<http://www.novell.com/documentation/lg/edir873/index.html>).

For documentation on the eDirectory management utility, see the *Novell iManager 2.0.x Administration Guide* (<http://www.novell.com/documentation/lg/imanager20/index.html>).

Documentation Updates

For the most recent version of this guide, see *Novell eDirectory 8.7.3 Administration Guide* (<http://www.novell.com/documentation/lg/edir873/index.html>).

Documentation Conventions

In this documentation, a greater-than symbol (>) is used to separate actions within a step and items within a cross-reference path.

A trademark symbol (® , ™ , etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark.

When a single pathname can be written with a backslash for some platforms or a forward slash for other platforms, the pathname is presented with a backslash. Users of platforms that require a forward slash, such as UNIX* , should use forward slashes as required by your software.

1

Understanding Novell eDirectory

Novell® eDirectory™ is a highly scalable, high-performing, secure directory service. It can store and manage millions of objects, such as users, applications, network devices, and data. Novell eDirectory offers a secure identity management solution that runs across multiple platforms, is internet-scalable, and extensible.

Novell eDirectory provides centralized identity management, infrastructure, Net-wide security, and scalability to all types of applications running behind and beyond the firewall. Novell eDirectory 8.7.3 includes Web-based and wireless management capabilities, allowing you to access and manage the directory and users, access rights, and network resources from a Web browser and a variety of handheld devices.

Novell eDirectory natively supports the directory standard Lightweight Directory Access Protocol (LDAP) 3 and provides support for TLS/SSL services based on the OpenSSL source code.

For more information on the eDirectory engine, see [eDirectory Process Requests \(http://developer.novell.com/research/sections/netmanage/dirprimer/2002/august/p020801.htm\)](http://developer.novell.com/research/sections/netmanage/dirprimer/2002/august/p020801.htm).

This chapter includes the following information:

- ◆ “What’s New” on page 18
- ◆ “Novell eDirectory” on page 19
- ◆ “Ease of Management through Novell iManager” on page 19
- ◆ “Object Classes and Properties” on page 22
- ◆ “Context and Naming” on page 37
- ◆ “Schema” on page 40
- ◆ “Partitions” on page 46
- ◆ “Replicas” on page 49
- ◆ “NetWare Bindery Emulation” on page 54
- ◆ “Server Synchronization in the Replica Ring” on page 54
- ◆ “Access to Resources” on page 54
- ◆ “eDirectory Rights” on page 55

What's New

Novell eDirectory 8.7.3 includes the following new features:

- ◆ Support for Windows* Server 2003
- ◆ UNIX* package-based install for all eDirectory server components for Linux*, Solaris*, and AIX*

For more information, see the *Novell eDirectory 8.7.3 Installation Guide*.

- ◆ Default port change from 80 and 443 to 8008 and 8010 (8009 on NetWare®)

For more information, see “[Finding Out eDirectory Port Numbers](#)” on page 473.

- ◆ Novell iManager 2.0.2

Provides a single Web-based management console for the administration of Novell products on NetWare, Windows, Linux, Solaris, and HP-UX*. iManager standardizes all Novell Web-based administration utilities on a single management framework. iManager also provides a best-of-breed architecture for easy development of Web-based administration and management modules through open standard application interfaces.

For more information, see the *Novell iManager 2.0.x Administration Guide* (<http://www.novell.com/documentation/lg/imanager20/index.html>).

- ◆ NMASTM 2.3

New features in Novell Modular Authentication Service™ (NMA) include advanced password policy enforcement, NMA Web Server Agent, challenge/response login method, challenge response API, and Kerberos method.

For more information, see the *Novell Modular Authentication Service 2.3 Administration Guide* (<http://www.novell.com/documentation/lg/nmas23/index.html>).

- ◆ Novell Certificate Server 2.7

New features include OCSP over SSL and directory name CRL support.

For more information, see the *Novell Certificate Server 2.7 Administration Guide* (<http://www.novell.com/documentation/lg/crt27/index.html>).

- ◆ Novell eGuide 2.1.2

Provides support for Role-Based Services used in iManager 2.0, backwards compatibility with iManager 1.5.x., enhanced iChain support (including support for all forms of authentication), new search attribute filters functionality, automatic configuration of SSL, an improved quick setup wizard, and improved counters.

For more information, see the *Novell eGuide 2.1.2 Administration Guide* (<http://www.novell.com/documentation/lg/eguide212/index.html>).

- ◆ With eDirectory 8.7.3.3, SNMP on Linux uses net-snmp-5.0.9-4.rh73.i386.rpm.

Novell eDirectory

In simplest terms, Novell eDirectory is a list of objects that represent network resources, such as network users, servers, printers, print queues, and applications. **Figure 1** shows a few of the objects as viewed in the Novell iManager management utility.

Figure 1 eDirectory Objects in iManager



Some object classes might not be available, depending on the actual schema configured on the eDirectory server and the operating system running eDirectory.

For more information on objects, see “**Object Classes and Properties**” on page 22.

The directory is physically stored as a set of database files on a server. If the server hosts file system volumes, these files are on volume sys:. If no volumes are present, the directory is stored on the server’s local disk.

If you have more than one eDirectory server on the network, the directory can be replicated on multiple servers.

Ease of Management through Novell iManager

Novell eDirectory allows for easy, powerful, and flexible management of network resources. It also serves as a repository of user information for groupware and other applications. These applications access your directory through the industry-standard Lightweight Directory Access Protocol (LDAP).

eDirectory ease-of-management features include a powerful tree structure, an integrated management utility, and single login and authentication.

Novell iManager lets you manage the directory and users, and access rights and network resources within the directory, from a Web browser and a variety of handheld devices. The eDirectory plug-ins to iManager give you access to basic directory management tasks, and to the eDirectory management utilities you previously had to run on the eDirectory server, such as DSRepair, DSMerge, and Backup and Restore.

After iManager is installed on a Web server, you can access iManager from any server or workstation running Internet Explorer 5.5 or later or Netscape 6.2 or later.

For more information, see the *Novell iManager 2.0.x Administration Guide* (<http://www.novell.com/documentation/lg/imanager20/index.html>).

Powerful Tree Structure

Novell eDirectory organizes objects in a tree structure, beginning with the top Tree object, which bears the tree's name.

Whether your eDirectory servers are running NetWare, UNIX, or Windows, all resources can be kept in the same tree. You won't need to access a specific server or domain to create objects, grant rights, change passwords, or manage applications.

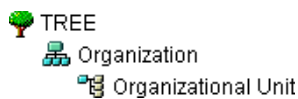
The hierarchical structure of the tree gives you great management flexibility and power. These benefits primarily result from the following two features:


- ◆ “Container Objects” on page 20
- ◆ “Inheritance” on page 21


Container Objects


Container objects allow you to manage other objects in sets, rather than individually. There are three common classes of container objects, as seen in [Figure 2](#):

Figure 2 Common Classes of Container Objects




 The Tree object is the top container object in the tree. It usually contains your company's Organization object.

 Organization is normally the first container class under the Tree object. The Organization object is typically named after your company. Small companies keep management simple by having all other objects directly under the Organization object.

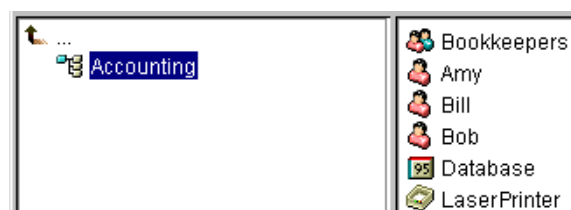
 Organizational Unit objects can be created under the Organization to represent distinct geographical regions, network campuses, or individual departments. You can also create Organizational Units under other Organizational Units to further subdivide the tree.

Other classes of container objects are Country and Locality, which are typically used only in multinational networks.

 The Domain object can be created under the Tree object or under Organization, Organizational Unit, Country, and Locality objects.

You can perform one task on the container object that applies to all objects within the container. Suppose you want to give a user named Amy complete management control over all objects in the Accounting container. (See [Figure 3](#).)

Figure 3 Container Object



To do this, right-click the Accounting object, select Trustees of This Object, then add Amy as a trustee. Next, select the rights you want Amy to have, then click OK. Now Amy has rights to manage the Database application, the Bookkeepers group, the LaserPrinter printer, and the users Amy, Bill, and Bob.

Inheritance

Another powerful feature of eDirectory is rights inheritance. Inheritance means that rights flow down to all containers in the tree. This allows you to grant rights with very few rights assignments. For example, suppose you want to grant management rights to the objects shown in [Figure 4 on page 21](#).

Figure 4 Sample eDirectory Objects



You could make any of the following assignments:

- ◆ If you grant a user rights to Allentown, the user can manage only objects in the Allentown container.
- ◆ If you grant a user rights to East, the user can manage objects in the East, Allentown, and Yorktown containers.
- ◆ If you grant a user rights to YourCo, the user can manage any objects in any of the containers shown.

For more information on assigning rights, see [“eDirectory Rights” on page 55](#).

Web-Based Management Utility

Novell iManager is a browser-based tool used for administering, managing, and configuring eDirectory objects. Novell iManager gives you the ability to assign specific tasks or responsibilities to users and to present the user with only the tools (with the accompanying rights) necessary to perform those sets of tasks.

You can use iManager on any server or workstation running Internet Explorer 5.5 SP2 or later or Netscape 6.2 or later to perform the following supervisory tasks:

- ◆ Configure LDAP- and XML-based access to eDirectory
- ◆ Create objects representing network users, devices, and resources
- ◆ Define templates for creating new user accounts
- ◆ Find, modify, move, and delete network objects
- ◆ Define rights and roles to delegate administrative authority
- ◆ Extend the eDirectory schema to allow custom object types and properties
- ◆ Partition and replicate the eDirectory database across multiple servers

- ◆ Run eDirectory management utilities such as DSRepair, DSMerge, and Backup and Restore

You can use iManager to perform other management functions based on plug-ins that have been loaded into iManager. The following eDirectory plug-ins are installed with iManager 1.5:

- ◆ eDirectory Backup and Restore
- ◆ eDirectory Log Files
- ◆ eDirectory Merge
- ◆ eDirectory Repair
- ◆ eDirectory Service Manager
- ◆ eGuide Content
- ◆ iManager Base Content
- ◆ Import Convert Export Wizard
- ◆ Index Management
- ◆ iPrint
- ◆ LDAP
- ◆ NLS
- ◆ NMAS
- ◆ PKI/Certificate
- ◆ Filtered Replica Configuration Wizard
- ◆ SNMP
- ◆ WAN Traffic Manager

For more information on installing, configuring, and running iManager, see the *Novell iManager 2.0.x Administration Guide* (<http://www.novell.com/documentation/lg/imanager20/index.html>).

Single Login and Authentication

With eDirectory, users log in to a global directory, so you don't need to manage multiple server or domain accounts for each user, and you don't need to manage trust relationships or pass-through authentication among domains.

A security feature of the directory is authentication of users. Before a user logs in, a User object must be created in the directory. The User object has certain properties, such as a name and password.

When the user logs in, eDirectory checks the password against the one stored in the directory for that user and grants access if they match.

Object Classes and Properties

The definition of each type of eDirectory object is called an object class. For instance, User and Organization are object classes. Each class of object has certain properties. A User object, for example, has Login Name, Password, Last Name, and many other properties.







The schema defines the object classes and properties, along with the rules of containment (what containers can contain which objects). eDirectory ships with a base schema that you, or the applications you use, can extend. For more information about schemas, see [“Schema” on page 40](#).

Container objects contain other objects and are used to divide the tree into branches, while leaf objects represent network resources.













List of Objects






The following tables list eDirectory object classes. Added services can create new object classes in eDirectory that are not listed below. Also, all classes might not be available on all server operating systems hosting eDirectory.

eDirectory Container Object Classes

iManager Icon	Container Object (Abbreviation)	Description
	Tree	Represents the beginning of your tree. For more information, see “Tree” on page 25 .
	Country (C)	Designates the countries where your network resides and organizes other directory objects within the country. For more information, see “Country” on page 27 .
	License Container (LC)	Created automatically when you install a license certificate or create a metering certificate using Novell Licensing Services (NLS) technology. When an NLS-enabled application is installed, it adds a License Container container object to the tree and a License Certificate leaf object to that container.
	Organization (O)	Helps you organize other objects in the directory. The Organization object is a level below the Country object (if you use the Country object). For more information, see “Organization” on page 25 .
	Organizational Unit (OU)	Helps you to further organize other objects in the directory. The Organizational Unit object is a level below the Organization object. For more information, see “Organizational Unit” on page 26 .
	Domain (DC)	Helps you to further organize other objects in the directory. The Domain object can be created under the Tree object or under Organization, Organizational Unit, Country, and Locality objects. For more information, see “Domain” on page 27 .

eDirectory Leaf Object Classes

iManager Icon	Leaf Object	Description
	AFP Server	Represents an AppleTalk* Filing Protocol server that operates as a node on your eDirectory network. It usually also acts as a NetWare router to, and the AppleTalk server for, several Macintosh* computers.
	Alias	Points to the actual location of an object in the directory. Any directory object located in one place in the directory can also appear to be in another place in the directory by using an Alias. For more information, see "Alias" on page 35 .
	Application	Represents a network application. Application objects simplify administrative tasks such as assigning rights, customizing login scripts, and launching applications.
	Computer	Represents a computer on the network.
	Directory Map	Refers to a directory in the file system. For more information, see "Directory Map" on page 36 .
	Group	Assigns a name to a list of User objects in the directory. You can assign rights to the group instead of to each user; then the rights transfer to each user in the group. For more information, see "Group" on page 31 .
	License Certificate	Use with NLS technology to install product license certificates as objects in the database. License Certificate objects are added to the Licensed Product container when an NLS-aware application is installed.
	Organizational Role	Defines a position or role within an organization.
	Print Queue	Represents a network print queue.
	Print Server	Represents a network print server.
	Printer	Represents a network printing device.
	Profile	Represents a login script used by a group of users who need to share common login script commands. The users don't need to be in the same container. For more information, see "Profile" on page 37 .

iManager Icon	Leaf Object	Description
	Server	Represents a server running any operating system. For more information, see “Server” on page 28 .
	Template	Represents standard User object properties that can be applied to new User objects.
	Unknown	Represents an object for which iManager has no custom icon.
	User	Represents the people who use your network. For more information, see “User” on page 29 .
	Volume	Represents a physical volume on the network. For more information, see “Volume” on page 29 .

Container Object Classes

Tree



The Tree container, formerly [Root], is created when you first install eDirectory on a server in your network. As the top-most container, it usually holds Organization objects, Country objects, or Alias objects.

What Tree Represents

Tree represents the top of your tree.

Usage

Tree is used to make universal rights assignments. Because of inheritance, any rights assignments you make to Tree as the target apply to all objects in the tree. See [“eDirectory Rights” on page 55](#). The [Public] trustee has the Browse right and Admin has the Supervisor right to Tree by default.

Important Properties

The Tree object has a Name property, which is the tree name you supply when installing the first server. The tree name is shown in the hierarchy of iManager.

Organization



An Organization container object is created when you first install eDirectory on a server in your network. As the top-most container under Tree, it usually holds Organizational Unit objects and leaf objects.

The User object named Admin is created by default in your first Organization container.

What an Organization Object Represents

Normally the Organization object represents your company, although you can create additional Organization objects under Tree. This is typically done for networks with distinct geographical districts or for companies with separate eDirectory trees that have merged.

Usage

The way you use Organization objects in your tree depends on the size and structure of your network. If the network is small, you should keep all leaf objects under one Organization object.

For larger networks, you can create Organizational Unit objects under the Organization to make resources easier to locate and manage. For example, you can create Organizational Units for each department or division in your company.

For networks with multiple sites, you should create an Organizational Unit for each site under the Organization object. That way, if you have (or plan to have) enough servers to partition the directory, you can do so logically along site boundaries.

For easy sharing of company-wide resources such as printers, volumes, or applications, create corresponding Printer, Volume, or Application objects under the Organization.

Important Properties

The most useful properties for Organization are listed below. Only the Name property is required. For a complete list of properties, select an Organization object in iManager. To display a description for each page of properties, click Help.

- ◆ Name

Typically, the Name property is the same as your company's name. Of course, you can shorten it for simplicity. For instance, if the name of your company is Your Shoe Company, you might use YourCo.

The Organization name becomes part of the context for all objects created under it.

- ◆ Login Script

The Login Script property contains commands that are executed by any User objects directly under the Organization. These commands are run when a user logs in.

Organizational Unit



You can create Organizational Unit (OU) container objects to subdivide the tree.

Organizational Units are created with iManager under an Organization, Country, or another Organizational Unit.

Organizational Units can contain other Organizational Units and leaf objects such as User and Application objects.

What an Organizational Unit Object Represents

Normally the Organizational Unit object represents a department, which holds a set of objects that commonly need access to each other. A typical example is a set of Users, along with the Printers, Volumes, and Applications that those Users need.

At the highest level of Organizational Unit objects, each Organizational Unit can represent each site (separated by WAN links) in the network.

Usage

The way you use Organizational Unit objects in your tree depends on the size and structure of your network. If the network is small, you might not need any Organizational Units.

For larger networks, you can create Organizational Unit objects under the Organization to make resources easier to locate and manage. For example, you can create Organizational Units for each

department or division in your company. Remember that administration is easiest when you keep User objects together in the Organizational Unit with the resources they use most frequently.

For networks with multiple sites, you can create an Organizational Unit for each site under the Organization object. That way, if you have (or plan to have) enough servers to partition the directory, you can do so logically along site boundaries.

Important Properties

The most useful properties for the Organizational Unit are listed below. Only the Name property is required. For a complete list of properties, select an Organizational Unit object in iManager. To display a description for each page of properties, click Help.

- ◆ Name


Typically, the Name property is the same as the department name. Of course, you can shorten it for simplicity. For instance, if the name of your department is Accounts Payable, you can shorten it to AP.

The Organizational Unit name becomes part of the context for all objects created under it.

- ◆ Login Script

The Login Script property contains commands that are executed by any User objects directly under the Organizational Unit. These commands are run when a user logs in.

Country

 You can create Country objects directly under the Tree object using iManager. Country objects are optional and required only for connection to certain X.500 global directories.

What a Country Object Represents

The Country object represents the political identity of its branch of the tree.

Usage


Most administrators do not create a Country object, even if the network spans countries, since the Country object only adds an unnecessary level to the tree. You can create one or many Country objects under the Tree object, depending on the multinational nature of your network. Country objects can contain only Organization objects.

If you do not create a Country object and find that you need one later, you can always modify the tree to add one.

Important Properties

The Country object has a two-letter Name property. Country objects are named with a standard two-letter code such as US, UK, or DE.

Domain

 You can create Domain objects directly under the Tree object using iManager. You can also create them under Organization, Organization Unit, Country, and Location objects.

What a Domain Object Represents

The Domain object represent DNS domain components. Domain objects let you use your Domain Name System location of services resource records (DNS SRV) to locate services in your tree.

Using Domain objects, a tree could look something like this:

```
DC=Novell.DC=Provo.DC=USA
```

In this example, all subcontainers are domains. You can also use Domain objects in a mixed tree, such as:

```
DC=Novell.O=Provo.C=USA
```

Or

```
OU=Novell.DC=Provo.C=USA
```

Usually, the topmost Domain is the overall Tree, with subdomains under Tree. For example, `machine1.novell.com` could be represented by `DC=machine1.DC=novell.DC=com` in a tree representation. Domains give you a more generic way to set up an eDirectory tree. If all containers and subcontainers are DC objects, users do not need to remember C, O, or OUs when searching for objects.

Usage

NetWare 4 and 5 trees cannot have Domain objects at the top of the tree. With NetWare 4 and 5, the NCP Server object can be placed in an Organization, Country, Organizational Unit, or Locality container, but not in a Domain container. With NetWare 6, however, you can place Domain objects at the top of the tree, and you can place the NCP Server object in a Domain container.

For older installations of NetWare (such as NetWare 4), when you prepare the tree to install or upgrade to NetWare 5 or later, the `nds500.sch` file will automatically run. After the first server is installed into the tree, this file extends the schema to allow the Domain container to be created anywhere and hold most directory objects.

Leaf Object Classes

Server



A Server object is automatically created in the tree whenever you install eDirectory on a server. The object class can be any server running eDirectory.

You can also create a Server object to represent a NetWare 2 or NetWare 3 bindery server.

What a Server Object Represents

The Server object represents a server running eDirectory or a bindery-based (NetWare 2 or NetWare 3) server.

Usage


The Server object serves as a reference point for replication operations. A Server object that represents a bindery-based server allows you to manage the server's volumes with iManager.

Important Properties

The Server object has a Network Address property, among others. The Network Address property displays the protocol and address number for the server. This is useful for troubleshooting at the packet level.

For a complete list of properties, select a Server object in iManager. To display a description for each page of properties, click Help.

Volume

 When you create a physical volume on a server, a Volume object is automatically created in the tree. By default, the name of the Volume object is the server's name with an underscore and the physical volume's name appended (for example, YOSERVER_SYS).

Volume objects are supported only on NetWare. UNIX file system partitions cannot be managed using Volume objects.

What a Volume Object Represents

A Volume object represents a physical volume on a server, whether it is a writable disk, a CD, or other storage medium. The Volume object in eDirectory does not contain information about the files and directories on that volume, although you can access that information through iManager. File and directory information is retained in the file system itself.

Usage

In iManager, click the Volume icon to manage files and directories on that volume. iManager provides information about the volume's free disk space, directory entry space, and compression statistics.

You can also create Volume objects in the tree for NetWare 2 and NetWare 3 volumes.

Important Properties

In addition to the required Name and Host Volume properties, there are other important Volume properties.

- ◆ Name

This is the name of the Volume object in the tree. By default, this name is derived from the name of the physical volume, though you can change the object name.

- ◆ Host Server

This is the server that the volume resides on.


- ◆ Version

This is the NetWare or eDirectory version of the server hosting the volume.

- ◆ Host Volume

This is the physical volume name. Because the actual Volume object name does not need to reflect the physical volume name, this property is necessary to associate the Volume object with the physical volume.

User

 A User object is required for logging in. When you install the first server into a tree, a User object named Admin is created. Log in as Admin the first time.

You can use the following methods to create or import User objects:

- ◆ iManager

For more information on iManager, see the *Novell iManager 2.0.x Administration Guide* (<http://www.novell.com/documentation/lg/imanager20/index.html>).

- ◆ Batches from database files

For more information on using batch files, see [“Designing the eDirectory Tree” on page 68](#).

- ◆ NetWare upgrade utilities

For more information on upgrade utilities, including importing users from existing bindery servers, see [“Designing the eDirectory Tree” on page 68](#).

What a User Object Represents

A User object represents a person who uses the network.

Usage

You should create User objects for all users who need to use the network. Although you can manage User objects individually, you can save time by

- ◆ Using Template objects to set default properties for most User objects. The Template applies automatically to new Users you create (not to already existing ones).
- ◆ Creating Group objects to manage sets of Users.
- ◆ Assigning rights using the container objects as trustees when you want that assignment to apply to all User objects in the container.
- ◆ Selecting multiple User objects by using Shift+click or Ctrl+click. When you do, you can change property values for all selected User objects.

Important Properties

User objects have over 80 properties. For a complete list of properties, select a User object in iManager. To display a description for each page of properties, click Help.

The Login Name and Last Name properties are required. These and some of the most useful properties are listed below.

- ◆ Account Expiration Date lets you limit the life of a user account. After the expiration date, the account is locked so the user cannot log in.
- ◆ Account Disabled has a system-generated value that indicates a lock on the account so the user cannot log in. The lock might occur if the account has expired or because the user has given too many incorrect passwords in succession.
- ◆ Force Periodic Password Changes lets you enhance security by requiring the user to change passwords after a specified interval.
- ◆ Group Memberships lists all the Group objects that include the User as a member.
- ◆ Home Directory refers to a NetWare volume and file system path for the user's own files. Most administrators like to create such a directory so that a user's working files can be kept on the network.

The directory referred to in this property can be automatically created when you create the User object.

- ◆ Last Login is a system-generated property that lists the date and time that the user last logged in.
- ◆ Last Name, although required, is not used directly by eDirectory. Applications that take advantage of the eDirectory name base can use this property, along with other identification properties such as Given Name, Title, Location, and Fax Number.

- ◆ Limit Concurrent Connections lets you set the maximum number of sessions a user can have on the network at any given time.
- ◆ Login Name is the name shown in iManager by the User icon. It is also the name supplied by the user when logging in.

eDirectory does not require that login names be unique throughout the network, only in each container. However, you might want to keep login names unique across the company to simplify administration.

Typically, login names are a combination of first and last names, such as STEVEJ or SJONES for Steve Jones.

- ◆ Login Script lets you create specific login commands for a User object. When a user logs in, the container login script runs first. Then a profile login script runs if the User object has been added to the membership list of a Profile object. Finally, the user login script runs (if one exists).

You should put most of the login commands in container login scripts to save administrative time. The user login script can be edited to manage unique exceptions to common needs.

- ◆ Login Time Restrictions lets you set times and days when the user can log in.
- ◆ Network Addresses contains system-generated values that list all the IPX™ and/or IP addresses that the user is logged in from. These values are useful for troubleshooting network problems at the packet level.
- ◆ Require a Password lets you control whether the user must use a password. Other related properties let you set common password constraints such as password length.
- ◆ Rights to Files and Directories lists all rights assignments made for this user to the NetWare file system. Using iManager, you can also check a user's effective rights to files and directories, which include those inherited from other objects.

Group



You can create Group objects to help you manage sets of User objects.

What a Group Object Represents

A Group object represents a set of User objects.

Usage

Container objects let you manage all User objects in that container, and Group objects are for subsets within a container or in multiple containers.

Group objects have two main purposes:

- ◆ They allow you to grant rights to a number of User objects at once.
- ◆ They allow you to specify login script commands using the `IF MEMBER OF` syntax.

Static Groups

Static groups identify the member objects explicitly. Each member is assigned to the group explicitly.

These groups provide a static list of members, as well as referential integrity between the members list of the group and the members of attributes on an object. Group membership is managed explicitly through the member attribute.

Dynamic Groups

Dynamic groups use an LDAP URL to define a set of rules which, when matched by eDirectory User objects, define the members of the group. Dynamic group members share a common set of attributes as defined by the search filter specified in the URL. For more information on the LDAP URL format, see [RFC 2255 \(http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2255.html\)](http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2255.html).

Dynamic groups let you specify the criteria to be used for evaluating membership in a group. The actual members of the group are dynamically evaluated by eDirectory, which lets you define the group members in terms of a logical grouping and lets eDirectory automatically add and remove group members. This solution is more scalable, reduces administrative costs, and can supplement normal groups in LDAP to provide increased flexibility.

eDirectory lets you create a dynamic group when you want to automatically group users based on any attribute, or when you want to apply ACLs to specific groups that contain matching DN's. For example, you can create a group that automatically includes any DN that contains the attribute `Department=Marketing`. If you apply a search filter for `Department=Marketing`, the search returns a group including all DN's containing the attribute `Department=Marketing`. You can then define a dynamic group from the search results based on this filter. Any User added to the directory who matches the `Department=Marketing` criteria is automatically added to the group. Any User whose `Department` is changed to another value (or who is removed from the directory) is automatically removed from the group.

Dynamic groups are created in eDirectory by creating an object of type `objectclass=dynamicGroup`. A static Group object can be converted into a dynamic group by associating an auxiliary class, `dynamicGroupAux`, to the Group object. The dynamic group has the `memberQueryURL` attribute associated with it.

A `dgIdentity` attribute can be set on the Dynamic Group object to the distinguished name of an entry, whose credentials and rights should be used to expand the dynamic members of the group.

The groups are managed using the `memberQueryURL`. A typical `memberQueryURL` has a base DN, a scope, a filter, and an optional extension. The base DN specifies the search base. Scope specifies the levels below the base to search, and filter is the search filter based on which entries are selected from within the specified scope.

NOTE: To address exceptions to the listing created by the `memberQueryURL`, dynamic groups also allow for explicit inclusion and exclusion of users.

Dynamic groups can be created and managed through Novell iManager. You can access the Dynamic Group management tasks by clicking the Dynamic Groups role on the Roles and Tasks page.

You can also use LDAP commands to manage such groups. The most useful properties associated with dynamic groups are `dgIdentity` and `memberQueryURL`.

Important Properties

The most useful properties of the Group object are Members and Rights to Files and Directories. For a complete list of properties, select a Group object in iManager. To display a description for each page of properties, click Help.

- ◆ `dgIdentity`

This property holds the DN whose identity the dynamic group will use for authentication while searching. The identity must be on the same partition as the dynamic group. The object specified by `dgIdentity` should have the necessary rights to do the search specified in the `memberQueryURL` attribute.

For example, if memberQueryURL value is

```
"ldap:///o=nov??sub?(title=*)"
```

then dgIdentity should have read/compare rights on the attribute title below the container o=nov.

- ◆ dgTimeout

This property specifies the maximum duration a server can take to read or compare a member attribute before it times out. When the server exceeds this dgTimeout value, the -6016 error is displayed.

- ◆ memberQueryURL

This property defines the set of rules that match with the attributes of the group members.

memberQueryURL is a multivalued attribute according to its schema definition. Although memberQueryURL is multivalued, eDirectory 8.6.1 servers used only the first value of memberQueryURL.

For example:

An administrator creates a dynamic group, which has two memberQueryURL values:

```
"ldap:///o=nov??sub?cn="
```

```
"ldap:///o=org??sub?cn="
```

eDirectory 8.6.x servers use "ldap:///o=nov??sub?cn=" to compute the members of the group. They accept more than one query, but only read the first query.

This limitation is overcome in eDirectory 8.7.3. eDirectory 8.7.3 servers compute the members based on all the memberQueryURL values, and the set of members is the union of the members computed using each of the memberQueryURL values.

In the above example, resultant members of the dynamic group are all entries under o=org and o=nov, which have cn values.

The memberQuery URL supports two URL extensions:

- ◆ x-chain

The request is chained if the URL value contains x-chain extension.

Example:

```
ldap:///O=org??sub?(objectClass=inetOrgPerson)?x-chain
```

- ◆ x-sparse

The sparse replicas are considered for dynamic group member search queries if the URL value contains x-sparse extension.

```
ldap:///O=org??sub?(objectClass=inetOrgPerson)?x-sparse
```

IMPORTANT: These extensions are mutually exclusive. When x-chain is present, x-sparse is automatically ignored.

- ◆ member

This property lists all objects in the group. Rights assignments made to the Group object apply to all members of that group. Adding values to the member property of a dynamic group will add the static members to the dynamic group. This can be used for specific inclusion of members.

- ◆ excludedMember

The property holds the DNs that are specifically excluded from the membership list of the dynamic group. This can be used to construct exclusion lists for dynamic groups.

excludedMember is used to exclude DNs from being dynamic members of a dynamic group.

Thus, a DN is a dynamic member of a dynamic group only if it is selected by the member criteria specified by memberQueryURL and is not listed in excludedMember or explicitly added to uniqueMember or member.

- ◆ staticMember

This property reads the static members of a dynamic group and also determines whether a DN is a static member of a dynamic group. staticMember can find the dynamic groups in which a DN is a static member alone and can also find which groups have dynamic members and no static members.

To add this property to the existing dynamic groups, extend the schema using dgstatic.sch.

Upgrading a Dynamic Groups on Pre-eDirectory 8.6.1 Databases

Dynamic groups functionality requires some internal values stored on the Dynamic Group objects, which are created either when a dynamic group is locally created or received as a part of synchronization.

Although older servers can hold dynamic groups, they are unable to generate these values, because dynamic groups were introduced in eDirectory 8.6.1.

In eDirectory 8.6.2, automatic upgrade of the Dynamic Group objects in a pre-8.6.1 database to match a eDirectory 8.6.1 database was implemented.

Support for Additional Syntaxes in memberQueryURL

The memberQueryURL attribute can hold a search filter that the eDirectory server uses to compute the members of a dynamic group.

In eDirectory 8.6.1, the syntaxes of attributes used in the filter were restricted only to the following basic string types:

- ◆ SYN_CE_STRING
- ◆ SYN_CI_STRING
- ◆ SYN_PR_STRING
- ◆ SYN_NU_STRING
- ◆ SYN_CLASS_NAME
- ◆ SYN_TEL_NUMBER
- ◆ SYN_INTEGER
- ◆ SYN_COUNTER
- ◆ SYN_TIME
- ◆ SYN_INTERVAL
- ◆ SYN_BOOLEAN
- ◆ SYN_DIST_NAME
- ◆ SYN_PO_ADDRESS
- ◆ SYN_CI_LIST

- ◆ SYN_FAX_NUMBER
- ◆ SYN_EMAIL_ADDRESS

In eDirectory 8.7.3, the following additional attribute syntaxes are supported in a memberQueryURL value:

- ◆ SYN_PATH
- ◆ SYN_TIMESTAMP
- ◆ SYN_TYPED_NAME

In both eDirectory 8.6.1 and eDirectory 8.7.x, binary syntaxes like SYN_OCTET_STRING and SYN_NET_ADDRESS are not supported in the memberQueryURL search filters.

For more information, see [How to Manage and Use Dynamic Groups in Novell eDirectory \(http://developer.novell.com/research/appnotes/2002/april/05/a020405.htm\)](http://developer.novell.com/research/appnotes/2002/april/05/a020405.htm).

Alias

You can create an Alias object that points to another object in the tree. An Alias object gives a user a local name for an object that lies outside their container.

When you rename a container, you have the option of creating an Alias in the former container's place that points to the new name. Workstations and login script commands that reference objects in the container can still access the objects without having the container name updated.

What an Alias Object Represents

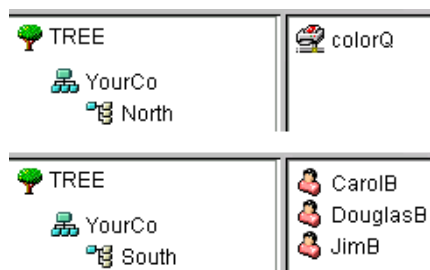
An Alias object represents another object, which can be a container, User object, or any other object in the tree. An Alias object does not carry trustee rights of its own. Any trustee authority you grant to the Alias object applies to the object it represents. The Alias can be a target of a trustee assignment, however.

Usage

Create an Alias object to make name resolution easier. Because object naming is simplest for objects in the current context, you should create Alias objects there that point to any resources outside the current context.

For example, suppose users log in and establish a current context in the South container as shown in [Figure 5](#), but need access to the Print Queue object named ColorQ in the North container.

Figure 5 Sample Containers



You can create an Alias object in the South container, as shown in [Figure 6](#).


Figure 6 Alias Object in eDirectory Container

The Alias object points to the original ColorQ object, so setting up printing for the users involves a local object.

Important Properties

Alias objects have an Aliased Object property, which associates the Alias object with the original object.

Directory Map

 The Directory Map object is a pointer to a path in the server file system. It allows you to make simpler references to directories.

If your network has no NetWare volumes, you cannot create Directory Map objects.

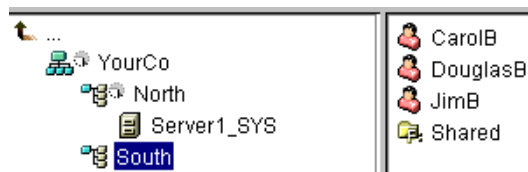
What a Directory Map Object Represents

A Directory Map object represents a directory on a NetWare volume. (An Alias object, on the other hand, represents an object.)

Usage

Create a Directory Map object to make drive mapping simpler, particularly in login scripts. Using a Directory Map object allows you to reduce complex file system paths to a single name.

Also, when you change the location of a file, you don't need to change login scripts and batch files to reference the new location. You only need to edit the Directory Map object. For example, suppose you were editing the login script for the container South, shown in [Figure 7](#).

Figure 7 Sample eDirectory Container

A command mapping drives to the Shared directory on volume sys: would look like the following:

```
MAP N:=sys.North.:Shared
```

If you created the Shared Directory Map object, the map command would be much simpler:

```
MAP N:=Shared
```

Important Properties

The Directory Map object has the following properties:

- ◆ Name
Identifies the object in the directory (for example, Shared) and is used in MAP commands.
- ◆ Volume
Contains the name of the Volume object that the Directory Map object references, such as Sys.North.YourCo.
- ◆ Path
Specifies the directory as a path from the root of the volume, such as public\winnt\nls\english.

Profile



Profile objects help you manage login scripts.

What a Profile Object Represents

A Profile object represents a login script that runs after the container login script and before the user login script.

Usage

Create a Profile object if you want login script commands to run for only selected users. The User objects can exist in the same container or be in different containers. After you have created the Profile object, you add the commands to its Login Script property. Then make the User objects trustees of the Profile object and add the Profile object to their Profile Membership property.

Important Properties

The Profile object has two important properties:

- ◆ Login Script
Contains the commands you want to run for users of the Profile.
- ◆ Rights to Files and Directories
If you have INCLUDE statements in the login script, you need to give the Profile object rights to the files included with the Rights to Files and Directories property.

Context and Naming

The context of an object is its position in the tree. It is nearly equivalent to a DNS domain.

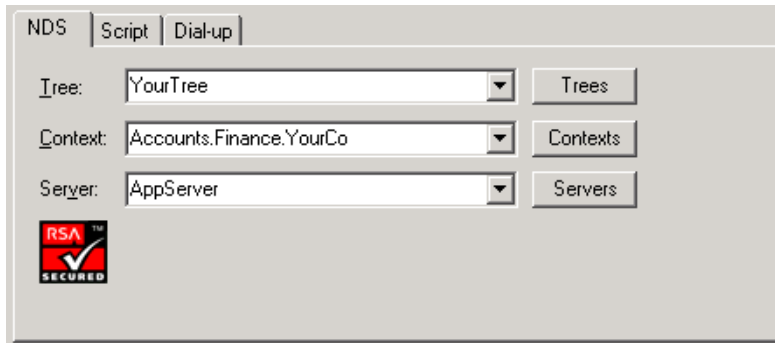
You can see in the following figure that User Bob is in Organizational Unit Accounts, which is in Organizational Unit Finance, which is in Organization YourCo.

Figure 8 Sample eDirectory Container



Sometimes, however, you need to express the context of an object in an eDirectory utility. For example, you could be setting up Bob's workstation and need to supply a name context, as shown in [Figure 9 on page 38](#).

Figure 9 Novell Client NDS Page



The context is specified as a list of containers separated by periods, between the object in question and the top of the Tree. In the example above, User object Bob is in the container Accounts, which is in the container Finance, which is in the container YourCo.

Distinguished Name

The distinguished name of an object is its object name with the context appended. For example, the complete name of User object Bob is Bob.Accounts.Finance>YourCo. The maximum number of characters in a distinguished name is limited to 256 characters of any language.

Typeful Name

Sometimes typeful names are displayed in eDirectory utilities. Typeful names include the object type abbreviations listed in the following table:

Object Class	Type	Abbreviation
All leaf object classes	Common Name	CN
Organization	Organization	O
Organizational Unit	Organizational Unit	OU
Country	Country	C
Locality	Locality or State/Province	L or S

In creating a typeful name, eDirectory uses the type abbreviation, an equal sign, and the object's name. For instance, Bob's partial typeful name is CN=Bob. Bob's complete typeful name is CN=Bob.OU=Accounts.OU=Finance.O=YourCo. You can use typeful names interchangeably with typeless names in eDirectory utilities.

Name Resolution

The process eDirectory uses to find an object's location in the directory tree is called *name resolution*. When you use object names in eDirectory utilities, eDirectory resolves the names relative to either the current context or the top of the tree.

Current Workstation Context

Workstations have a context set when the networking software runs. This context relatively identifies the location of the workstation in the network. For example, Bob's workstation would be set to the current context as follows:

Accounts.Finance.YourCo

Current context is a key to understanding the use of leading periods, relative naming, and trailing periods, discussed in the following sections.

Leading Period

Use a leading period to resolve the name from the top of the tree, no matter where the current context is set. In the example below, the leading period tells the CX (Change Context) utility to resolve the name relative to the top of the tree.

CX .Finance.YourCo

eDirectory interprets the command as “Change the context to the Finance container, which is in the YourCo container, resolved from the top of the tree.”

Relative Naming

Relative naming means that names are resolved relative to the workstation's current context, rather than from the top of the tree. Relative naming never involves a leading period, since a leading period indicates resolution from the top of the tree.

Suppose a workstation's current context is set to Finance. (See [Figure 10](#).)

Figure 10 Sample eDirectory Container



The relative object name of Bob is

Bob.Accounts

eDirectory interprets the name as “Bob, which is in Accounts, resolved from the current context, which is Finance.”

Trailing Periods

Trailing periods can be used only in relative naming. Therefore, you can't use both a leading period and a trailing period. A trailing period changes the container that eDirectory resolves the name from.

Each trailing period changes the resolution point one container toward the top of the tree. For example, suppose you want to change your workstation's current context from Timmins to Allentown in the example in [Figure 11 on page 40](#).

Figure 11 Sample eDirectory Container



The proper CX command uses relative naming with trailing periods:

```
CX Allentown.East..
```

eDirectory interprets the command as “Change the context to Allentown, which is in East, resolved from two containers up the tree from the current context.”

Similarly, if Bob is in the Allentown container and your workstation's current context is Timmins, then Bob's relative name would be

```
Bob.Allentown.East..
```

Context and Naming on UNIX

When UNIX user accounts are migrated to eDirectory, the eDirectory context is not used to name users. The context of the user is determined by the UAM component.

Schema

Schema defines the types of objects that can be created in your tree (such as Users, Printers, and Groups) and what information is required or optional at the time the object is created. Every object has a defined schema class for that type of object.

The schema that originally shipped with the product is called the base schema. After the base schema has been modified in any way—such as adding a new class or a new attribute—then it is considered the extended schema.

You aren't required to extend the schema, but you have the ability to do so. The Schema role in iManager lets you extend the schema to meet organizational needs. For example, you might want to extend your schema if your organization requires special footwear for employees and you need to keep track of employee shoe sizes. You might want to create a new attribute called Shoe Size and then add it to the User class.

For more information, see [Chapter 4, “Managing the Schema,” on page 103](#).

Schema Management

The Schema role in Novell iManager lets users who have the Supervisor rights to a tree customize the schema of that tree. The Schema role, and its associated tasks, is available on the Roles and Task page in iManager.

Use the Schema role to

- ◆ View a list of all classes and attributes in the schema.
- ◆ View information on an attribute such as its syntax and flags.
- ◆ Extend the schema by adding a class or an attribute to the existing schema.
- ◆ Create a class by naming it and specifying attributes, flags, containers that it can be added to, and parent classes that it can inherit attributes from.
- ◆ Create an attribute by naming it and specifying its syntax and flags.
- ◆ Add an optional attribute to an existing class.
- ◆ Delete a class or attribute that is not used or that is obsolete.

Schema Classes, Attributes, and Syntaxes

Classes

A class is like a template for a directory object. A directory object is a class that has been filled in with data. In other words:

CLASS + DATA = DIRECTORY OBJECT

Each class has a class name, an inheritance class (unless it is at the top of the class hierarchy), class flags, and a group of attributes. Classes are named like directory objects (User, Printer, Queue, Server, etc.), yet they are just structure, with no content.

An inheritance class is a class that is a starting point for defining other object classes. All of the attributes of the inheritance class are inherited by the classes that come below it in the class hierarchy.

A class hierarchy shows how a class is associated with its parent classes. This is a way of associating similar classes and allowing attributes to be inherited. It also defines the types of containers the class is valid in.

When creating a new class, you can use the class hierarchy and the additional attributes available to customize each class. You can specify an inheritance class (which allows the new class to inherit all of the attributes and flags of a class higher in the hierarchy) and then customize the new class by selecting one or more attributes to add to those that were inherited. The additional attributes can be selected as mandatory, naming, or optional attributes.

You can also modify existing classes by adding optional attributes.

Attributes

Attributes are the data fields in the eDirectory database. For example, if a class is like a form, then an attribute is one field on the form. When an attribute is created, it is named (such as *surname* or *employee number*) and given a syntax type (such as *string* or *number*). From then on, it is available in the attribute lists in Schema Manager.

Syntaxes

There are several syntax options to choose from. These are used to specify the type of data entered for each attribute. The syntax can be specified only when an attribute is created. You cannot modify it later. Available syntaxes include the following:

- ◆ **Backlink**

Used to keep track of other servers referring to an object. It is used for internal eDirectory management purposes.
- ◆ **Boolean**

Used by attributes whose values are True (represented as 1) or False (represented as 0). The single-valued flag is set for this syntax type.
- ◆ **Case Exact String**

Used by attributes whose values are Unicode strings that are case sensitive in comparison operations. Two Case Exact Strings match when they are of the same length and their corresponding characters, including case, are identical.
- ◆ **Case Ignore List**

Used by attributes whose values are ordered sequences of Unicode strings that are not case sensitive in comparisons operations. Two Case Ignore Lists match if the number of strings in each is the same and all corresponding strings match (that is, they are the same length and their corresponding characters are identical).
- ◆ **Case Ignore String**

Used by attributes whose values are Unicode strings that are not case sensitive in comparison operations. Two Case Ignore Strings match when they are of the same length and their corresponding characters are identical in all respects except that of case.
- ◆ **Class Name**

Used by attributes whose values are object class names. Two Class Names match when they are of the same length and their corresponding characters are identical in all respects except that of case.
- ◆ **Counter**

Used by attributes whose values are incrementally modified numeric signed integers. Any attribute defined using Counter is a single-valued attribute. This syntax differs from Integer in that any value added to an attribute of this syntax is arithmetically added to the total, and any value deleted is arithmetically subtracted from the total.
- ◆ **Distinguished Name**

Used by attributes whose values are the names of objects in the eDirectory tree. Distinguished Names (DN) are not case sensitive, even if one of the naming attributes is case sensitive.
- ◆ **E-mail Address**

Used by attributes whose values are strings of binary information. eDirectory makes no assumption about the internal structure of the content of this syntax.
- ◆ **Facsimile Telephone Number**

Specifies a string that complies with the E.123 standard for storing international telephone numbers and an optional bit string formatted according to recommendation T.20. Facsimile Telephone Number values match when they are of the same length and their corresponding

characters are identical, except that all spaces and hyphen characters are ignored during comparison.

- ◆ Hold

Used by attributes that are accounting quantities, whose values are signed integers. This syntax is an accounting quantity (which is an amount tentatively held against a subject's credit limit, pending completion of a transaction). The hold amount is treated similarly to the Counter syntax, with new values added to or subtracted from the base total. If the evaluated hold amount goes to 0, the Hold record is deleted.

- ◆ Integer

Used by attributes represented as signed numeric values. Two Integer values match if they are identical. The comparison for ordering uses signed integer rules.

- ◆ Interval

Used by attributes whose values are signed numeric integers and represent intervals of time. The Interval syntax uses the same representation as the Integer syntax. The Interval value is the number of seconds in a time interval.

- ◆ Net Address

Represents a network layer address in the server environment. The address is in binary format. For two values of Net Address to match, the type, length, and value of the address must match.

- ◆ Numeric String

Used by attributes whose values are numerical strings as defined in the CCITT X.208 definition of Numeric String. For two Numeric Strings to match, the strings must be the same length and their corresponding characters must be identical. Digits (0...9) and space characters are the only valid characters in the numeric string character set.

- ◆ Object ACL

Used by attributes whose values represent Access Control List (ACL) entries. An Object ACL value can protect either an object or an attribute.

- ◆ Octet List

Describes an ordered sequence of strings of binary information or Octet String. An Octet List matches a stored list if it is a subset of the stored list. For two Octet Lists to match, they must be the same length, and the corresponding bit sequence (octet) must be identical.

- ◆ Octet String

Used by attributes whose values are strings of binary information not interpreted by eDirectory. These octet strings are non-Unicode strings. For two octet strings to match, they must be the same length, and the corresponding bit sequence (octet) must be identical.

- ◆ Path

Attributes that represent a file system path contain all the information to locate a file on a server. Two paths match when they are of the same length and their corresponding characters, including case, are identical.

- ◆ Postal Address

Used by attributes whose values are Unicode strings of postal addresses. An attribute value for Postal Address is typically composed of selected attributes from the MHS Unformatted Postal O/R Address Specification version 1 according to recommendation F.401. The value is limited to six lines of 30 characters each, including a postal country name. Two postal

addresses match if the number of strings in each is the same and all corresponding strings match (that is, they are the same length and their corresponding characters are identical).

- ◆ Printable String

Used by attributes whose values are printable strings, as defined in CCITT X.208. The printable character set consists of the following:

- ◆ Uppercase and lowercase alphabetic characters
- ◆ Digits (0...9)
- ◆ Space character
- ◆ Apostrophe (')
- ◆ Left and right parentheses ()
- ◆ Plus sign (+)
- ◆ Comma (,)
- ◆ Hyphen (-)
- ◆ Period (.)
- ◆ Forward slash (/)
- ◆ Colon (:)
- ◆ Equals sign (=)
- ◆ Question mark (?)

Two printable strings are equal when they are the same length and their corresponding characters are the same. Case is significant.

- ◆ Replica Pointer

Used by attributes whose values represent partition replicas. A partition of an eDirectory tree can have replicas on different servers. The syntax has six components:

- ◆ Server Name
- ◆ Replica Type (master, secondary, read-only, subordinate reference)
- ◆ Replica Number
- ◆ Replica Root ID
- ◆ Number of Address
- ◆ Address Record

- ◆ Stream

Represents arbitrary binary information. The Stream syntax provides a way to make an eDirectory attribute out of a file on a file server. Login scripts and other stream attributes use this syntax. The data stored in a stream file has no syntax enforcement of any kind. It is completely arbitrary data, defined by the application that created and uses it.

- ◆ Telephone Number

Used by attributes whose values are telephone numbers. The length of telephone number strings must be between 1 and 32 characters. Two telephone numbers match when they are of the same length and their corresponding characters are identical, except that all spaces and hyphen characters are ignored during comparison.

- ◆ Time

Used by attributes whose values are unsigned integers and represent time expressed in seconds.
- ◆ Timestamp

Used by attributes whose values mark the time when a particular event occurred. When a significant event occurs, an eDirectory server mints a new Timestamp value and associates the value with the event. Every Timestamp value is unique within an eDirectory partition. This provides a total ordering of events occurring on all servers holding replicas of a partition.
- ◆ Typed Name

Used by attributes whose values represent a level and an interval associated with an object. This syntax names an eDirectory object and attaches two numeric values to it:

 - ◆ Level of the attribute indicative of its priority
 - ◆ Interval representing the number of seconds between certain events or the frequency of the reference
- ◆ Unknown

Used by attributes whose attribute definition has been deleted from the schema. This syntax represents strings of binary information.

Understanding Mandatory and Optional Attributes

Every object has a schema class that has been defined for that type of object, and a class is a group of attributes organized in a meaningful way. Some of these attributes are mandatory and some are optional.

Mandatory Attributes

A mandatory attribute is one that must be filled in when an object is being created. For example, if a new user is being created using the User class, which has the employee number as a mandatory attribute, then the new User object cannot be created without providing the employee number.

Optional Attributes

An optional attribute is one that can be filled in if desired but can be left without content. For example, if a new User object is being created using the User class, which has Other Names as an optional attribute, then the new User object can be created with or without data provided for that attribute, depending on whether the new user is known by other names.

An exception to the rule is when an optional attribute is used for naming, the attribute then becomes mandatory.

Sample Schema

Figure 12 on page 46 is a sample of part of a schema, which might be similar to your base schema. This figure shows information on the Organization class. Most of the information displayed on this screen was specified when the class was created. Some of the optional attributes were added later.



This icon is assigned to all classes and attributes that are extensions to the base schema.

Figure 12 Class Information Page in iManager

Class flags:

Container	▲
Effective	▼

[Add a new attribute](#)
[View superclass](#)

Can Be Contained By:

[Nothing]	▲
Country	
domain	▼

Attribute:

teletexTerminalIdentifier	<input type="checkbox"/>	<input type="checkbox"/>	▲
telexNumber	<input type="checkbox"/>	<input type="checkbox"/>	
x121Address	<input type="checkbox"/>	<input type="checkbox"/>	
Account Balance	<input type="checkbox"/>	<input type="checkbox"/>	
Allow Unlimited Credit	<input type="checkbox"/>	<input type="checkbox"/>	▼

ASN1 ID:

2.5.6.4

Designing the Schema

Designing your schema initially can save you time and effort in the long run. You can view the base schema and determine if it will meet your needs or if modifications are required. If changes are needed, use Schema Manager to extend the schema. See [“Extending the Schema” on page 104](#) and [“Viewing the Schema” on page 107](#) for more information.

Partitions

If you have slow or unreliable WAN links or your directory has so many objects that the server is overwhelmed and access is slow, you should consider partitioning the directory. For a complete discussion of partitions, see [Chapter 5, “Managing Partitions and Replicas,” on page 113](#).

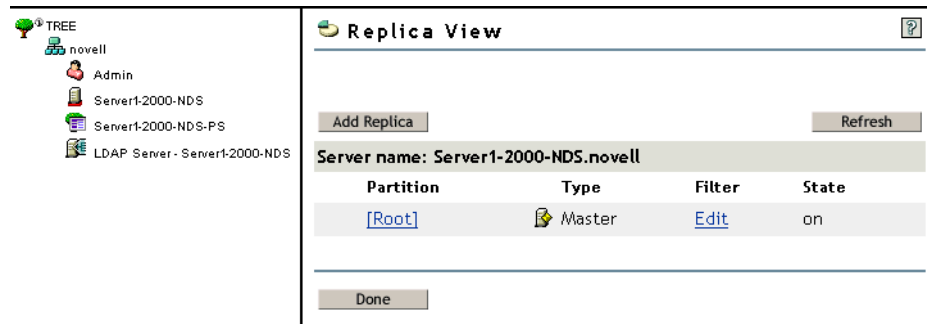
Partitioning allows you to take part of the directory off one server and put it on another server.

A partition is a logical division of the eDirectory database. A directory partition forms a distinct unit of data in the tree that stores directory information.

Each directory partition consists of a set of container objects, all the objects contained in them, and data about those objects. eDirectory partitions don't include any information about the file system or the directories and files contained there.

Partitioning is done with Novell iManager. Partitions are identified in iManager by the following partition icon ().

Figure 13 Replica View for a Server



In the above example, the partition icon is next to the Tree object. This means it is the top-most container in the partition. No partitions are shown by any other containers, so this partition is the only one.

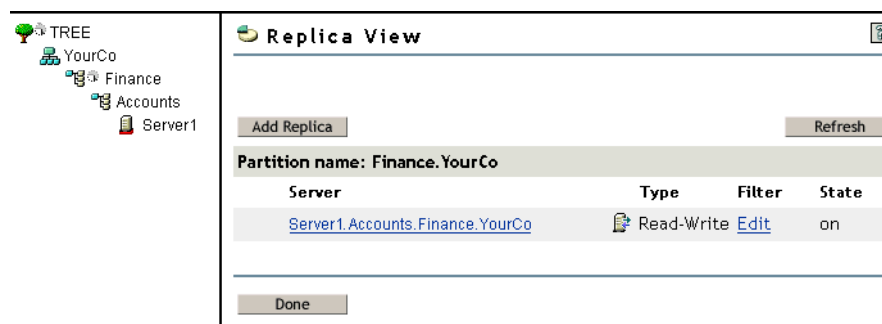
This is the default partitioning for eDirectory, keeping the entire directory together in one partition.

Notice in the example that the Replica View for Server1 is displayed. When you display the Replica View for a server in iManager, any replicas held on that server are shown on the right. In this case, Server1 holds a replica of the only partition. For more information, see [“Replicas” on page 49](#) and [“Viewing Replicas on an eDirectory Server” on page 121](#).

Partitions

Partitions are named by their topmost container. In [Figure 14](#) there are two partitions, named Tree and Finance. Finance is called a child partition of Tree, because it was split off from Tree. Tree is called the parent partition of Finance.

Figure 14 Replica View for a Partition



You might create such a partition because the directory has so many objects that the server is overwhelmed and access to eDirectory is slow. Creating the new partition allows you to split the database and pass the objects in that branch to a different server.

The example above shows the Replica View for the Finance partition. When you display the Replica View for a partition in iManager, any servers holding a replica of that partition are shown on the right. In this case, Server1 holds a Read-Write replica of the Finance partition. For more information, see [“Viewing a Partition's Replicas” on page 123](#).

Distributing Replicas for Performance

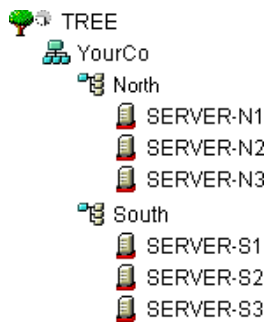
In the preceding example, suppose that Server1 holds replicas of both the Tree partition and the Finance partition. At this point, you haven't gained any performance advantage from eDirectory because Server1 still holds the entire directory (replicas of both partitions).

To gain the desired performance advantage, you need to move one of the replicas to a different server. For instance, if you move the Tree partition to Server2, then Server2 holds all objects in the Tree and YourCo containers. Server1 holds only objects in the Finance and Accounts containers. The load on both Server1 and Server2 is less than it would be with no partitioning.

Partitions and WAN Links

Suppose your network spans two sites, a North site and a South Site, separated by a WAN link. Three servers are at each site.

Figure 15 Sample eDirectory Containers

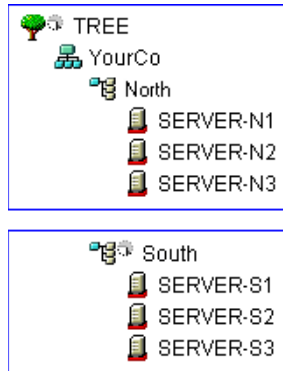


eDirectory performs faster and more reliably in this scenario if the directory is divided in two partitions.

With a single partition, the replicas are either kept at one site or distributed between the two sites. This proves unwieldy for two reasons:

- ◆ If all replicas are kept on servers at the North site, for example, users at the South site encounter delays when logging in or accessing resources. If the link goes down, users at the South site can't log in or access resources at all.
- ◆ If replicas are distributed between sites, users can access the directory locally. However, server-to-server synchronization of replicas happens over the WAN link, so there can be eDirectory errors if the link is unreliable. Any changes to the directory are slow to propagate across the WAN link.

The two-partition solution shown in [Figure 16 on page 49](#) solves performance and reliability problems over the WAN link.

Figure 16 Sample Partitions

Replicas of the Tree partition are kept on servers at the North site. Replicas of the South partition are kept on servers at the South site, as shown in [Figure 17](#).

Figure 17 Sample Partitions, Servers, and Replicas

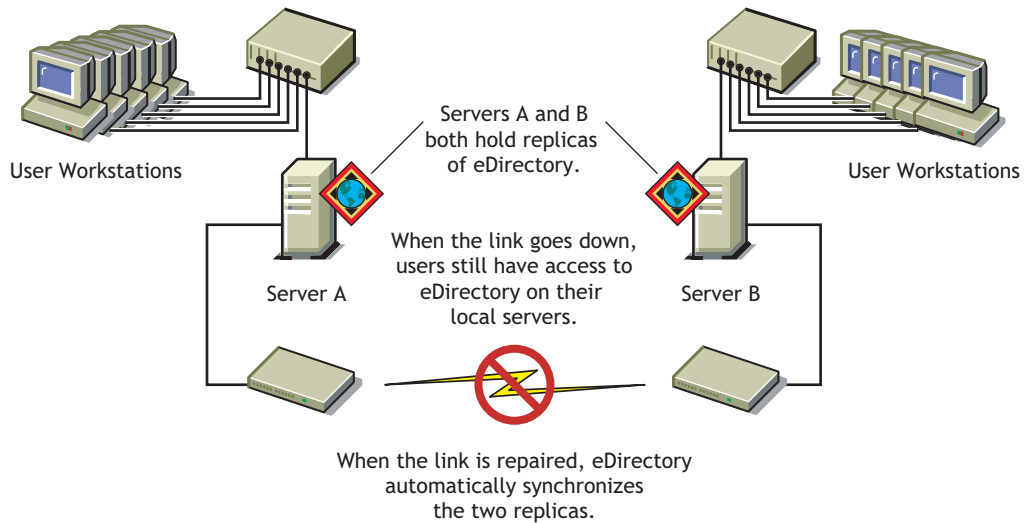
Partition	Server	Replica Type
TREE	SERVER-N1	Master
	SERVER-N2	Read/write
	SERVER-N3	Read/write
South	SERVER-S1	Master
	SERVER-S2	Read/write
	SERVER-S3	Read/write

For each site, the objects that represent local resources are kept locally. Synchronization traffic among servers also happens locally over the LAN, rather than over the slow, unreliable WAN link.

eDirectory traffic is generated over the WAN link, however, when a user or administrator accesses objects at a different site.

Replicas

A replica is a copy or an instance of a user-defined partition that is distributed to an eDirectory server. If you have more than one eDirectory server on your network, you can keep multiple replicas (copies) of the directory. That way, if one server or a network link to it fails, users can still log in and use the remaining network resources (see [Figure 18 on page 50](#)).

Figure 18 eDirectory Replicas

Each server can store more than 65,000 eDirectory replicas; however, only one replica of the same user-defined partition can exist on the same server. For a complete discussion of replicas, see [Chapter 5, “Managing Partitions and Replicas,” on page 113](#).

We recommend that you keep three replicas for fault tolerance of eDirectory (assuming you have three eDirectory servers to store them on). A single server can hold replicas of multiple partitions.

A replica server is a dedicated server that stores only eDirectory replicas. This type of server is sometimes referred to as a DSMASTER server. This configuration is popular with some companies that use many single-server remote offices. The replica server provides a place for you to store additional replicas for the partition of a remote office location. (It can also be a part of your disaster recovery planning, as described in [“Using DSMASTER Servers as Part of Disaster Recovery Planning” on page 377](#).)

eDirectory replication does not provide fault tolerance for the server file system. Only information about eDirectory objects is replicated. You can get fault tolerance for file systems by using the Transaction Tracking System™ (TTS™), disk mirroring/duplexing, RAID, or Novell Replication Services (NRS).

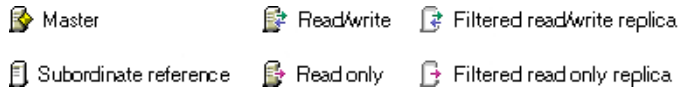
A master or read/write replica is required on NetWare servers that provide bindery services.

If users regularly access eDirectory information across a WAN link, you can decrease access time and WAN traffic by placing a replica containing the needed information on a server that users can access locally.

The same is true to a lesser extent on a LAN. Distributing replicas among servers on the network means information is usually retrieved from the nearest available server.

Replica Types

eDirectory supports the types of replicas shown in the following figure:

Figure 19 Replica Types

- ◆ “Master Replica” on page 51
- ◆ “Read/Write Replica” on page 51
- ◆ “Read-Only Replica” on page 52
- ◆ “Filtered Read/Write Replica” on page 52
- ◆ “Filtered Read-Only Replica” on page 52
- ◆ “Subordinate Reference Replica” on page 52

Master Replica



The master replica is a writable replica type used to initiate changes to an object or partition. The master replica manages the following types of eDirectory partition operations:

- ◆ Adding replicas to servers
- ◆ Removing replicas from servers
- ◆ Creating new partitions in the eDirectory tree
- ◆ Removing existing partitions from the eDirectory tree
- ◆ Relocating a partition in the eDirectory tree

The master replica is also used to perform the following types of eDirectory object operations:

- ◆ Adding new objects to the eDirectory tree
- ◆ Removing, renaming, or relocating existing objects in the eDirectory tree
- ◆ Authenticating objects to the eDirectory tree
- ◆ Adding new object attributes to the eDirectory tree
- ◆ Modifying or removing existing attributes

By default, the first eDirectory server on your network holds the master replica. There is only one master replica for each partition at a time. If other replicas are created, they are read/write replicas by default.

If you're going to bring down the server holding a master replica for longer than a day or two, you can make one of the read/write replicas the master. The original master replica automatically becomes read/write.

A master replica must be available on the network for eDirectory to perform operations such as creating a new replica or creating a new partition.

Read/Write Replica



eDirectory can access and change object information in a read/write replica as well as the master replica. All changes are then automatically propagated to all replicas.

If eDirectory responds slowly to users because of delays in the network infrastructure (such as slow WAN links or busy routers), you can create a read/write replica closer to the users who need

it. You can have as many read/write replicas as you have servers to hold them, although more replicas cause more traffic to keep them synchronized.

Read-Only Replica



The read-only replica is a readable replica type used to read information about all objects in a partition's boundaries. Read-only replicas receive synchronization updates from master and read/write replicas but don't receive changes directly from clients.

This replica type is not able to provide bindery emulation, but it does provide eDirectory tree fault tolerance. If the master replica and all read/write replicas are destroyed or damaged, the read-only replica can be promoted to become the new master replica.

It also provides NDS Object Reads, Fault Tolerance (contains all objects within the Partition boundaries), and NDS Directory Tree Connectivity (contains the Partition Root object).

A read-only replica should never be used to establish a security policy within a tree to restrict the modification of objects, because the client can always access a read/write replica and still make modifications. There are other mechanisms that exist in the directory for this purpose, such as using an Inherited Rights Filter. For more information, see [“Inherited Rights Filter \(IRF\)” on page 60](#).

Filtered Read/Write Replica



Filtered read/write replicas contain a filtered set of objects or object classes along with a filtered set of attributes and values for those objects. The contents are limited to the types of eDirectory objects and properties specific in the host server's replication filter. Users can read and modify the contents of the replica, and eDirectory can access and change selected object information. The selected changes are then automatically propagated to all replicas.

With filtered replicas, you can have only one filter per server. This means that any filter defined for a server applies to all filtered replicas on that server. You can, however, have as many filtered replicas as you have servers to hold them, although more replicas cause more traffic to keep them synchronized.

For more information, see [“Filtered Replicas” on page 53](#).

Filtered Read-Only Replica



Filtered read-only replicas contain a filtered set of objects or object classes along with a filtered set of attributes and values for those objects. They receive synchronization updates from master and read/write replicas but don't receive changes directly from clients. Users can read but not modify the contents of the replica. The contents are limited to the types of eDirectory objects and properties specific in the host server's replication filter.

For more information, see [“Filtered Replicas” on page 53](#).

Subordinate Reference Replica

Subordinate reference replicas are system-generated replicas that don't contain all the object data of a master or a read/write replica. Subordinate reference replicas, therefore, don't provide fault tolerance. They are internal pointers that are generated to contain enough information for eDirectory to resolve object names across partition boundaries.

You can't delete a subordinate reference replica; eDirectory deletes it automatically when it is not needed. Subordinate reference replicas are created only on servers that hold a replica of a parent partition but no replicas of its child partitions.

If a replica of the child partition is copied to a server holding the replica of the parent, the subordinate reference replica is automatically deleted.

Filtered Replicas

Filtered replicas contain a filtered set of objects or object classes along with a filtered set of attributes and values for those objects. For example, you might want to create a set of filtered replicas on a single server that contains only User objects from various partitions in the eDirectory tree. In addition to this, you can choose to include only a subset of the User objects' data (for example, Given Name, Surname, and Telephone Number).

A filtered replica can construct a view of eDirectory data onto a single server. To do this, filtered replicas let you create a scope and a filter. This results in an eDirectory server that can house a well-defined data set from many partitions in the tree.

The descriptions of the server's scope and data filters are stored in eDirectory and can be managed through the Server object in iManager.

A server hosting one or more filtered replicas has only a single replication filter. Therefore, all filtered replicas on the server contain the same subset of information from their respective partitions. The master partition replica of a filtered replica must be hosted on an eDirectory server running eDirectory 8.5 or later.

Filtered replicas can

- ◆ Reduce synchronization traffic to the server by reducing the amount of data that must be replicated from other servers.
- ◆ Reduce the number of events that must be filtered by DirXML[®].

For more information on DirXML, see the *DirXML Administration Guide* (<http://www.novell.com/documentation/lg/dirxml11a/index.html>).

- ◆ Reduce the size of the directory database.

Each replica adds to the size of the database. By creating a filtered replica that contains only specific classes (instead of creating a full replica), you can reduce the size of your local database.

For example, if your tree contains 10,000 objects but only a small percentage of those objects are Users, you could create a filtered replica containing only the User objects instead of a full replica containing all 10,000 objects.

Other than the ability to filter data stored in a local database, the filtered replica is like a normal eDirectory replica and it can be changed back to a full replica at any time.

NOTE: Filtered replicas by default will have the Organization and the Organizational Unit as mandatory filters.

For more information on setting up and managing filtered replicas, see “[Setting Up and Managing Filtered Replicas](#)” on page 120.

NetWare Bindery Emulation

Many applications, such as print servers and backup software, were written for NetWare versions earlier than NetWare 4. These applications used the NetWare bindery instead of eDirectory for network access and object manipulation.

The bindery is a flat database of objects such as Users, Groups, and Volumes known to a given server. The bindery is server specific and server centric.

Older NetWare client software (such as the NETX bindery shell) used a bindery login procedure in which a user logged in to a specific server only. Access to multiple servers required multiple logins using multiple user accounts.

eDirectory allows applications written for a bindery to function using bindery services. Bindery services allows you to set an eDirectory context or a number of contexts (up to 12) as an eDirectory server's virtual bindery. The context you set is called the server's bindery context.

Following are some important facts about bindery services:

- ◆ To use bindery services, you must set a bindery context for the eDirectory server.
- ◆ Not all objects map to bindery objects. Many objects, such as Alias objects, do not have a bindery equivalent.
- ◆ Most bindery applications have been upgraded to work with eDirectory. Check with your application vendor to get the newest version.
- ◆ Each eDirectory server with a bindery context must hold a master or read/write replica of the partition that includes the bindery context.

Server Synchronization in the Replica Ring

When multiple servers hold replicas of the same partition, those servers are considered a replica ring. eDirectory automatically keeps those servers synchronized, so the object data is consistent on all replicas.

The following eDirectory processes keep servers in the replica ring synchronized:

- ◆ Replica synchronization
For more information on replica synchronization, see [“Administering Replicas” on page 117](#).
- ◆ Schema synchronization
- ◆ Limber
- ◆ Backlink
For more information on Backlink, see [“Forcing the Backlink Process to Run”](#) in the *Novell eDirectory 8.7.3 Installation Guide*.
- ◆ Connection management

Access to Resources

eDirectory provides a basic level of network access security through default rights. You can provide additional access control by completing the tasks outlined below.

- ◆ Assigning rights

Each time a user attempts to access a network resource, the system calculates the user's effective rights to that resource. To ensure that users have the appropriate effective rights to resources, you can make explicit trustee assignments, grant security equivalences, and filter inherited rights.

To simplify the assignment of rights, you can create Group and Organizational Role objects, then assign users to the groups and roles.

- ◆ Adding login security

Login security is not provided by default. You can set up several optional login security measures, including login passwords, login location and time restrictions, limits on concurrent login sessions, intruder detection, and login disabling.

- ◆ Setting up role-based administration

You can set up administrators for specific object properties and grant them rights to only those properties. This allows you to create administrators with specific responsibilities that can be inheritable to subordinates of any given container object. A role-based administrator can have responsibilities over any specific properties, such as those that relate to employee information or passwords.

See [Installing RBS \(http://www.novell.com/documentation/lg/imanager20/imanager20/data/am48xf3.html\)](http://www.novell.com/documentation/lg/imanager20/imanager20/data/am48xf3.html) in the *Novell iManager 2.0.x Administration Guide* for instruction on setting up Role-Based Services.

You can also define roles in terms of the specific tasks that administrators can perform in role-based administration applications. See “[Configuring Role-Based Services](#)” on [page 97](#) for more information.

eDirectory Rights

When you create a tree, the default rights assignments give your network generalized access and security. Some of the default assignments are as follows:

- ◆ User Admin has the Supervisor right to the top of the tree, giving Admin complete control over the entire directory. Admin also has the Supervisor right to the NetWare Server object, giving complete control over any volumes on that server.
- ◆ [Public] has the Browse right to the top of the tree, giving all users the right to view any objects in the tree.
- ◆ Objects created through an upgrade process such as a NetWare migration, printing upgrade, or Windows NT user migration receive trustee assignments appropriate for most situations.

Trustee Assignments and Targets

The assignment of rights involves a trustee and a target object. The trustee represents the user or set of users that are receiving the authority. The target represents those network resources the users have authority over.

- ◆ If you make an Alias a trustee, the rights apply only to the object the alias represents. The Alias object can be an explicit target, however.
- ◆ A file or directory in the NetWare file system can also be a target, although file system rights are stored in the file system itself, not in eDirectory.

NOTE: The [Public] trustee is not an object. It is a specialized trustee that represents any network user, logged in or not, for rights assignment purposes.

eDirectory Rights Concepts

The following concepts can help you better understand eDirectory rights.

- ◆ “Object (Entry) Rights” on page 56
- ◆ “Property Rights” on page 56
- ◆ “Effective Rights” on page 57
- ◆ “How Effective Rights Are Calculated” on page 57
- ◆ “Security Equivalence” on page 59
- ◆ “Access Control List (ACL)” on page 59
- ◆ “Inherited Rights Filter (IRF)” on page 60

Object (Entry) Rights

When you make a trustee assignment, you can grant object rights and property rights. Object rights apply to manipulation of the entire object, while property rights apply only to certain object properties. An object right is described as an entry right because it provides an entry into the eDirectory database.

A description of each object right follows:

- ◆ **Supervisor** includes all rights to the object and all of its properties.
- ◆ **Browse** lets the trustee see the object in the tree. It does not include the right to see an object's properties.
- ◆ **Create** applies only when the target object is a container. It allows the trustee to create new objects below the container and also includes the Browse right.
- ◆ **Delete** lets the trustee delete the target from the directory.
- ◆ **Rename** lets the trustee change the name of the target.

Property Rights

When you make a trustee assignment, you can grant object rights and property rights. Object rights apply to manipulation of the entire object, while property rights apply only to certain object properties.

iManager gives you two options for managing property rights:

- ◆ You can manage all properties at once when the [All Attributes Rights] item is selected.
- ◆ You can manage one or more individual properties when the specific property is selected.

A description of each property right follows:

- ◆ **Supervisor** gives the trustee complete power over the property.
- ◆ **Compare** lets the trustee compare the value of a property to a given value. This right allows searching and returns only a true or false result. It does not allow the trustee to actually see the value of the property.
- ◆ **Read** lets the trustee see the values of a property. It includes the Compare right.
- ◆ **Write** lets the trustee create, change, and delete the values of a property.

- ◆ **Add Self** lets the trustee add or remove itself as a property value. It only applies to properties with object names as values, such as membership lists or Access Control Lists (ACLs).

Effective Rights

Users can receive rights in a number of ways, such as explicit trustee assignments, inheritance, and security equivalence. Rights can also be limited by Inherited Rights Filters and changed or revoked by lower trustee assignments. The net result of all these actions—the rights a user can employ—are called *effective rights*.

A user's effective rights to an object are calculated each time the user attempts an action.

How Effective Rights Are Calculated

Each time a user attempts to access a network resource, eDirectory calculates the user's effective rights to the target resource using the following process:

1. eDirectory lists the trustees whose rights are to be considered in the calculation. These include
 - ◆ The user who is attempting to access the target resource.
 - ◆ The objects that the user is security equivalent to.
2. For each trustee in the list, eDirectory determines its effective rights as follows:
 - a. eDirectory starts with the inheritable rights that the trustee has at the top of the tree.

eDirectory checks the Object Trustees (ACL) property of the Tree object for entries that list the trustee. If any are found and they are inheritable, eDirectory uses the rights specified in those entries as the initial set of effective rights for the trustee.
 - b. eDirectory moves down a level in the branch of the tree that contains the target resource.
 - c. eDirectory removes any rights that are filtered at this level.

eDirectory checks the ACL at this level for Inherited Rights Filters (IRFs) that match with the right types (object, all properties, or a specific property) of the trustee's effective rights. If any are found, eDirectory removes from the trustee's effective rights any rights that are blocked by those IRFs.

For example, if the trustee's effective rights so far include an assignment of Write All Properties, but an IRF at this level blocks Write All Properties, the system removes Write All Properties from the trustee's effective rights.
 - d. eDirectory adds any inheritable rights that are assigned at this level, overriding as needed.

eDirectory checks the ACL at this level for entries that list the trustee. If any are found, and they are inheritable, eDirectory copies the rights from those entries to the trustee's effective rights, overriding as needed.

For example, if the trustee's effective rights so far include the Create and Delete object rights but no property rights, and if the ACL at this level contains both an assignment of zero object rights and an assignment of Write all properties for this trustee, then the system replaces the trustee's existing object rights (Create and Delete) with zero rights and adds the new all property rights.
 - e. eDirectory repeats the filtering and adding steps (c and d above) at each level of the tree, including at the target resource.
 - f. eDirectory adds any noninheritable rights assigned at the target resource, overriding as needed.

eDirectory uses the same process as in Step 2d above. The resulting set of rights constitutes the effective rights for this trustee.

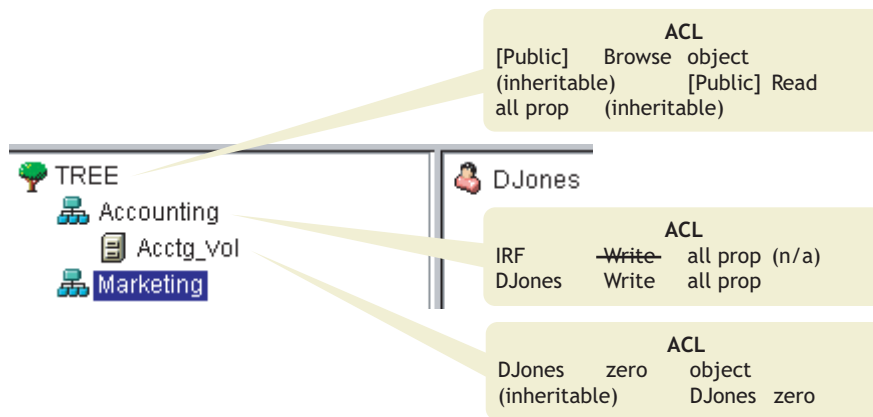
3. eDirectory combines the effective rights of all the trustees in the list as follows:
 - a. eDirectory includes every right held by any trustee in the list and excludes only those rights that are missing from every trustee in the list. eDirectory does not mix right types. For example, it does not add rights for a specific property to rights for all properties or vice versa.
 - b. eDirectory adds rights that are implied by any of the current effective rights.

The resulting set of rights constitutes the user's effective rights to the target resource.

Example

User DJones is attempting to access volume Acctg_Vol. (See [Figure 20](#).)

Figure 20 Sample Trustee Rights



The following process shows how eDirectory calculates DJones' effective rights to Acctg_Vol:

1. The trustees whose rights are to be considered in the calculation are DJones, Marketing, Tree, and [Public].

This assumes that DJones doesn't belong to any groups or roles and has not been explicitly assigned any security equivalences.

2. The effective rights for each trustee are as follows:

- ♦ DJones: Zero object, zero all properties

The assignment of zero all property rights at Acctg_Vol overrides the assignment of Write all properties at Accounting.

- ♦ Marketing: Zero all properties

The assignment of Write all properties at the top of the tree is filtered out by the IRF at Accounting.

- ♦ Tree: No rights

No rights are assigned for Tree anywhere in the pertinent branch of the tree.

- ♦ [Public]: Browse object, Read all properties

These rights are assigned at the root and aren't filtered or overridden anywhere in the pertinent branch of the tree.

3. Combining the rights from all these trustees results in the following:

DJones: Browse object, Read all properties

4. Adding the Compare all properties right that is implied by the Read all properties right, DJones has the following final effective rights to Acctg_Vol:

DJones: Browse object, Read and Compare all properties

Blocking Effective Rights

Because of the way that effective rights are calculated, it is not always obvious how to block particular rights from being effective for specific users without resorting to an IRF (an IRF blocks rights for all users).

To block particular rights from being effective for a user without using an IRF, do either of the following:

- ◆ Ensure that neither the user nor any of the objects that the user is security equivalent to ever gets assigned those rights, either at the target resource or at any level above the target resource in the tree.
- ◆ If the user or any object that the user is security equivalent to does get assigned those rights, ensure that that object also has an assignment lower in the tree that omits those rights. Do this for every trustee (associated with the user) that has the unwanted rights.

Security Equivalence

Security equivalence means having the same rights as another object. When you make one object security equivalent to another object, the rights of the second object are added to the rights of the first object when the system calculates the first object's effective rights.

For example, suppose you make User object Joe security equivalent to the Admin object. After you create the security equivalence, Joe has the same rights to the tree and file system as Admin.

There are three types of security equivalence:

- ◆ Explicit: By assignment
- ◆ Automatic: By membership in a group or role
- ◆ Implied: Equivalent to all parent containers and the [Public] trustee

Security equivalence is effective only for one step. For example, if you make a third user security equivalent to Joe in the example above, that user does not receive Admin rights.

Security equivalence is recorded in eDirectory as values in the User object's Security Equal To property.

When you add a User object as an occupant to an Organizational Role object, that User automatically becomes security equivalent to the Organizational Role object. The same is true when a User becomes a member of a Group role object.

Access Control List (ACL)

The Access Control List (ACL) is also called the Object Trustees property. Whenever you make a trustee assignment, the trustee is added as a value to the Object Trustees (ACL) property of the target.

This property has strong implications for network security for the following reasons:

- ◆ Anyone who has the Supervisor or Write right to the Object Trustees (ACL) property of an object can determine who is a trustee of that object.
- ◆ Any users with the Add Self right to the Object Trustees (ACL) property of an object can change their own rights to that object. For example, they can grant themselves the Supervisor right.

For these reasons, be careful giving Add Self rights to all properties of a container object. That assignment makes it possible for the trustee to become Supervisor of that container, all objects in it, and all objects in containers beneath it.

Inherited Rights Filter (IRF)

The Inherited Rights Filter allows you to block rights from flowing down the eDirectory Tree. For more information on configuring this filter, see [“Blocking Inherited Rights to an eDirectory Object or Property” on page 64](#).

Default Rights for a New Server

When you install a new Server object into a tree, the following trustee assignments are made:


Default Trustees	Default Rights
Admin (first eDirectory server in the tree)	Supervisor object right to the Tree object. Admin has the Supervisor object right to the NetWare Server object, which means that Admin also has the Supervisor right to the root directory of the file system of any volumes on the server.
[Public] (first eDirectory server in the tree)	Browse object right to the Tree object.
Tree	The Tree Read property right to the Host Server Name and Host Resource properties on all Volume objects. This gives all objects access to the physical volume name and physical server name.
Container objects	Read and File Scan rights to sys: \public. This allows User objects under the container to access NetWare utilities in \public.
User objects	If home directories are automatically created for users, the users have the Supervisor right to those directories.


Delegated Administration

eDirectory lets you delegate administration of a branch of the tree, revoking your own management rights to that branch. One reason for this approach is that special security requirements require a different administrator with complete control over that branch.

To delegate administration:

- 1 Grant the Supervisor object right to a container.

- 1a In Novell iManager, click the Roles and Tasks button .

- 1b** Click Rights > Modify Trustees.
 - 1c** Enter the name and context of the container object that you want to control access to, then click OK.
 - 1d** Click Assigned Rights.
 - 1e** Click the Supervisor checkbox for the properties you want.
 - 1f** Click Done, then click OK.
- 2** Create an IRF on the container that filters the Supervisor and any other rights you want blocked.
- 2a** In Novell iManager, click the Roles and Tasks button 
 - 2b** Click Rights > Modify Inherited Rights Filter.
 - 2c** Specify the name and context of the object whose inherited rights filter you want to modify, then click OK.
 - 2d** Edit the list of inherited rights filters as needed.

To edit the list of filters, you must have the Supervisor or Access Control right to the ACL property of the object. You can set filters that block inherited rights to the object as a whole, to all the properties of the object, and to individual properties.

NOTE: These filters won't block rights that are explicitly granted a trustee on this object, since such rights aren't inherited.
 - 2e** Click OK.

IMPORTANT: If you delegate administration to a User object and that object is subsequently deleted, there are no objects with rights to manage that branch.

To delegate administration of specific eDirectory properties, such as Password Management, see [“Granting Equivalence” on page 63](#).

To delegate the use of specific functions in role-based administration applications, see [“Configuring Role-Based Services” on page 97](#).

Administering Rights

- ◆ [“Assigning Rights Explicitly” on page 61](#)
- ◆ [“Granting Equivalence” on page 63](#)
- ◆ [“Blocking Inherited Rights to an eDirectory Object or Property” on page 64](#)
- ◆ [“Viewing Effective Rights to an eDirectory Object or Property” on page 65](#)


Assigning Rights Explicitly

When the default rights assignments in your eDirectory tree provide users with either too much or not enough access to resources, you can create or modify explicit rights assignments. When you create or modify a rights assignment, you start by selecting either the resource that you are controlling access to or the trustee (the eDirectory object that possesses, or will possess, the rights).

TIP: To manage users' rights collectively rather than individually, make a group, role, or container object the trustee. To restrict access to a resource globally (for all users), see [“Blocking Inherited Rights to an eDirectory Object or Property” on page 64](#).

- ◆ [“Controlling Access to Novell eDirectory by Resource” on page 62](#)
- ◆ [“Controlling Access to Novell eDirectory by Trustee” on page 62](#)

Controlling Access to Novell eDirectory by Resource

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Rights > Modify Trustees.
- 3** Specify the name and context of the eDirectory resource (object) that you want to control access to, then click OK.

Choose a container if you want to control access to all the objects below it.

- 4** Edit the list of trustees and their rights assignments as needed.
 - 4a** To modify a trustee's rights assignment, select the trustee, click Assigned Rights, modify the rights assignment as needed, then click Done.

- 4b** To add an object as a trustee, click Add Trustee, select the object, click OK, click Assigned Rights to assign the trustee's rights, then click Done.


When creating or modifying a rights assignment, you can grant or deny access to the object as a whole, to all the properties of the object, and to individual properties.

- 4c** To remove an object as a trustee, select the trustee, then click Delete Trustee.

The deleted trustee no longer has explicit rights to the object or its properties but might still have effective rights through inheritance or security equivalence.

- 5** Click OK.

Controlling Access to Novell eDirectory by Trustee

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Rights > Rights to Other Objects.
- 3** Enter the name and context of the trustee (the object that possesses, or will possess, the rights) whose rights you want to modify.
- 4** In the Context to Search From field, specify the part of the eDirectory tree to be searched for eDirectory objects that the trustee currently has rights assignments to.
- 5** Click OK.

A screen appears showing the progress of the search. When the search is done, the Rights to Other Objects page appears with the results of the search filled in.

- 6** Edit the trustee's eDirectory rights assignments as needed.
 - 6a** To add a rights assignment, click Add Object, select the object to control access to, click OK, click Assigned Rights, assign the trustee's rights, then click Done.
 - 6b** To modify a rights assignment, select the object you want to control access to, click Assigned Rights, modify the trustee's rights assignment as needed, then click Done.

When creating or modifying a rights assignment, you can grant or deny access to the object as a whole, to all the properties of the object, and to individual properties.

- 6c** To remove a rights assignment, select the object you want to control access to, then click Delete Object.

The trustee no longer has explicit rights to the object or its properties but might still have effective rights through inheritance or security equivalence.

- 7** Click OK.

Granting Equivalence

A user who is security equivalent to another eDirectory object effectively has all the rights of that object. A user is automatically security equivalent to the groups and roles that they belong to. All users are implicitly security equivalent to the [Public] trustee and to each container above their User objects in the eDirectory tree, including the Tree object. You can also explicitly grant a user security equivalence to any eDirectory object.

NOTE: The tasks in this section allow you to delegate administrative authority through eDirectory rights. If you have administration applications that use Role-Based Services (RBS) roles, you can also delegate administrative authority by assigning users membership in those roles.

- ◆ “Granting Security Equivalence by Membership” on page 63
- ◆ “Granting Security Equivalence Explicitly” on page 63
- ◆ “Setting Up an Administrator For an Object’s Specific eDirectory Properties” on page 64

Granting Security Equivalence by Membership

- 1** If you haven’t already done so, create the group or role object that you want the users to be security equivalent to.

See “Creating an Object” on page 90 for details.

- 2** Grant the group or role the eDirectory rights that you want the users to have.

See “Assigning Rights Explicitly” on page 61 for details.

- 3** Edit the membership of the group or role to include those users who need the rights of the group or role.



- ◆ For a Group object, use the Members property page.

In Novell iManager, click eDirectory Administration > Modify Object, specify the name and context of a Group object, click OK, then click the Members tab.

- ◆ For an Organizational Role object, use the Role Occupant field on the Role Occupant property page.


In Novell iManager, click eDirectory Administration > Modify Object, specify the name and context of an rbsRole object, click OK, then click Role Occupant on the General tab.

- ◆ For an rbsRole object, use the Modify iManager Members page.

In Novell iManager, click the Configure button , click Role Configuration > Modify iManager Roles, click the Modify Members button  to the left of the role you want to modify, then use the options on the Modify iManager Members page to add or remove members from a role.

- 4** Click OK.

Granting Security Equivalence Explicitly

- 1** In Novell iManager, click the Roles and Tasks button .

- 2** Click eDirectory Administration > Modify Object.

- 3** Enter the name and context of the user or object that you want the user to be security equivalent to, then click OK.

- 4** Click the Security tab, then grant the security equivalence as follows:

- ◆ If you chose a user, click Security Equal To > enter the name and context of the object that you want the user to be security equivalent to, press Enter, then click OK.

- ♦ If you chose an object that you want the user to be security equivalent to, click Security Equal to Me, enter the name and context of the user that you want the object to be security equivalent to, press Enter, then click OK.

The contents of these two property pages are synchronized by the system.


- 5 Click OK.

Setting Up an Administrator For an Object's Specific eDirectory Properties

- 1 If you haven't already done so, create the User, Group, Role, or Container object that you want to make a trustee of the object's specific properties.


If you create a container as a trustee, all objects inside and below the container will have the rights you grant. You must make the property inheritable or the container and its members will not have rights below its level.

See ["Creating an Object" on page 90](#) for information.

- 2 In Novell iManager, click the Roles and Tasks button .
- 3 Click Rights > Modify Trustees.
- 4 Specify the name and context of the highest-level container that you want the administrator to manage, then click OK.
- 5 On the Modify Trustees page, click Add Trustee, select the object that represents the administrator, then click OK.
- 6 Click Assigned Rights for the trustee you just added, then click Add Property.
- 7 Select the properties you want to add to the property list, then click OK.
- 8 For each property that the administrator will manage, assign the needed rights.
Be sure to select the Inheritable check box on each rights assignment.
- 9 Click Done, then click OK.

Blocking Inherited Rights to an eDirectory Object or Property

In eDirectory, rights assignments on containers can be inheritable or non-inheritable. In the NetWare file system, all rights assignments on folders are inheritable. In both eDirectory and NetWare, you can block such inheritance on individual subordinate items so that the rights aren't effective on those items, no matter who the trustee is. One exception is that the Supervisor right can't be blocked in the NetWare file system.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click Rights > Modify Inherited Rights Filter.
- 3 Specify the name and context of the object whose inherited rights filter you want to modify, then click OK.

This displays a list of the inherited rights filters that have already been set on the object.

- 4 On the property page, edit the list of inherited rights filters as needed.


To edit the list of filters, you must have the Supervisor or Access Control right to the ACL property of the object. You can set filters that block inherited rights to the object as a whole, to all the properties of the object, and to individual properties.

NOTE: These filters won't block rights that are explicitly granted a trustee on this object, because such rights aren't inherited.

- 5 Click OK.

Viewing Effective Rights to an eDirectory Object or Property

Effective rights are the actual rights users can exercise on specific network resources. They are calculated by eDirectory based on explicit rights assignments, inheritance, and security equivalence. You can query the system to determine a user's effective rights to any resource.

- 1 In Novell iManager, click the Roles and Tasks button 
- 2 Click Rights > View Effective Rights.
- 3 Enter the name and context of the trustee whose effective rights you want to view, then click OK.
- 4 Choose from the following options:

Option	Description
Property Name	<p>Lists the properties that the trustee has effective rights to. The properties are read from eDirectory and so are always shown in English. Each item in the list is one of the following types:</p> <p>[All Attributes Rights]—Represents all the properties of the object.</p> <p>[Entry Rights]—Represents the object as a whole. Rights to this item don't imply any property rights, except in the case of Supervisor.</p> <p>Specific properties—These are specific properties that the trustee has rights to individually. By default, only properties of this object class are listed (see below).</p>
Effective Rights	Shows the trustee's effective rights to the selected property, as calculated by eDirectory.
Show All Properties in Schema	<p>Leave this check box deselected to show only the properties of this object class.</p> <p>To show the properties of all classes defined in the eDirectory schema, select this check box. The additional properties are pertinent only if this object is a container, or if it has been extended to include the properties of an auxiliary class. The additional properties are shown without a bullet next to them.</p>

- 5 Click Done.

2 Designing Your Novell eDirectory Network

The design of Novell® eDirectory™ impacts virtually every network user and resource. A good eDirectory design can enhance the performance and value of the entire network by making the network more efficient, fault tolerant, secure, and scalable, and operable. This chapter provides suggestions for designing your eDirectory network.

- ♦ “eDirectory Design Basics” on page 67
- ♦ “Designing the eDirectory Tree” on page 68
- ♦ “Guidelines for Partitioning Your Tree” on page 74
- ♦ “Guidelines for Replicating Your Tree” on page 75
- ♦ “Planning the User Environment” on page 78
- ♦ “Designing eDirectory for e-Business” on page 79
- ♦ “Understanding the Novell Certificate Server” on page 80
- ♦ “Synchronizing Network Time” on page 84

eDirectory Design Basics

An efficient eDirectory design is based on the network layout, organizational structure of the company, and proper preparation.

If you are designing eDirectory for e-business, refer to “[Designing eDirectory for e-Business](#)” on page 79.

Network Layout

The network layout is the physical setup of your network. To develop an efficient eDirectory design, you need to be aware of the following:

- ♦ WAN links
- ♦ Users that need remote access
- ♦ Network resources (such as number of servers)
- ♦ Network conditions (such as frequent power outages)
- ♦ Anticipated changes to the network layout

Organizational Structure

The organizational structure of the company will influence the eDirectory design. To develop an efficient eDirectory design you need

- ◆ The organizational chart and an understanding of how the company operates.
- ◆ Personnel who have the skills needed to complete the design and implementation of your eDirectory tree.

You will need to identify personnel who can do the following:

- ◆ Maintain the focus and schedule of the eDirectory design
- ◆ Understand eDirectory design, design standards, and security
- ◆ Understand and maintain the physical network structure
- ◆ Manage the internetwork backbone, telecommunications, WAN design, and router placement

Preparing for eDirectory Design

Before you actually create the eDirectory design, you should

- ◆ Set realistic expectations concerning scope and schedule.
- ◆ Notify all users who will be affected by the design of your implementation of eDirectory.
- ◆ Review the information in “[Network Layout](#)” on page 67 and “[Organizational Structure](#)” on page 68.

Designing the eDirectory Tree

Designing the eDirectory tree is the most important procedure in the design and implementation of a network. The design consists of the following tasks:

- ◆ “[Creating a Naming Standards Document](#)” on page 68
- ◆ “[Designing the Upper Layers of the Tree](#)” on page 71
- ◆ “[Designing the Lower Layers of the Tree](#)” on page 73

Creating a Naming Standards Document

Using standard names such as object names makes your network more intuitive to both users and administrators. Written standards can also specify how administrators set other property values, such as telephone numbers and addresses.

Searching and browsing the directory rely greatly on the consistency of naming or property values.

The use of standard names also makes it easier for DirXML[®] to move data between eDirectory and other applications. For more information on DirXML, see the [DirXML Administration Guide](http://www.novell.com/documentation/lg/dirxml11a/index.html) (<http://www.novell.com/documentation/lg/dirxml11a/index.html>).

Naming Conventions

Tree Name

When creating a new tree, the tree name must be

- ◆ Unique in the entire network.
- ◆ Should be 2 to 32 characters long.
- ◆ Should only contain characters such as letters A-Z, numbers 0-9, hyphens (-), and underscores (_).

Objects

- ◆ The name must be unique in the container. For example, Debra Jones and Daniel Jones cannot both be named DJONES if they are in the same container.
- ◆ Special characters are allowed. However, plus signs (+), equals signs (=), and periods (.) must be preceded by a backslash (\) if used. Additional naming conventions apply to Server and Country objects, as well as to bindery services and multilingual environments.
- ◆ Uppercase and lowercase letters, as well as underscores and spaces, are displayed as you first entered them, but they aren't distinguished. For example, Manager_Profile and MANAGER PROFILE are considered identical.
- ◆ If you use spaces, you must enclose the name in quotes when entering it on the command line or in login scripts.

Server Objects

- ◆ Server objects are automatically created when you install new servers.
- ◆ You can create additional Server objects for existing NetWare[®] and NT servers and for eDirectory servers in other trees, but they are all treated as bindery objects.
- ◆ When creating a Server object, the name must match the physical server name, which
 - ◆ Is unique in the entire network.
 - ◆ Is from 2 to 47 characters long.
 - ◆ Contains only letters A-Z, numbers 0-9, hyphens (-), periods (.), and underscores (_).
 - ◆ Does not use a period as the first character.
- ◆ Once named, the Server object cannot be renamed in Novell iManager. If you rename it at the server, the new name automatically appears in iManager.

Country Objects

Country objects should follow the standard two-letter ISO country code.

For more information, see the [ISO 3166 Code Lists \(http://www.iso.ch/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/list-en1.html\)](http://www.iso.ch/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/list-en1.html).

Bindery Objects

If the object is accessed from NetWare 2 or NetWare 3 through bindery services, the following restrictions apply:

- ◆ Spaces in the name are replaced with underscores
- ◆ Names are truncated to 47 characters
- ◆ The following characters are not allowed: slash (/), backslash (\), colon (:), comma (,), asterisk (*), and question mark (?)

IMPORTANT: Bindery emulation is not supported on Linux, Solaris, AIX, or HP-UX platforms.

Multilingual Considerations

If you have workstations running in different languages, you might want to limit object names to characters that are viewable on all the workstations. For example, a name entered in Japanese cannot contain characters that aren't viewable in Western languages.

HP-UX supports only English language.

IMPORTANT: The Tree name should always be specified in English.

Sample Standards Document

The following is a sample document containing standards for some of the most frequently used properties. You need to have standards only for those properties you use. Distribute the standards document to all administrators responsible for creating or modifying objects.

Object Class Property	Standard	Examples	Rationale
User Login name	First initial, middle initial (if applicable), and last name (all lowercase). Eight characters maximum. All common names are unique in the company.	msmith, bjohnson	Using unique names company-wide is not required by eDirectory but helps avoid conflicts within the same context (or bindery context).
User Last name	Last name (normal capitalization).	Smith	Used for generating mailing labels.
Telephone and fax numbers	Numbers separated by hyphens.	US: 123-456-7890 Other: 44-344-123456	Used by autodialing software.
Multiple classes Location	Two-letter location code (uppercase), hyphen, mail stop.	BA-C23	Used by interoffice mail carriers.
Organization Name	The name of your company for all trees.	YourCo	If you have separate trees, a standard Organization name allows for future merging of trees.
Organizational Unit Name (based on location)	Two- or three-letter location code, all uppercase.	ATL, CHI, CUP, LA, BAT, BOS, DAL	Short, standard names are used for efficient searching.
Organizational Unit Name (based on department)	Department name or abbreviation.	Sales, Eng	Short, standard names make it easy to identify which department the container is servicing.
Group Name	Descriptive name.	Project Managers	Avoid extremely long names; some utilities will not display them.
Directory Map Name	Contents of the directory indicated by the Directory Map.	DOSAPPS	Short, standard names make it easy to identify which department the container is servicing.
Profile Name	Purpose of the profile.	MobileUser	Short, standard names make it easy to identify which department the container is servicing.
Server Name	SERV, hyphen, department, hyphen, unique number.	SERV-Eng-1	eDirectory requires server names to be unique in the tree.

Designing the Upper Layers of the Tree

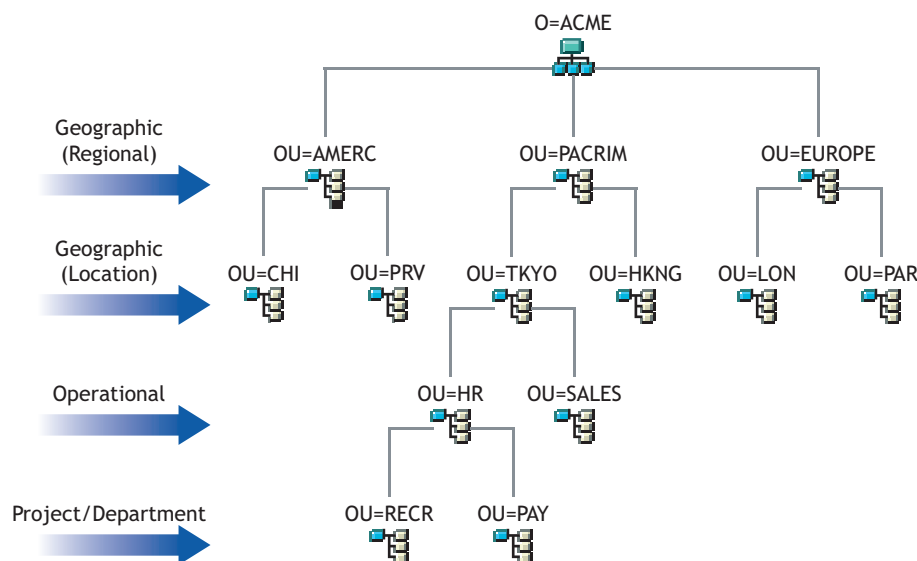
You should carefully design the upper layers of the tree because changes to the upper layers affect the rest of the tree, especially if your organization has WAN links. You want to design the top of the tree so that few changes will be necessary.

Use the following eDirectory design rules to create your eDirectory tree:

- ◆ Use a pyramid design.
- ◆ Use one eDirectory tree with a unique name.
- ◆ Create a single Organization object.
- ◆ Create first-level Organizational Units that represent the physical network infrastructure.

Figure 21 depicts the eDirectory design rules.

Figure 21 eDirectory Design Rules



To create the upper layers of the tree, see [“Creating an Object”](#) on page 90 and [“Modifying an Object’s Properties”](#) on page 90.

Using a Pyramid Design

With a pyramid-designed eDirectory, managing, initiating changes to large groups, and creating logical partitions are easier.

The alternative to the pyramid design is a flat tree that places all objects in the top layers of the tree. eDirectory can support a flat tree design; however, a flat tree design can be more difficult to manage and partition.

Using One eDirectory Tree with a Unique Name

A single tree works best for most organizations. By default, one tree is created. With one tree you have single-user identity on the network, simpler administration of security, and single point of management.

This recommendation for a single tree for business use does not preclude additional trees for testing and development.

Some organizations, however, might need multiple trees because of legal, political, or corporate issues. For example, an organization consisting of several autonomous organizations might need to create several trees. If your organization needs multiple trees, consider using DirXML to simplify management. For more information on DirXML, see the *DirXML Administration Guide* (<http://www.novell.com/documentation/lg/dirxml11a/index.html>).

NOTE: HP-UX does not support DirXML.

When you name the tree, use a unique name that will not conflict with other tree names. Use a name that is short and descriptive, such as EDL-TREE. Maximum number of characters allowed for naming a tree is 32 characters.

If two trees have the same name and are located on the same network, you might encounter the following problems:

- ◆ Updates going to the wrong tree
- ◆ Resources disappearing
- ◆ Rights disappearing
- ◆ Corruption

You can change the tree name using the DSMERGE utility, but do so with caution. A tree name change impacts the network because you need to reconfigure the clients to use the new tree name.

Creating a Single Organization Object

Generally, an eDirectory tree should have one Organization object. By default, a single Organization object is created and named after your company. This allows you to configure changes that apply to the whole company from a single location in the tree.

For example, you can use ZENworks[®] to create a Workstation Import Policy object in the Organization object. In this policy, which affects the whole organization, you define how Workstation objects are named when created in eDirectory.

In the Organization container, the following objects are created:

- ◆ Admin
- ◆ Server
- ◆ Volume

Networks with only a Windows, Linux, Solaris, AIX, or HP-UX server running eDirectory have no Volume objects.

You might want to create multiple Organization objects if your company has the following needs:

- ◆ It comprises multiple companies that do not share the same network.
- ◆ It needs to represent separate business units or organizations.
- ◆ It has a policy or other internal guidelines that dictate that organizations remain separate.

Creating Organizational Units That Represent the Physical Network

First-level Organizational Unit design is important because it affects the partitioning and efficiency of eDirectory.

For networks that span more than one building or location using either a LAN or a WAN, the first-level Organizational Unit object design should be based on location. This allows you to partition eDirectory in a way that keeps all objects in a partition at one location. It also provides a natural place to make security and administrator assignments for each location.

Designing the Lower Layers of the Tree

You should design the lower layers of the tree based on the organization of network resources. You have more freedom in designing the lower layers of an eDirectory tree than the upper layers because lower-layer design affects only objects at the same location.

To create the lower layers of the tree, see “[Creating an Object](#)” on page 90 and “[Modifying an Object’s Properties](#)” on page 90.

Determining Container, Tree, and Database Size

The number of lower-level container objects you create depends on the total number of objects in your tree and your disk space and disk I/O speed limitations. eDirectory has been tested with over 1 billion objects in a single eDirectory tree, so the only real limitations are disk space, disk I/O speed, and RAM to maintain performance. Keep in mind that the impact of replication on a large tree is significant.

A typical object in eDirectory is 3 to 5 KB in size. Using this object size, you can quickly calculate disk space requirements for the number of objects you have or need. Keep in mind that the object size will grow depending upon how many attributes are completed with data and what the data is. If objects will hold binary large object (BLOB) data such as pictures, sounds, or biometrics, the object size will subsequently grow.

The larger the partitions, the slower the replication cycles. If you are using products that require the use of eDirectory, such as ZENworks and DNS/DHCP services, the eDirectory objects created by these products will affect the size of the containers they are located in. You might consider placing objects that are for administration purposes only, such as DNS/DHCP, in their own partition so user access is not affected with slower replication. Also, managing partitions and replicas will be easier.

If you are interested, you can easily determine the size of your eDirectory database or the Directory Information Base (DIB) Set.

- ◆ For NetWare, download toolbox.nlm from the [Novell Support Web site \(http://support.novell.com\)](http://support.novell.com) to see the sys:_netware directory on your server.
- ◆ For Windows, look at the DIB Set at \novell\nds\dibfiles.
- ◆ For Linux, Solaris, AIX, or HP-UX, look at the DIB Set in the directory you specified during installation.

Deciding Which Containers to Create

In general, create containers for objects that have access needs in common with other eDirectory objects. This lets you service many users with one trustee assignment or login script. You can create containers specifically to make container login scripts more effective, or you can place two departments in one container to make login script maintenance more feasible.

Keep users close to the resources they need to limit traffic over the network. For example, people who work in the same department generally work near each other. They usually need access to the same file system and they print to the same printers.

Exceptions to general workgroup boundaries are not hard to manage. If two workgroups use a common printer, for instance, you can create an Alias object to the printer in one of the workgroups. You can create Group objects to manage some User objects within a workgroup or User objects across multiple workgroups. You can create Profile objects for subsets of users with unique login script requirements.

Guidelines for Partitioning Your Tree

When you partition eDirectory, you allow parts of the database to exist on several servers. With this capability, you can optimize network use by distributing the eDirectory data processing and storage load over multiple servers on the network. By default, a single partition is created. For more information on partitions, refer to [“Partitions” on page 46](#). For information on creating partitions, refer to [Chapter 5, “Managing Partitions and Replicas,” on page 113](#).

The following are guidelines for most networks. However, depending on the specific configuration, hardware, and traffic throughput of the network, you might need to adjust some guidelines to fit your needs.

Determining Partitions for the Upper Layers of the Tree

Just as you design your tree with a pyramid design, you will also partition with a pyramid design. Your partition structure will have few partitions at the top of the tree and more partitions as you move toward the bottom. Such a design creates fewer subordinate references than an eDirectory tree structure that has more partitions at the top than at the bottom.

This pyramid design can be achieved if you always create the partitions relatively close to the leaf objects, particularly the users. (An exception is the partition created at the root of the tree during installation.)

When designing the partitions for the upper layers, keep the following in mind:

- ◆ Partition the top of the tree based on the WAN infrastructure. Place fewer partitions at the top of the tree with more at the bottom.
You can create containers for each site separated by WAN links (placing each Server object in its local container), then create a partition for each site.
- ◆ In a network with WAN links, partitions should not span multiple locations.
This design ensures that replication traffic between different sites is not unnecessarily consuming WAN bandwidth.
- ◆ Partition locally around the servers. Keep physically distant servers in separate partitions.

For more information on managing your WAN traffic, see [Chapter 10, “WAN Traffic Manager,” on page 229](#).

Determining Partitions for the Lower Layers of the Tree

When designing the partitions for the lower layers of the eDirectory tree, keep the following in mind:

- ◆ Define lower-layer partitions by organizational divisions, departments, and workgroups, and their associated resources.
- ◆ Partition so that all objects in each partition are at a single location. This ensures that updates to eDirectory can occur on a local server.

Determining Partition Size

With eDirectory, we recommend the following design limits for partition sizes:

Element	Limit
Partition Size	Unlimited Objects Replica Directory Information Base (DIB) limited to 1 TB
Total number of partitions in tree	Unlimited
Number of child partitions per parent	150
Number of replicas per partition	50 Limited by replica DIB
Number of replicas per replica server	250

This change in design guidelines from NDS[®] 6 and 7 is due to architectural changes in NDS 8. These recommendations apply to distributed environments such as corporate enterprises. These recommendations might not subsequently apply to e-business or applications.

Although typical e-business users require that all the data be stored on a single server, eDirectory 8.7.3 provides filtered replicas that can contain a subset of objects and attributes from different areas of the tree. This allows for the same e-business needs without storing all the data on the server. For more information, see [“Filtered Replicas” on page 53](#).

Considering Network Variables

Consider the following network variables and their limitations when planning your partitions:

- ◆ The number and speed of servers
- ◆ The speed of network infrastructure (such as network adapters, hubs, and routers)
- ◆ The amount of network traffic

Guidelines for Replicating Your Tree

Creating multiple eDirectory partitions does not, by itself, increase fault tolerance or improve performance of the directory; however, strategically using multiple replicas does. The placement of replicas is extremely important for accessibility and fault tolerance. eDirectory data needs to be available as quickly as possible and needs to be copied in several places to ensure fault tolerance.

For information on creating replicas, refer to [Chapter 5, “Managing Partitions and Replicas,”](#) on [page 113](#).

The following guidelines will help determine your replica placement strategy.

- ◆ [“Workgroup Needs”](#) on [page 76](#)
- ◆ [“Fault Tolerance”](#) on [page 76](#)
- ◆ [“Determining the Number of Replicas”](#) on [page 77](#)
- ◆ [“Replicating the Tree Partition”](#) on [page 77](#)
- ◆ [“Replicating for Administration”](#) on [page 77](#)
- ◆ [“Meeting Bindery Services Needs for NetWare”](#) on [page 77](#)
- ◆ [“Managing WAN Traffic”](#) on [page 78](#)

Workgroup Needs

Place replicas of each partition on servers that are physically close to the workgroup that uses the information in that partition. If users on one side of a WAN link often access a replica stored on a server on the other side, place a replica on servers on both sides of the WAN link.

Place replicas in the location of highest access by users, groups, and services. If groups of users in two separate containers need access to the same object within another partition boundary, place the replica on a server that exists in the container one level above the two containers holding the group.

Fault Tolerance

If a disk crashes or a server goes down, replicas on servers in other locations can still authenticate users to the network and provide information on objects in partitions stored on the disabled server.

With the same information distributed on several servers, you are not dependent on any single server to authenticate you to the network or to provide services (such as login).

To create fault tolerance, plan for three replicas for each partition if the directory tree has enough servers to support that number. There should be at least two local replicas of the local partition. There is no need to have more than three replicas unless you need to provide for accessibility of the data at other locations, or you participate in e-business or other applications that need to have multiple instances of the data for load balancing and fault tolerance.

You can have only one master replica. Additional replicas must be read/write, read-only, or filtered. Most replicas should be read/write. They can handle object viewing, object management, and user login, just as the master replica can. They send out information for synchronization when a change is made.

Read-only replicas cannot be written to. They allow object searching and viewing, and they are updated when the replicas of the partition synchronize.

Do not depend on a subordinate reference or filtered replicas for fault tolerance. A subordinate reference is a pointer and does not contain objects other than the partition root object. Filtered replicas do not contain all objects within the partition.

eDirectory allows for an unlimited number of replicas per partition, but the amount of network traffic increases as the number of replicas increase. Balance fault tolerance needs with network performance needs.

You can store only one replica per partition on a server. A single server can store replicas of multiple partitions.

Depending on your organization's disaster recovery plan, the major work of rebuilding the network after a loss of a server or location can be done using partition replicas. If the location has only one server, back up eDirectory regularly. (Some backup software does not back up eDirectory.) Consider purchasing another server for fault tolerance replication.

Determining the Number of Replicas

The limiting factor in creating multiple replicas is the amount of processing time and traffic required to synchronize them. When a change is made to an object, that change is communicated to all replicas in the replica ring. The more replicas in a replica ring, the more communication is required to synchronize changes. If replicas must synchronize across a WAN link, the time cost of synchronization is greater.

If you plan partitions for many geographical sites, some servers will receive numerous subordinate reference replicas. eDirectory can distribute these subordinate references among more servers if you create regional partitions.

Replicating the Tree Partition

The Tree partition is the most important partition of the eDirectory tree. If the only replica of this partition becomes corrupted, users will experience impaired functionality on the network until the partition is repaired or the eDirectory tree is completely rebuilt. You will also not be able to make any design changes involving the Tree.

When creating replicas of the Tree partition, balance the cost of synchronizing subordinate references with the number of replicas of the Tree partition.

Replicating for Administration

Because partition changes originate only at the master replica, place master replicas on servers near the network administrator in a central location. It might seem logical to keep masters at remote sites; however, master replicas should be where the partition operations will occur.

We recommend that major eDirectory operations, such as partitioning, be handled by one person or group in a central location. This methodology limits errors that could have adverse effects to eDirectory operations and provides for a central backup of the master replicas.

The network administrator should perform high-cost activities, such as creating a replica, at times when network traffic is low.

Meeting Bindery Services Needs for NetWare

If you are using eDirectory on NetWare and your users require access to a server through bindery services, that server must contain a master or read/write replica that contains the bindery context. The bindery context is set by the SET BINDERY CONTEXT statement in autoexec.ncf.

Users can access objects providing bindery services only if real objects exist on that server. Adding a replica of a partition to the server adds real objects to the server and lets users with User objects in that partition log in to the server with a bindery connection.

For more information on bindery services, refer to [“NetWare Bindery Emulation” on page 54](#).

Managing WAN Traffic

If users currently use a WAN link to access particular directory information, you can decrease access time and WAN traffic by placing a replica containing the needed information on a server that users can access locally.

If you are replicating the master replicas to a remote site or are forced to place replicas over the WAN for accessibility or fault tolerance, keep in mind the bandwidth that will be used for replication.

Replicas should only be placed in nonlocal sites to ensure fault tolerance if you are not able to get the recommended three replicas, increase accessibility, and provide centralized management and storage of master replicas.

To control the replication of eDirectory traffic over WAN links, use WAN Manager. For more information, see [Chapter 10, “WAN Traffic Manager,” on page 229](#).

Planning the User Environment

After you have designed the basic structure of the eDirectory tree and have set up partitioning and replication, you should plan the user environment to simplify management and increase access to network resources. To create a user environment plan, review the users' needs and create accessibility guidelines for each area.

Reviewing Users' Needs

When you review users' needs, consider the following:

- ◆ Physical network needs, such as printers or file storage space

Evaluate if resources are shared by groups of users within a tree or shared by groups of users from multiple containers. Also consider the physical resource needs of remote users.

- ◆ Bindery services needs for NetWare users

Consider which applications are bindery-based and who uses them.

- ◆ Application needs

Consider which applications and data files are needed by users, what operating systems exist, and which groups or users need access to applications. Consider if the shared applications should be manually or automatically launched by applications such as ZENworks.

Creating Accessibility Guidelines

After you have gathered information about user needs, you should determine the eDirectory objects that you will use to create the users' environments. For example, if you create policy packages or Application objects, you should determine how many you will create and where you will allow them to be placed in the tree.

You should also determine how you will implement security to restrict user access. You should identify any security precautions related to specific security practices. For example, you could warn network administrators to avoid granting the eDirectory Supervisor right to Server objects because this right is inherited by the file system.

Designing eDirectory for e-Business

If you use eDirectory for e-Business, whether you are providing a portal for services or sharing data with another business, the recommendations already mentioned in this chapter might not apply to you.

You might want to follow these suggested eDirectory e-business design guidelines instead:

- ◆ Create a tree with a limited number of containers.

This guideline depends on the applications you use and your implementation of eDirectory. For example, a global deployment of a messaging server might require the more traditional eDirectory design guidelines discussed earlier in this chapter. Or, if you are going to distribute administration of users, you might create a separate Organizational Unit (OU) for each area of administrative responsibility.

- ◆ Maintain at least two partitions.

Maintain the default partition at the Tree level, and create a partition for the rest of the tree. If you have created separate OUs for administrative purposes, create partitions for each of the OUs.

If you are splitting the load over multiple servers, consider limiting the number of partitions, but still maintain at least two for backup or disaster recovery.

- ◆ Create at least three replicas of your tree for fault tolerance and load balancing.

Keep in mind that LDAP does not load balance itself. To balance the load on LDAP, consider using Layer 4 switches.

- ◆ Create a separate tree for e-Business. Limit the network resources, such as servers and printers, included in the tree. Consider creating a tree that contains only User objects.

You can use DirXML to link this user tree to your other trees that contain network information. For more information, see the *DirXML Administration Guide* (<http://www.novell.com/documentation/lg/dirxml11a/index.html>).

- ◆ Use auxiliary classes to customize your schema.

If a customer or application requires a User object that is different from the standard inetOrgPerson, use auxiliary classes to customize your schema. Using auxiliary classes allows application designers to change the attributes used in the class without needing to re-create the tree.

- ◆ Increase LDIF-import performance.

When the Novell Import Conversion Export utility is used, eDirectory indexes each object during the process. This can slow down the LDIF-import process. To increase the LDIF-import performance, suspend all indexes from the attributes of the objects you are creating, use the Novell Import Conversion Export utility, then resume indexing the attributes.

- ◆ Implement globally unique common names (CN).

eDirectory allows the same CN in different containers. However, if you use globally unique CNs, you can perform searches on CN without implementing logic for dealing with multiple replies.

Understanding the Novell Certificate Server

Novell Certificate Server™ allows you to mint, issue, and manage digital certificates by creating a Security container object and an Organizational Certificate Authority (CA) object. The Organizational CA object enables secure data transmissions and is required for Web-related products such as NetWare Web Manager and NetWare Enterprise Web Server. The first eDirectory server will automatically create and physically store the Security container object and Organizational CA object for the entire eDirectory tree. Both objects are created and must remain at the top of the eDirectory tree.

Only one Organizational CA object can exist in an eDirectory tree. After the Organizational CA object is created on a server, it cannot be moved to another server. Deleting and re-creating an Organizational CA object invalidates any certificates associated with the Organizational CA.

IMPORTANT: Make sure that the first eDirectory server is the server that you intend to permanently host the Organizational CA object and that the server will be a reliable, accessible, and continuing part of your network.

If this is not the first eDirectory server on the network, the installation program finds and references the eDirectory server that holds the Organizational CA object. The installation program accesses the Security container and creates a Server Certificate object.

If an Organizational CA object is not available on the network, Web-related products will not function.

Rights Required to Perform Tasks on Novell Certificate Server

To complete the tasks associated with setting up Novell Certificate Server, the administrator needs to have rights as described in the following table.

Novell Certificate Server Task	Rights Required
Base security setup for installing the first server into a new tree or upgrading the first server in a tree where there is no base security previously installed	Supervisor right at the root of the tree Supervisor right on the Security container
Base security setup for installing subsequent servers	Supervisor right on the server's container Supervisor right on the W0 object (located inside the Security container)
Creating the Organizational CA	Supervisor right on the Security container
Creating Server Certificate objects	Supervisor right on the server's container Read right to the NDSPKI:Private Key attribute on the Organizational CA's object

The root administrator can also delegate the authority to use the Organizational CA by assigning the following rights to subcontainer administrators. Subcontainer administrators require the following rights to install Novell eDirectory with SSL security:

- ◆ Read right to the NDSPKI:Private Key attribute on the Organizational CA's object, located in the Security container.
- ◆ Supervisor right to the W0 object located in the Security container, inside the KAP object.

These rights are assigned to a group or a role, where all the administrative users are defined. For a complete list of required rights to perform specific tasks associated with Novell Certificate Server, refer to the [Novell Certificate Server \(http://www.novell.com/documentation/lg/crt27/index.html\)](http://www.novell.com/documentation/lg/crt27/index.html) online documentation.

Ensuring Secure eDirectory Operations on Linux, Solaris, AIX, and HP-UX Systems

eDirectory includes Public Key Cryptography Services (PKCS), which contains the Novell Certificate Server that provides Public Key Infrastructure (PKI) services, Novell International Cryptographic Infrastructure (NICI), and SAS*-SSL server.

The following sections provide information about performing secure eDirectory operations:

- ♦ “Verifying Whether NICI Is Installed and Initialized on the Server” on page 81
- ♦ “Initializing the NICI Module on the Server” on page 81
- ♦ “Starting the Certificate Server (PKI Services)” on page 82
- ♦ “Stopping the Certificate Server (PKI Services)” on page 82
- ♦ “Creating an Organizational Certificate Authority Object” on page 82
- ♦ “Creating a Server Certificate Object” on page 83
- ♦ “Exporting an Organizational CA’s Self-Signed Certificate” on page 83

For information about using external certificate authority, refer to the *Novell Certificate Server Administration Guide* (<http://www.novell.com/documentation/lg/crt27/index.html>).

Verifying Whether NICI Is Installed and Initialized on the Server

Verify the following conditions, which indicate that the NICI module has been properly installed and initialized:

- ♦ The file `/etc/nici.cfg` exists
- ♦ The directory `/var/novell/nici` exists
- ♦ The file `/var/novell/nici/primenici` exists

If these conditions are not met, follow the procedure in the next section, “[Initializing the NICI Module on the Server](#).”

Initializing the NICI Module on the Server

- 1** Stop the eDirectory server.
 - ♦ On Linux systems, enter


```
/etc/rc.d/init.d/ndsd stop
```
 - ♦ On Solaris systems, enter


```
/etc/init.d/ndsd stop
```
 - ♦ On AIX systems, enter


```
/etc/rc.d/init.d/ndsd stop
```
 - ♦ On HP-UX systems, enter


```
/sbin/init.d/ndsd stop
```
- 2** Verify whether the NICI package is installed.
 - ♦ On Linux systems, enter


```
rpm -qa | grep nici
```
 - ♦ On Solaris systems, enter


```
pkginfo | grep NOVLniu0
```

- ♦ On AIX systems, enter
`rpm -qa | grep nici`
 - ♦ On HP-UX systems, enter
`swlist | grep NOVLniu0`
- 3** (Conditional) If the NICI package is not installed, install it now.
You will not be able to proceed if the NICI package is not installed.
- 4** Copy the .nfk file provided with the package to the /var/novell/nici directory.
Execute the /var/novell/nici/primenici program.
- 5** Start the eDirectory server.
- ♦ On Linux systems, enter
`/etc/rc.d/init.d/ndsd start`
 - ♦ On Solaris systems, enter
`/etc/init.d/ndsd start`
 - ♦ On AIX systems, enter
`/etc/rc.d/init.d/ndsd start`
 - ♦ On HP-UX systems, enter
`/sbin/init.d/ndsd start`

Starting the Certificate Server (PKI Services)

To start PKI services, enter


`npki -l.`

Stopping the Certificate Server (PKI Services)

To stop PKI services, enter

`npki -u.`

Creating an Organizational Certificate Authority Object

- 1** Launch Novell iManager.
- 2** Log in to the eDirectory tree as an administrator with the appropriate rights.
To view the appropriate rights for this task, see [Creating an Organizational CA \(http://www.novell.com/documentation/lg/crt27/crtadmin/data/a2ziby.html#a2zisy5\)](http://www.novell.com/documentation/lg/crt27/crtadmin/data/a2ziby.html#a2zisy5) in the *Novell Certificate Server Administration Guide*.
- 3** Click the Roles and Tasks button , click PKI Certificate Management, then click Create Certificate Authority.


This opens the Create Organizational Certificate Authority Object Wizard. Follow the prompts to create the object. For specific information on any of the wizard pages, click Help.

NOTE: You can have only one Organizational CA for your eDirectory tree.

Creating a Server Certificate Object

Server Certificate objects are created in the container that holds the eDirectory Server object. Depending on your needs, you might create a separate Server Certificate object for each cryptography-enabled application on the server. Or you might create one Server Certificate object for all applications used on that server.

NOTE: The terms Server Certificate Object and Key Material Object (KMO) are synonymous. The schema name of the eDirectory object is NDSPKI:Key Material.

- 1 Launch Novell iManager.
- 2 Log in to the eDirectory tree as an administrator with the appropriate rights.
To view the appropriate rights for this task, see [Creating Server Certificate Objects \(http://www.novell.com/documentation/lg/crt27/crtadmin/data/a2ziby.html#a2zisy9\)](http://www.novell.com/documentation/lg/crt27/crtadmin/data/a2ziby.html#a2zisy9) in the *Novell Certificate Server Administration Guide*.
- 3 Click the Roles and Tasks button , click PKI Certificate Management, then click Create Server Certificate.


This opens the Create Server Certificate Wizard. Follow the prompts to create the object. For specific information on any of the wizard pages, click Help.

Exporting an Organizational CA's Self-Signed Certificate

A self-signed certificate can be used for verifying the identity of the Organizational CA and the validity of a certificate signed by the Organizational CA.

From the Organizational CA's property page, you can view the certificates and properties associated with this object. From the Self-Signed Certificate property page, you can export the self-signed certificate to a file for use in cryptography-enabled applications.

The self-signed certificate that resides in the Organizational CA is the same as the Trusted Root certificate in a Server Certificate object that has a certificate signed by the Organizational CA. Any service that recognizes the Organizational CA's self-signed certificate as a trusted root will accept a valid user or server certificate signed by the Organizational CA.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Administration > Modify Object.
- 3 Specify the name and context of an Organizational Certificate Authority object, then click OK.

Organizational Certificate Authority objects are located in Security container.

- 4 Click the Certificates tab, then click Self-Signed Certificate.
- 5 Click Export.

This opens the Export Certificate Wizard. Follow the prompts to export the certificate. For specific information on any of the wizard pages, click Help.

- 6 On the Export Certificate Summary page, click Save the Exported Certificate to a File.

The certificate is saved to a file and is available to be imported into a cryptography-enabled application as the trusted root.

- 7 Click Close.

Include this file in all command line operations that establish secure connections to eDirectory

Synchronizing Network Time

Time synchronization is a service that maintains consistent server time across the network. Time synchronization is provided by the server operating system, not by eDirectory. eDirectory maintains its own internal time to ensure the proper order of eDirectory packets, but it gets its time from the server operating system.

This section focuses on integrating NetWare time synchronization with that of Windows, Linux, Solaris, AIX, and HP-UX.

Synchronizing Time on NetWare Servers

In IP networks and mixed protocol networks, NetWare 5 servers communicate time with other servers using IP. NetWare 5 servers use timesync.nlm and Network Time Protocol (NTP) to accomplish this.

Time synchronization in NetWare 5 and 6 always uses timesync.nlm, whether servers are using IP only, IPX™ only, or both protocols. Timesync.nlm loads when a server is installed. NTP can be configured through timesync.nlm.

If your network also uses Windows, Linux, Solaris, AIX, or HP-UX, you should use NTP to synchronize the servers because it is a standard to provide time synchronization.

For NetWare 3 and NetWare 4, third-party NTP time services are available.

For more information on time synchronization software, see [The Network Time Protocol \(http://www.ntp.org\)](http://www.ntp.org) Web site.

NTP

NTP functions as part of the UDP protocol suite, which is part of the TCP/IP protocol suite. Therefore, a computer using NTP must have the TCP/IP protocol suite loaded. Any computers on your network with Internet access can get time from NTP servers on the Internet.

NTP synchronizes clocks to the Universal Time Coordinated (UTC) standard, which is the international time standard.

NTP introduces the concept of a stratum. A stratum-1 server has an attached accurate time piece such as a radio clock or an atomic clock. A stratum-2 server gets time from a stratum-1 server, and so on.

For NetWare 5 and 6 servers, you can load ntp.nlm to implement NTP time synchronization through timesync.nlm. When NTP is configured with the timesync.nlm on an IP server, NTP becomes the time source for both IP and IPX servers. In this case, IPX servers must be set to secondary servers.

For more information on time synchronization, see the *Network Time Management Administration Guide* (http://www.novell.com/documentation/lg/nw65/time_enu/data/h15k6r0y.html) and the *Network Time Protocol Administration Guide* (<http://www.novell.com/documentation/lg/nw65/ntp/data/aizwub2.html>).

TIMESYNC.NLM

Timesync.nlm synchronizes time among NetWare servers. You can use timesync.nlm with an external time source like an Internet NTP server. You can also configure Novell Client™ workstations to update their clocks to servers running the timesync.nlm.

For more information on time synchronization, refer to the *Network Time Management Administration Guide* (http://www.novell.com/documentation/lg/nw65/time_enu/data/hl5k6r0y.html).

Synchronizing Time on Windows Servers

For information on time synchronization for Windows NT and Windows 2000 servers, refer to the operating system documentation.

Synchronizing Time on Linux, Solaris, AIX, or HP-UX Systems

You can use the xntpd Network Time Protocol (NTP) daemon to synchronize time on Linux, Solaris, AIX, and HP-UX servers. xntpd is an operating system daemon that sets and maintains the system time-of-day in synchronism with Internet standard time servers.

For more information on running xntpd on AIX systems, see *xntpd Daemon* (http://publibn.boulder.ibm.com/doc_link/en_US/a_doc_lib/cmds/aixcmds6/xntpd.htm) in the *AIX Commands Reference, Volume 6*.

For more information on running xntpd on Solaris system, see <http://docs.sun.com/?p=/doc/806-0625/6j9vfim2v&a=view#xntpd-1m-indx-2> (<http://docs.sun.com/?p=/doc/806-0625/6j9vfim2v&a=view#xntpd-1m-indx-2>).

For more information on running xntpd on HP-UX system, see *Configuring NTP* (http://docs.hp.com/cgi-bin/fsearch/framedisplay?top=/hpux/onlinedocs/B2355-90147/B2355-90147_top.html&con=/hpux/onlinedocs/B2355-90147/00/00/58-con.html&toc=/hpux/onlinedocs/B2355-90147/00/00/58-toc.html&searchterms=ntp%7cconfiguring&queryid=20030922-153023).

For information on running ntpd on Linux systems, see *ntpd - Network Time Protocol (NTP) Daemon* (<http://www.eecis.udel.edu/~mills/ntp/html/ntpd.html>).

Verifying Time Synchronization

To verify that time is synchronized in the tree, run DSRepair from a server in the Tree that has at least Read/Write rights to the Tree object.

NetWare

- 1 At the server console, load dsrepair.nlm.
- 2 Select Time Synchronization.

For help interpreting the log, click F1.

NOTE: The following command will help troubleshoot time synchronization issues:

```
set timesync debug=7
```

Windows

- 1 Click Start > Settings > Control Panel > Novell eDirectory Services.
- 2 Click dsrepair.dlm > Start.
- 3 Click Repair > Time Synchronization.

Linux, Solaris, AIX, and HP-UX

- 1 Run the following command:

```
ndsrepair -T
```

3

Managing Objects

Novell® eDirectory™ includes Novell iManager 2.0.2, a Web-based network management application that lets you manage the objects in your eDirectory tree. To understand the features and benefits of Novell iManager, see the *Novell iManager 2.0.x Administration Guide* (<http://www.novell.com/documentation/lg/imanager20/index.html>).

Managing eDirectory objects involves creating, modifying, and manipulating objects. For example, you might need to create user accounts and administer user rights. Use Novell iManager to:

- ◆ Perform administration basics, such as browsing, creating, editing, and organizing objects.
- ◆ Create user accounts, including specifying a user's login name and supplying other information used by eDirectory
- ◆ Administer rights (assign rights, grant equivalence, block inheritance, and view effective rights). See **“Administering Rights” on page 61** for more information.
- ◆ Configure role-based administration (define administrator roles for specific administrative applications through the role-based services object).
- ◆ Manage NetWare® server resources (viewing and modifying server and file system information, managing files and folders on NetWare volumes, salvaging and purging deleted files, controlling allocation of volume space, and creating objects to facilitate file management).

This chapter contains information on the following topics:


- ◆ **“General Object Tasks” on page 87**
- ◆ **“Managing User Accounts” on page 92**
- ◆ **“Configuring Role-Based Services” on page 97**


General Object Tasks

This section contains steps for basic tasks you will use when managing your eDirectory tree:

- ◆ **“Browsing the eDirectory Tree” on page 88**
- ◆ **“Creating an Object” on page 90**
- ◆ **“Modifying an Object's Properties” on page 90**
- ◆ **“Copying Objects” on page 90**
- ◆ **“Moving Objects” on page 91**
- ◆ **“Deleting Objects” on page 91**
- ◆ **“Renaming Objects” on page 91**

Browsing the eDirectory Tree

The View Objects button () in Novell iManager lets you search or browse for objects in your eDirectory tree. You can view the structure of your tree and right-click objects to perform tasks. The tasks available depend on the type of object you select.

The eDirectory Object Selector page in Novell iManager also lets you search or browse for objects. In most entry fields in Novell iManager, you can specify an object name and context, or you can click the Object Selector button () to search or browse for the object you want. Selecting an object in the eDirectory Object Selector page inserts the object and the object's context into the entry field.

This section contains the following information:


- ♦ [“Using the View Object Button” on page 88](#)
- ♦ [“Using the Object Selector Button” on page 89](#)



Using the View Object Button

Use the techniques described below to locate the specific objects you want to manage.

- ♦ [“Using Browse” on page 88](#)
- ♦ [“Using Search” on page 89](#)


Using Browse

- 1** In Novell iManager, click the View Objects button (.
- 2** Click Browse.
- 3** Use the following options to browse for an object:

Option	Description
	Lets you move down one level in the tree.
	Lets you move up one level in the tree.
Context	<p>Lets you specify the name of the container whose contents you want to view.</p> <p>To use this option, specify the name of the container you want, then click Apply.</p>
Name	<p>Lets you specify the name of an object.</p> <p>You can use an asterisk (*) as a wildcard character in this field. For example, g* finds all objects starting with g, such as Germany or Greg, and *te finds all entries ending in te, such as Kate or Corporate.</p> <p>To use this option, type the name you want, then click Apply.</p>
Type	<p>Lets you specify the type of object you want to search for. The default is All Available Types.</p> <p>To use this option, select an object type from the drop-down list, then click Apply.</p>

- 4 When you find the object you are looking for, right-click the object, then choose from the list of available tasks to perform.

Using Search


- 1 In Novell iManager, click the View Objects button .
- 2 Click Search.
- 3 In the Context field, specify the name of the container you want to search in.
Click Search Sub-containers to include all subcontainers located within the current container in the search.
- 4 In the Name field, specify the name of the object you want to search for.
You can use an asterisk (*) as a wildcard character in this field. For example, g* finds all objects starting with g, such as Germany or Greg, and *te finds all entries ending in te, such as Kate or Corporate.
- 5 Select the type of object you want to search for from the Type drop-down list.
- 6 Click Search.
- 7 When you find the object you are looking for, right-click the object, then choose from the list of available tasks to perform.



Using the Object Selector Button

Use the techniques described below to locate the specific objects you want to manage.


- ♦ [“Using Browse” on page 89](#)
- ♦ [“Using Search” on page 90](#)

Using Browse



- 1 Click the Object Selector button  on an iManager property page.
- 2 Click Browse.
- 3 Use the following options to browse for an object:

Option	Description
	Lets you move down one level in the tree.
	Lets you move up one level in the tree.
Look In	Specify the name of the container whose contents you want to view, then click Apply.
Look for Objects Named	Lets you specify the name of an object. You can use an asterisk (*) as a wildcard character in this field. For example, g* finds all objects starting with g, such as Germany or Greg, and *te finds all entries ending in te, such as Kate or Corporate. To use this option, type the name you want, then click Apply.



Using Search

- 1 Click the Object Selector button  on an iManager property page.
- 2 Click Search.
- 3 In the Start Search In field, specify the name of the container you want to search in.
Click Search Sub-containers to include all subcontainers located within the current container in the search.
- 4 In the Search For Objects Named field, specify the name of the object you want to search for.
You can use an asterisk (*) as a wildcard character in this field. For example, g* finds all objects starting with g, such as Germany or Greg, and *te finds all entries ending in te, such as Kate or Corporate.
- 5 Click Search.

Creating an Object


- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Administration > Create Object.
- 3 Select an object from the list of available object classes, then click OK.
- 4 Specify the information requested, then click OK.
The information requested depends on the type of object you are creating. Click  for more information.
- 5 Click OK.

Modifying an Object's Properties

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Administration > Modify Object.
- 3 Specify the name and context of the object or objects you want to modify, then click OK.
- 4 Edit the property pages you want.
Click  for more information on specific property pages.
- 5 Click OK.

Copying Objects

This option lets you create a new object with the same attribute values as an existing object, or copy attribute values from one object to another.


- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Administration > Copy Object.
- 3 In the Object to Copy From field, specify the name and context of the object you want to copy.
- 4 Select one of the following options:
 - ♦ Create New Object and Copy Attribute Values
 - ♦ Copy Attribute Values to an Existing Object

- 5 If you want to copy access control list (ACL) rights to the object you are creating/modifying, select Copy ACL Rights.

Copying ACL rights can take additional processing time depending upon your system and networking environment.

- 6 Click OK.


Moving Objects

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Administration > Move Object.
- 3 In the Object Name field, specify the name and context of the object or objects you want to move.
- 4 In the Move To field, specify the container you want to move the object or objects to.
- 5 If you want to create an Alias in the old location for each object being moved, select Create an Alias in Place of Moved Object.


This allows any operations that are dependent on the old location to continue uninterrupted until you can update those operations to reflect the new location.

- 6 Click OK.

Deleting Objects

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Administration > Delete Object.
- 3 Specify the name and context of the object or objects you want to delete.
- 4 Click OK.

Renaming Objects

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Administration > Rename Object.
- 3 In the Object Name field, specify the name and context of the object you want to rename.
- 4 In the New Object Name field, specify the new name for the object.
Do not include the object's context in the New Object Name field.
- 5 If you want to create an Alias for the object being renamed, select Create an Alias in Place of Renamed Object.

This allows any operations that are dependent on the old object name to continue uninterrupted until you can update those operations to reflect the new name.

- 6 If you want to save the old object name, select Save Old Name.

This saves the old name as an additional (unofficial) value of the Name property. Saving the old name lets users search for the object based on that name. After renaming the object, you can view the old name in the Other Name field on the General Identification tab for that object.

- 7 Click OK.

Managing User Accounts

Setting up an eDirectory user account involves creating a User object and setting properties to control login and the user's network computing environment. You can use a template object to facilitate these tasks.

You can create login scripts to cause users to be connected automatically to the files, printers, and other network resources they need when they log in. If several users use the same resources, you can put the login script commands in container and profile login scripts.

This section contains the following information:

- ◆ “Creating and Modifying User Accounts” on page 92
- ◆ “Setting Up Optional Account Features” on page 93
- ◆ “Setting Up Login Scripts” on page 95
- ◆ “Login Time Restrictions for Remote Users” on page 96
- ◆ “Deleting User Accounts” on page 97



Creating and Modifying User Accounts

A user account is a User object in the eDirectory tree. A User object specifies a user's login name and supplies other information used by eDirectory to control the user's access to network resources.



This section contains the following information:

- ◆ “Creating a User Object” on page 92
- ◆ “Modifying a User Account” on page 92
- ◆ “Enabling a User Account” on page 93
- ◆ “Disabling a User Account” on page 93


Creating a User Object

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Users > Create User.
- 3** Specify a user name and a last name for the user.
- 4** Specify a container to create the user in.
- 5** Specify any additional (optional) information you want, then click OK.
Click  for more information on the available options.
- 6** Click OK.


Modifying a User Account

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Users > Modify User.
- 3** Specify the name and context of the User or Users you want to modify, then click OK.
- 4** Edit the property pages you want.
Click  for more information on specific properties.
- 5** Click OK.

Enabling a User Account

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Users > Enable Account.
- 3** Specify the name and context of the User, then click OK.



Disabling a User Account

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Users > Disable Account.
- 3** Specify the name and context of the User, then click OK.



Setting Up Optional Account Features

After creating a User object, you can set up the user's network computing environment and implement extra login security features.

Setting Up a User's Network Computing Environment

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Users > Modify User.
- 3** Specify the name and context of the User or Users you want to modify, then click OK.
- 4** On the General tab, select the Environment page.
- 5** Fill in the property page.
Click  for more information on specific properties.
- 6** Click OK.

Setting Up Extra Login Security for a User


- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Users > Modify User.
- 3** Specify the name and context of the User or Users you want to modify, then click OK.
- 4** On the Restrictions tab, fill in the property pages you want.
Click  for details on any page.

Page	Description
Password Restrictions	Sets up a login password.
Login Restrictions	<ul style="list-style-type: none"> ◆ Enable or disable the account. ◆ Limit the number of concurrent login sessions. ◆ Set a login expiration and lockout date.

Page	Description
Time Restrictions	<p>Restricts the times when the user can be logged in. If you set a restriction and the object is logged in when the restricted time arrives, the system issues a five-minute warning and then (after five minutes) logs the object out if it isn't logged out already.</p> <p>If the user will log in remotely, see “Login Time Restrictions for Remote Users” on page 96.</p>
Address Restrictions	Restricts the network locations (workstations) that this user can log in from. If you don't set restrictions on this page, the user can log in from any network location.
Account Balance	Sets up an accounting of this user's server usage.
Intruder Lockout	Lets you work with this account if it has been locked because of intruder detection. To manage the intruder detection setup, use the Intruder Detection property page of the parent container.

- 5 Click OK.

Setting Up Intruder Detection for All Users in a Container

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Administration > Modify Object.
- 3 Specify the name and context of a container object, then click OK.
- 4 On the General tab, select the Intruder Detection page.
- 5 Select from the following options:

Option	Description
Detect Intruders	Enables the intruder detection system for the user accounts in the container.
Incorrect Login Attempts	Specifies the number of consecutive failed login attempts that are allowed before intruder detection is activated. If a person uses any of the user accounts in this container to log in and fails consecutively more than this number of times, intruder detection is activated. The number is stored in the Login Intruder Limit property of the container.
Intruder Attempt Reset Interval	Specifies the time span in which consecutive failed logins must occur for intruder detection to be activated. Enter the number of days, hours, and minutes.
Lock Account After Detection	Specifies whether to disable login if intruder detection is activated on a user account in this container. If you don't check this check box, no action is taken when intruder detection is activated. If you check this check box and the system locks a user account due to intruder detection, you can unlock the account by unchecking the Account Locked check box on the Intruder Lockout property page of the User object.

Option	Description
Days, Hours, Minutes	These three fields specify the length of time that login is disabled when intruder detection is activated on a user account in this container. Enter the number of days, hours, and minutes you want, or accept the default of 15 minutes. After the specified time elapses, the system re-enables login for the user account. The contents of these fields are stored in the Intruder Lockout Reset Interval property of the container.

- 6 Click OK.


Setting Up Login Scripts

A login script is a list of commands that executes when a user logs in. It is typically used to connect the user to network resources like files and printers. Login scripts execute on the user's workstation in the following order:

1. Container login script
2. Profile login script
3. User login script

During login, if the system doesn't find one of these login scripts, it skips to the next one in the list. If none are found, the system executes a default script that maps a search drive to a folder on the user's default server. The default server is set on the Environment property page of the user object.

Creating a Login Script

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Administration > Modify Object.
- 3 Specify the name and context of the object that you want to create the login script on.

To Have the Login Script Apply To	Create It On
One user only	The User object
One or more users that haven't been created yet	A Template object
All the users in a container	The container object
A set of users in one or more containers	A Profile object

- 4 Click OK.
- 5 On the General tab, select the Login Script page.
- 6 Enter the login script commands you want.


See the *Login Script Commands Guide* (<http://www.novell.com/documentation/lg/noclienu/index.html>) for more information.

- 7 Click OK.

Assigning a Profile to a User

Associating a profile with a User object causes the profile's login script to execute during the user's login. Make sure that the user has Browse rights to the Profile object and Read rights to the Login Script property of the profile object.


See [“Viewing Effective Rights to an eDirectory Object or Property” on page 65](#) for more information.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click User > Modify User.
- 3 Specify the name and context of the User object that you want to create the login script on.
- 4 Click OK.
- 5 On the General tab, select the Login Script page.
- 6 To associate a profile object with this object, enter the name and context of the profile object in the Profile field.
- 7 Click OK.

Login Time Restrictions for Remote Users

On the Time Restrictions property page of a User object, you can restrict the times when the user can be logged in to eDirectory. (By default, there are no login time restrictions.) If you set a login time restriction and the user is logged in when the restricted time arrives, the system issues a warning to log out within five minutes. If the user is still logged in after five minutes, he or she is logged out automatically and loses any unsaved work.

If a user logs in remotely from a different time zone than the server processing the login request, any login time restrictions that have been set for the user are adjusted for the time difference. For example, if you restrict a user from logging in Mondays from 1:00 a.m. to 6:00 a.m. and the user logs in remotely from a time zone that is one hour later than the server, the restriction effectively becomes 2:00 a.m. to 7:00 a.m. for that user.


- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click Users > Modify User.
- 3 Specify the name and context of the User or Users you want to modify, then click OK.
- 4 On the Restrictions tab, click Time Restrictions.
- 5 Select from the following options:

Option	Description
Time Grid	Each cell in the time grid represents a half hour on a particular day of the week. Red cells represent restricted times (when this object cannot be logged in). Gray cells represent unrestricted times (when the object can be logged in). To create a time restriction, click the desired times to make them dark gray. You can also select multiple times by holding down the Shift key, clicking a cell, then dragging across the corresponding cells. The login time restrictions you set are stored in the Login Allowed Time Map property of this object.

Option	Description
Add Time Restrictions	To add a time restriction, select a gray cell, then select this option.
Remove Time Restrictions	To remove a time restriction, select a red cell, then select this option.
Update	Click this button to enable the selection.
Reset	Click this button to reset the time grid to the way it was before you opened this property page.

- 6 Click OK.

Deleting User Accounts

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click Users > Delete User.
- 3 Specify the name and context of the User or Users you want to delete.
- 4 Click OK.







Configuring Role-Based Services

Novell iManager gives administrators the ability to assign specific responsibilities to users and to present the user with only the tools (and their accompanying rights) necessary to perform those sets of responsibilities. This functionality is called *Role-Based Services (RBS)*.

Role-Based Services allows administrators to focus the user on a specified set of functions, called *tasks*, and objects as determined by the grouping of tasks called *roles*. What users see when they access iManager is based on their role assignments in eDirectory. Only the tasks assigned to that user are displayed. The user does not need to browse the tree to find an object to administer; the iManager plug-in for that task presents the necessary tools and interface to perform the task.

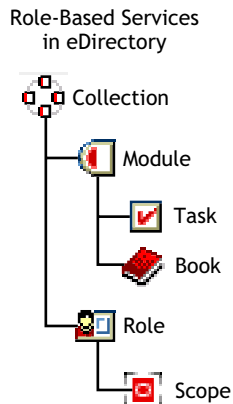
You can assign multiple roles to a single user. You can also assign the same role to multiple users.

Role-Based Services is represented by objects defined in eDirectory. The base eDirectory schema gets extended during the iManager installation. The RBS object types are listed in the following table.

Object	Description
 rbsCollection	<p>A container object that holds all RBS Role and Module objects.</p> <p>rbsCollection objects are the topmost containers for all RBS objects. A tree can have any number of rbsCollection objects. These objects have “owners,” which are users who have management rights over the collection.</p> <p>rbsCollection objects can be created in any of the following containers:</p> <ul style="list-style-type: none"> ◆ Country ◆ Domain ◆ Locality ◆ Organization ◆ Organizational Unit
 rbsRole	<p>A container object that specifies the tasks that users (members) are authorized to perform. Defining a role includes creating an rbsRole object and specifying the tasks that the role can perform.</p> <p>Role members can be Users, Groups, Organizations, or Organizational Units, and they are associated to a role in a specific scope of the tree. The rbsTask and rbsBook objects are assigned to rbsRole objects.</p> <p>rbsRole objects can be created only in rbsCollection containers.</p>
 rbsModule	<p>A container object that holds rbsTask and rbsBook objects. rbsModule objects have a module name attribute that represents the name of the product that defines the tasks or books (for example, eDirectory Maintenance Utilities, NMAS Management, or Novell Certificate Server Access).</p> <p>rbsModule objects can be created only in rbsCollection containers.</p>
 rbsTask	<p>A leaf object that represents a specific function, such as resetting login passwords.</p> <p>rbsTask objects are located only in rbsModule containers.</p>
 rbsBook	<p>A leaf object that containing a list of pages assigned to the book. An rbsBook can be assigned to one or more Roles and to one or more Object class types.</p> <p>rbsBook objects are located only in rbsModule containers.</p>
 rbsScope	<p>A leaf object used for ACL assignments (instead of making assignments for each User object). rbsScope objects represent the context in the tree where a role will be performed and are associated with rbsRole objects. They inherit from the Group class. User objects are assigned to an rbsScope object. These objects have a reference to the scope of the tree that they are associated with.</p> <p>This object is dynamically created when needed, then automatically deleted when no longer needed. They are located only in rbsRole containers.</p> <p>WARNING: Never change the configuration of a Scope object. Doing so will have serious consequences and could possibly break the system.</p>

The RBS objects reside in the eDirectory tree as depicted in the following figure.

Figure 22 RBS Objects in the eDirectory Tree



Defining RBS Roles

RBS roles specify the tasks that users are authorized to perform. Defining an RBS role includes creating an `rbRole` object and specifying the tasks that the role can perform and the User, Group, or container objects that can perform those tasks. In some cases, Novell iManager plug-ins (product packages) provide predefined RBS roles that you can modify.


The tasks that RBS roles can perform are exposed as `rbTask` objects in your eDirectory tree. These objects are added automatically during the installation of product packages. They are organized into one or more `rbModules`, which are containers that correspond to the different functional modules of the product.

For information on assigning members to a role, see [“Assigning RBS Role Membership and Scope” on page 100](#).

- ◆ [“Creating a Role Object” on page 99](#)
- ◆ [“Modifying the Tasks Associated with a Role” on page 100](#)
- ◆ [“Assigning RBS Role Membership and Scope” on page 100](#)
- ◆ [“Deleting a Role-Based Services Object” on page 100](#)

Creating a Role Object



Use the Create iManager Role Wizard to create a new `rbRole` object. We recommend creating the new `rbRole` object in the same `rbCollection` container where the other `rbRole` objects reside (for example, the Role-Based Services Collection container).

- 1** In Novell iManager, click the Configure button .
- 2** Click Role Configuration > Create iManager Role.
- 3** Follow the instructions in the Create iManager Role Wizard.

See [“Defining Custom RBS Tasks” on page 101](#) for information on adding members to roles.

Modifying the Tasks Associated with a Role

Each RBS role has a set of available tasks associated with it. You can choose which tasks are assigned to a particular role, adding or removing tasks as necessary.

- 1 In Novell iManager, click the Configure button .
- 2 Click Role Configuration > Modify iManager Roles.
- 3 To add or remove tasks from a role, click the Modify Tasks button  to the left of the role you want to modify.
- 4 Add or remove tasks from the Assigned Task list.
- 5 Click OK.

Assigning RBS Role Membership and Scope



After you have defined the RBS roles needed in your organization, you can assign members to each role. In doing so, you specify the scope in which each member can exercise the functions of the role. The scope is the location or context in the eDirectory tree where this role can be performed.

A user can be assigned to a role in the following ways:


- ♦ Directly
- ♦ Through group and dynamic group assignments. If a user is a member of a group or a dynamic group that is assigned to a role, then the user has access to the role.
- ♦ Through organizational role assignments. If a user is an occupant of a organizational role that is assigned a role, then the user has access to the role.
- ♦ Through container assignment. A user object has access to all of the roles that its parent container is assigned. This could also include other containers up to the root of the tree.

A user can be associated with a role multiple times, each with a different scope. You can also assign the same task to multiple members.

To assign role membership and scope:

- 1 In Novell iManager, click the Configure button .
- 2 Click Role Configuration > Modify iManager Roles.
- 3 To add or remove members from a role, click the Modify Members button  to the left of the role you want to modify.
- 4 In the Name field, specify an object name (a User, Group, or Container object) and context.
- 5 In the Scope field, specify an Organization or Organizational Unit object name and context.
- 6 Click Add, then click OK.


Deleting a Role-Based Services Object

- 1 In Novell iManager, click the Configure button .
- 2 Click Role Configuration > Delete Role.
- 3 Specify the name and context of the RBS role you want to delete.
- 4 Click OK.

Defining Custom RBS Tasks


- ◆ “Creating an iManager Task” on page 101
- ◆ “Creating a Server Administration Task” on page 101
- ◆ “Modify Role Assignment” on page 101
- ◆ “Deleting a Task” on page 101

Creating an iManager Task


- 1** In Novell iManager, click the Configure button .
- 2** Click Task Configuration > Create iManager Task.
- 3** Follow the instructions in the Task Builder to create a custom task.

Creating a Server Administration Task


Use the Create Server Administration Task Wizard to build custom tasks to access a server's services. The system administrator should verify that the service is available on the server.

- 1** In Novell iManager, click the Configure button .
- 2** Click Task Configuration > Create Server Administration Task.
- 3** Follow the instructions in the Create Server Administration Task Wizard.

Modify Role Assignment

- 1** In Novell iManager, click the Configure button .
- 2** Click Task Configuration > Modify Role Assignment.
- 3** Specify the name and context of the task you want to modify, then click Next.
- 4** Move the roles you want from the Available Roles column to the Assigned Roles column.
- 5** Click OK.

Deleting a Task

- 1** In Novell iManager, click the Configure button .
- 2** Click Task Configuration > Delete Task.
- 3** Specify the name and context of the task you want to delete, then click OK.

4

Managing the Schema

The schema of your Novell® eDirectory™ tree defines the classes of objects that the tree can contain, such as Users, Groups, and Printers. It specifies the attributes (properties) that comprise each object type, including those that are required when creating the object and those that are optional. The maximum number of characters in a class name or an attribute name is limited to 32 characters of any language.

Each eDirectory object belongs to an object class that specifies which attributes can be associated with the object. All attributes are based on a set of attribute types that are, in turn, based on a standard set of attribute syntaxes.

The eDirectory schema not only controls the structure of individual objects, but it also controls the relationship among objects in the eDirectory tree. The schema rules allow some objects to contain other subordinate objects. Thus the schema gives structure to the eDirectory tree.

You might need to make changes to your schema as your organization's informational needs change. For example, if you never required a fax number on your User object before but you need one now, you can create a new User class that has Fax Number as a mandatory attribute, then begin using the new User class to create User objects.

The Schema Management role in Novell iManager lets those with the Supervisor right to a tree customize the schema of that tree and perform the following tasks:

- ◆ View a list of all classes and attributes in the schema.
- ◆ Extend the schema by adding a class or an attribute to the existing schema.
- ◆ Create a class by naming it and specifying applicable attributes, flags, and containers to which it can be added, and parent classes from which it can inherit attributes.
- ◆ Create an attribute by naming it and specifying its syntax and flags.
- ◆ Add an attribute to an existing class.
- ◆ Delete a class or an attribute that is not in use or that has become obsolete.
- ◆ Identify and resolve potential problems.

This chapter contains information on the following topics:

- ◆ [“Extending the Schema” on page 104](#)
- ◆ [“Viewing the Schema” on page 107](#)
- ◆ [“Manually Extending the Schema” on page 108](#)
- ◆ [“Schema Flags Added in eDirectory 8.7” on page 110](#)
- ◆ [“Using the eMBox Client to Perform Schema Operations” on page 111](#)

For more detailed schema information, see the *NDS Schema Reference* (http://developer.novell.com/ndk/doc/ndslib/index.html?schem_enu/data/h4q1mn1i.html).

Extending the Schema

You can extend the schema of a tree by creating a new class or attribute. To extend the schema of your eDirectory tree, you need the Supervisor right to the entire tree.

You can extend the schema by


- ◆ [Creating a Class](#)
- ◆ [Deleting a Class](#)
- ◆ [Creating an Attribute](#)
- ◆ [Adding an Optional Attribute to a Class](#)
- ◆ [Deleting an Attribute](#)

You can extend the schema for auxiliary attributes by

- ◆ [Creating an Auxiliary Class](#)
- ◆ [Extending an Object with the Properties of an Auxiliary Class](#)
- ◆ [Modifying an Object's Auxiliary Properties](#)
- ◆ [Deleting Auxiliary Properties from an Object](#)

Creating a Class

You can add a class to your existing schema as your organizational needs change.

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Schema > Create Class.
- 3** Follow the instructions in the Create Class Wizard to define the object class.

Help is available throughout the wizard.

If you need to define custom properties to add to the object class, cancel the wizard and define the custom properties first. See [“Creating an Attribute” on page 105](#) for more information.


Deleting a Class

You can delete unused classes that aren't part of the base schema of your eDirectory tree. iManager only prevents you from deleting classes that are currently being used in locally replicated partitions.

You might also want to consider deleting a class from the schema in the following instances:


- ◆ After merging two trees and resolving class differences
- ◆ Any time a class has become obsolete

To delete a class:

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Schema > Delete Class.
- 3** Select the class you want to delete.
Only the classes that are allowed to be deleted are shown.
- 4** Click Delete.

Creating an Attribute

You can define your own custom types of attributes and add them as optional attributes to existing object classes. You can't, however, add mandatory attributes to existing classes.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click Schema > Create Attribute.
- 3 Follow the instructions in the Create Attribute Wizard to define the new attribute.
Help is available throughout the wizard.




Adding an Optional Attribute to a Class

You can add optional attributes to existing classes. This might be necessary if

- ♦ Your organization's informational needs change.
- ♦ You are preparing to merge trees.

NOTE: Mandatory attributes can only be defined while creating a class.

To add an optional attribute class:

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click Schema > Add Attribute.
- 3 Select the class you want to add an attribute to, then click OK.
- 4 In the Available Optional Attributes list, select the attributes you want to add, then click  to add these attributes to the Add These Optional Attributes list.
If you add an attribute by mistake or change your mind, select the attribute in the Add These Optional Attributes list, then click  to remove it from the list of attributes you want to add.
- 5 Click OK.

Objects you create of this class will now have the properties you added. To set values for the added properties, use the generic Other property page of the object.

TIP: You can modify an existing class by using this page to add to the Current Attributes list. You can remove only attributes you have added prior to clicking OK. You cannot remove any attribute that has been previously added and saved.


Deleting an Attribute

You can delete unused attributes that aren't part of the base schema of your eDirectory tree.

You might also want to delete an attribute from the schema in the following instances:

- ♦ After merging two trees and resolving attribute differences
- ♦ Any time an attribute has become obsolete

To delete an attribute:


- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click Schema > Delete Attribute.
- 3 Select the attribute you want to delete.
Only the attributes that are allowed to be deleted are shown.
- 4 Click Delete.

Creating an Auxiliary Class


An auxiliary class is a set of properties (attributes) added to particular eDirectory object instances rather than to an entire class of objects. For example, an e-mail application could extend the schema of your eDirectory tree to include an E-Mail Properties auxiliary class and then extend individual objects with those properties as needed.

With Schema Manager, you can define your own auxiliary classes. You can then extend individual objects with the properties defined in your auxiliary classes.

To create an auxiliary class:

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click Schema > Create Class.
- 3 Specify a class name and (optional) ASN1 ID, then click Next.
- 4 Select Auxiliary Class when setting the class flags, then click Next.
- 5 Follow the instructions in the Create Class Wizard to define the new auxiliary class.
Help is available throughout the wizard.


Extending an Object with the Properties of an Auxiliary Class

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click Schema > Object Extensions.
- 3 Specify the name and context of the object want to extend, then click OK.
- 4 Depending on whether the auxiliary class that you want to use is already listed under Current Auxiliary Class Extensions, complete the appropriate action:

Auxiliary Class Already Listed?	Action
Yes	Quit this procedure. See “Modifying an Object’s Auxiliary Properties” on page 106 instead.
No	Click Add, select the auxiliary class, then click OK.

- 5 Click Close.


Modifying an Object’s Auxiliary Properties

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Administration > Modify Object.
- 3 Specify the name and context of the object you want to modify, then click OK.
- 4 On the General tab, click the Other page.
- 5 On the screen that appears, set the attribute values you want.
 - ♦ Double-click any unvalued attributes to add them to the list of valued attributes.
 - ♦ Select a valued attribute, then click Edit to edit the attribute, or Delete to remove the attribute.

- ♦ You must know the syntax of a property to set it correctly. For more information, see [Understanding Schema Manager \(http://www.novell.com/documentation/lg/ndsv8/docui/index.html#./usnds/schm_enu/data/hnpkthb2.html\)](http://www.novell.com/documentation/lg/ndsv8/docui/index.html#./usnds/schm_enu/data/hnpkthb2.html).

6 Click Apply, then click OK.

Deleting Auxiliary Properties from an Object

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Schema > Object Extensions.
- 3** Specify the name and context of the object want to extend, then click OK.
- 4** In the list of current auxiliary class extensions, select the auxiliary class whose properties you want to delete.
- 5** Click Remove, then click OK.

This deletes all the properties added by the auxiliary class except for any that the object already had innately.

6 Click Close.



Viewing the Schema

You can view the schema to evaluate how well the schema meets your organization's informational needs. The larger and more complex your organization, the more likely it is that you need to customize the schema, but even small organizations might have unique tracking needs. Viewing the schema can help you determine what, if any, extensions you need to make to the base schema.



Viewing Class Information

The Class Information page in iManager displays information about the selected class and lets you add attributes. Most of the information displayed on the page was specified when the class was created. Some of the optional attributes might have been added later.

During class creation, if the class was specified to inherit attributes from another class, the inherited attributes are classified as they are in the parent class. For instance, if Object Class is a mandatory attribute for the parent class, then it displays on this screen as a mandatory attribute for the selected class.

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Schema > Class Information.
- 3** Select the class you want information on, then click View.
Click  for more information.

Viewing Attribute Information

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Schema > Attribute Information.
- 3** Select the attribute you want information on, then click View.
Click  for more information.

Manually Extending the Schema

You can manually extend the eDirectory schema using files with a .sch extension.

This section contains the following information:

- ◆ [“Extending the Schema on NetWare” on page 108](#)
- ◆ [“Extending the Schema on Windows” on page 108](#)
- ◆ [“Extending the Schema on Linux, Solaris, AIX, or HP-UX Systems” on page 108](#)

Extending the Schema on NetWare

Use NWConfig.nlm to extend the schema on NetWare servers. Schema files (*.sch) that come with eDirectory are installed into the sys:\system\schema directory.

- 1** At the server console, enter **nwconfig**.
- 2** Select Directory Options > Extend Schema.
- 3** Log in as a user with administrative rights.
- 4** Press F3 to specify a different path, then type **sys:\system\schema** (or the path for your *.sch file) and the name of your schema file.
- 5** Press Enter.

Extending the Schema on Windows

Use NDSCons.exe to extend the schema on Windows servers. Schema files (*.sch) that come with eDirectory are installed by default into the C:\Novell\NDS directory.

- 1** Click Start > Settings > Control Panel > Novell eDirectory Services.
- 2** Click install.dlm, then click Start.
- 3** Click Install Additional Schema Files, then click Next.
- 4** Log in as a user with administrative rights, then click OK.
- 5** Specify the schema file path and name.
- 6** Click Finish.

Extending the Schema on Linux, Solaris, AIX, or HP-UX Systems

The following sections provide information about extending the schema on Linux, Solaris, AIX, and HP-UX systems:

- ◆ [“Using the ndssch Utility to Extend the Schema on Linux, Solaris, AIX, or HP-UX” on page 108](#)
- ◆ [“Extending the RFC 2307 Schema” on page 109](#)

Using the ndssch Utility to Extend the Schema on Linux, Solaris, AIX, or HP-UX

In addition to Novell iManager, you can use ndssch, the eDirectory schema extension utility, to extend the schema on Linux, Solaris, AIX, or HP-UX systems. The attributes and classes that you specify in the schema file (.sch) will be used to modify the schema of the tree. The association between the attributes and classes are created as specified in the .sch file.

1 Use the following syntax:

```
ndssch [-h hostname[:port]] [-t tree_name] admin-FDN schemafilename...
```

```
ndssch [-h hostname[:port]] [-t tree_name] [-d] admin_FDN schemafilename
[schema_description]...
```

ndssch Parameter	Description
-h <i>hostname</i>	Name or IP address of the server that the schema is to be extended on. The schema of the tree that the specified server belongs to will be extended. This is an optional parameter if the tree is located on the host whose schema is to be extended; otherwise, it is a mandatory parameter.
<i>port</i>	The server port.
-t <i>tree_name</i>	Name of the tree that the schema is to be extended on. This is an optional parameter. The default tree name is the one specified in the /etc/nds.conf file. For more information, see “ Configuration Parameters ” in the <i>Novell eDirectory 8.7.3 Installation Guide</i> .
<i>admin-FDN</i>	Name with the full context of the user with eDirectory administrator rights to the tree.
<i>schemafilename</i>	Filename that contains information about the schema to be extended.
-d, <i>schema_description</i>	When this option is used, every schema file must be followed by a description of the schema file.

Extending the RFC 2307 Schema

The attributes and object classes defined in [RFC 2307](http://www.ietf.org/rfc/rfc2307.txt) (<http://www.ietf.org/rfc/rfc2307.txt>) are user or group related and NIS related. The user- or group-related definitions are compiled into the /usr/lib/nds-modules/schema/rfc2307-usergroup.sch file. The NIS-related definitions are compiled into the /usr/lib/nds-modules/schema/rfc2307-nis.sch file. The corresponding files in the LDIF format are also provided (/usr/lib/nds-modules/schema/rfc2307-usergroup.ldif and /usr/lib/nds-modules/schema/rfc2307-nis.ldif respectively).

You can extend the RFC 2307 schema using the ndssch utility or the ldapmodify tool.

- ◆ “[Using the ndssch Utility](#)” on page 109
- ◆ “[Using the ldapmodify Utility](#)” on page 110

Using the ndssch Utility

Enter one of the following commands:

```
ndssch -t tree .admin.novell /usr/lib/nds-schema/rfc2307-
usergroup.sch
```

or

```
ndssch -t tree .admin.novell /usr/lib/nds-schema/rfc2307-nis.sch
```

Parameter	Description
-t	Name of the tree on that the schema is to be extended on. This is an optional parameter. If this parameter is not specified, the tree name is taken from the <code>/etc/nds.conf</code> file.

Using the `ldapmodify` Utility

Enter one of the following commands:

```
ldapmodify -h -D -w -f /usr/lib/nds-schema/rfc2307-usergroup.ldif
```

or

```
ldapmodify -h -D -w -f /usr/lib/nds-schema/rfc2307-nis.ldif
```

Parameter	Description
-h <i>ldaphost</i>	Specifies an alternate host on which the LDAP server is running.
-D <i>binddn</i>	Uses <i>binddn</i> to bind to the X.500 directory. It should be a string-represented DN as defined in RFC 1779.
-w <i>passwd</i>	Uses <i>passwd</i> as the password for simple authentication.
-f <i>file</i>	Reads the entry modification information from file instead of from standard input.

Schema Flags Added in eDirectory 8.7

The `READ_FILTERED` and `BOTH_MANAGED` schema flags were added to eDirectory 8.7.

`READ_FILTERED` is used to indicate that an attribute is an LDAP OPERATIONAL attribute. LDAP uses this flag when it requests to read the schema to indicate that an attribute is “operational.” Some internally defined schema attributes now have this flag set. The LDAP “operational” definition includes three schema flags. In addition to the new `READ_FILTERED` flag, the other existing flags that are used to indicate “operational” are the `READ_ONLY` flag and the `HIDDEN` flag. If any of these flags is present on a schema definition, LDAP treats the attribute as “operational” and will not return that attribute unless specifically requested to do so.

`BOTH_MANAGED` is a new security rights enforcement mechanism. It is only meaningful on an attribute of Distinguished Name syntax. If set on such an attribute, it will require that the requesting connection have rights on both the target object and attribute and the object being referenced by the target attribute. This is an expansion of the current `WRITE_MANAGED` flag functionality. This flag is not currently set on any base schema attributes. This new security behavior will only occur on an eDirectory 8.7.x server, so for consistent behavior relating to this flag, the entire tree must be upgraded to eDirectory 8.7 or later.

Because only an eDirectory 8.7.x server will recognize these new flags, they can be set only on a schema definition by an eDirectory 8.7.x server which holds a copy of the root partition (because only servers holding root can do schema modifications). The normal installation of a new server or upgrading an existing server that doesn't hold the root partition will not successfully add these new flags to the schema in your tree.

If you want either of these new features enabled in your tree, you need to ensure that the schema is successfully extended to add these new flags. There are two ways to do this. The first option is

to choose a server that holds a writable copy of the root partition to be upgraded to eDirectory 8.7 or later. This will automatically extend the schema correctly with the new flags.

The second option is more involved and contains the following steps:

- 1** Install a new 8.7.x server or upgrade any existing server in the tree. This server does not need to hold a copy of [Root].
- 2** Manually add a copy of the root partition to this new server.
- 3** Rerun the appropriate schema extension files on that server to extend the schema:

Platform	Instructions
Windows	Load install.dlm, then click Install Additional Schema Files.
NetWare	Load nwconfig, then select Directory Options/Extend Schema.
Linux, Solaris, AIX, and HP-UX	Use the ndssch utility. See “Using the ndssch Utility to Extend the Schema on Linux, Solaris, AIX, or HP-UX” on page 108 for more information.

- 4** Install the new schema files you choose that have these new flags set.
- 5** (Optional) After the schema has synchronized, you can remove the root replica from this server.

NOTE: These new schema flags enable optional features. If you don't need or want the new functionality, the absence of these new flags on the schema definitions will not cause any problems in the normal operation of eDirectory in your tree. In the case of the READ_FILTERED flag, it would not be present on some attribute definitions; therefore, an LDAP read request for all attributes of an object might get some extra data it would not otherwise have received. Some attributes will still be treated as operational anyway because of the presence of the READ_ONLY or HIDDEN flag. The BOTH_MANAGED flag is intended only to be enabled on fully upgraded trees, because consistent operation of this feature can be achieved only in that environment.

Using the eMBox Client to Perform Schema Operations

The eDirectory Management Toolbox (eMBox) Client is a command line Java client that gives you remote access to DSSchema operations. You can use the DSSchema eMTool to synchronize schema, import remote schema, declare a new schema epoch, reset the local schema, and perform a global schema update (operations normally performed using DSRepair. For more information, see [“Maintaining the Schema” on page 214](#)).

The emboxclient.jar file is installed on your server as part of eDirectory. You can run it on any machine with a JVM. For more information on the eMBox Client, see [“Using the eMBox Command Line Client” on page 465](#).

Using the DSSchema eMTool

- 1** Run the eMBox Client in interactive mode by entering the following at the command line:

```
java -cp path_to_the_file/emboxclient.jar embox -i
```

(If you have already put the emboxclient.jar file in your class path, you only need to enter `java embox -i`.)

The eMBox Client prompt appears:

```
eMBox Client>
```

- 2** Log in to the server you want to repair by entering the following:

```
login -sserver_name_or_IP_address -pport_number
      -username.context -wpassword -n
```

The port number is usually 80 or 8008, unless you have a Web server that is already using the port. The -n option opens a nonsecure connection.

The eMBox Client indicates whether the login is successful.

- 3** Enter a repair command, using the following syntax:

```
dsschema.task options
```

For example:

dsschema.rst requests the master replica of the root of the tree to synchronize its schema to this server.

dsschema.irs -nMyTree imports remote schema from the tree named MyTree.

A space must be between each switch. The order of the switches is not important.

The eMBox Client will indicate whether the repair is successful.

See “[DSSchema eMTool Options](#)” on page 112 for more information on the DSSchema eMTool options.

- 4** Log out from the eMBox Client by entering the following command:

```
logout
```

- 5** Exit the eMBox Client by entering the following command:

```
exit
```

DSSchema eMTool Options

The following tables lists the DSSchema eMTool options. You can also use the list -tdsschema command in the eMBox Client to list the DSSchema options with details. See “[Listing eMTools and Their Services](#)” on page 469 for more information.

Option	Description
rst	Synchronizes the schema of the master replica of the root of the tree to this server.
irs -ntree_name	Imports remote schema from another tree.
dse	Declares a new schema epoch on the server that holds the master replica of root.
rls	Resets the local schema with a copy from the server with the master replica of the root partition.
gsu	Performs a global schema update to Post NetWare 5 level.
scc	Adds schema circular containment rules for the Domain class.

5

Managing Partitions and Replicas

Partitions are logical divisions of the Novell® eDirectory™ database that form a distinct unit of data in the eDirectory tree for administrators to store and replicate eDirectory information. Each partition consists of a container object, all objects contained in it, and the information about those objects. Partitions do not include any information about the file system or the directories and files contained there.

Instead of storing a copy of the entire eDirectory database on each server, you can make a copy of the eDirectory partition and store it on many servers across the network. Each copy of the partition is known as a replica. You can create any number of replicas for each eDirectory partition and store them on any server. The types of replicas include master, read/write, read-only, subordinate references, filtered read/write, and filtered read-only.

The following table describes the replica types.

Replica	Description
Master, read/write, and read-only	Contain all objects and attributes for a particular partition.
Subordinate references	Used for tree connectivity.
Filtered replicas	<p>Contains a subset of information from the entire partition, consisting of only the desired classes and attributes—which are defined by the server's replication filter. This filter is used to identify the classes and attributes allowed to pass during inbound synchronization and local changes.</p> <p>Filtered replicas allow administrators to create sparse and fractional replicas.</p> <ul style="list-style-type: none"> ◆ Sparse replicas contain only the object classes that you specify. ◆ Fractional replicas contain only the attributes you specify. <p>The functionality of filtered replicas enables fast response when the data stored in eDirectory is procured by applications. Filtered replicas also allow more replicas to be stored on a single server.</p>
Read/write filtered replicas	Allows local modifications to classes and attributes that are a subset of the server's replication filter. However, these replicas can create objects only if all mandatory attributes for the class are within the replication filter.
Read-only filtered replicas	Does not allow local modifications.

This chapter describes how to manage partitions and replicas.

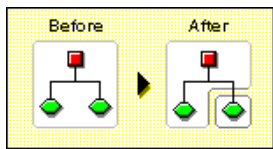
- ◆ “Creating a Partition” on page 114
- ◆ “Merging a Partition” on page 115
- ◆ “Moving Partitions” on page 116
- ◆ “Cancelling Create or Merge Partition Operations” on page 117
- ◆ “Administering Replicas” on page 117
- ◆ “Setting Up and Managing Filtered Replicas” on page 120
- ◆ “Viewing Partitions and Replicas” on page 123

Creating a Partition

When you create partitions, you make logical divisions of your tree. These divisions can be replicated and distributed among different eDirectory servers in your network.

When you create a new partition, you split the parent partition and end up with two partitions. The new partition becomes a child partition, as seen in the following illustration.

Figure 23 Before and After a Partition Split




For example, if you choose an Organizational Unit and create it as a new partition, you split the Organizational Unit and all of its subordinate objects from its parent partition.

The Organizational Unit you choose becomes the root of a new partition. The replicas of the new partition exist on the same servers as the replicas of the parent, and objects in the new partition belong to the new partition's root object.

Creating a partition might take some time, because all of the replicas need to be synchronized with the new partition information. If you attempt another partition operation while a partition is still being created, you receive a message telling you that the partition is busy.

You can look at the replica list for the new partition and know that the operation is complete when all replicas in the list are in an On state. You must manually refresh the view periodically because the states are not automatically refreshed.

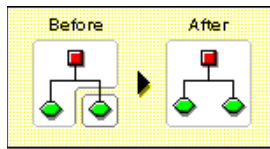
To create a partition:

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Partition and Replicas > Create Partition.
- 3** Specify the name and context of the container you want to create a new partition from, then click OK.

Merging a Partition

When you merge a partition with its parent partition, the chosen partition and its replicas combine with the parent partition. You do not delete partitions—you only merge and create partitions to define how the directory tree is split into logical divisions, as shown in the following illustration.

Figure 24 Before and After a Partition Merge



There are several reasons you might want to merge a partition with its parent:

- ◆ The directory information in the two partitions is closely related.
- ◆ You want to delete a subordinate partition, but you don't want to delete the objects in it.
- ◆ You're going to delete the objects in the partition.
- ◆ You want to delete all replicas of the partition. (Merging a partition with its parent is the only way to delete the partition's master replica.)
- ◆ After moving a container (which must be a partition root with no subordinate partitions), you don't want the container to be a partition anymore.
- ◆ You experience changes in your company organization, so you want to redesign your directory tree by changing the partition structure.

Consider keeping partitions separate if the partitions are large (contain hundreds of objects), because large partitions slow down network response time.

The root-most partition in the tree cannot be merged because it is the top partition and has no parent partition to merge with.


The partition is merged when the process is completed on the servers. The operation could take some time to complete depending on partition sizes, network traffic, server configuration, etc.

IMPORTANT: Before merging a partition, check the partition synchronization of both partitions and fix any errors before proceeding. By fixing the errors, you can isolate problems in the directory and avoid propagating the errors or creating new ones.

Make sure all servers that have replicas (including subordinate references) of the partition you want to merge are up before attempting to merge a partition. If a server is down, eDirectory won't be able to read the server's replicas and won't be able to complete the operation.

If you receive errors in the process of merging a partition, resolve the errors as they appear. Don't try to fix the error by continuing to perform operations—doing so only results in more errors.

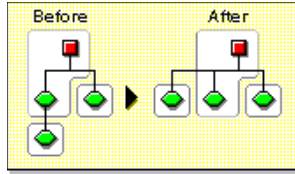
To merge a child partition with its parent partition:

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Partition and Replicas > Merge Partition.
- 3** Specify the name and context of the partition you want to merge with its parent partition, then click OK.

Moving Partitions

Moving a partition lets you move a subtree in your directory tree. You can move a partition root object (which is a container object) only if it has no subordinate partitions.

Figure 25 Before and After a Partition Move



When you move a partition, you must follow eDirectory containment rules. For example, you cannot move an Organizational Unit directly under the root of the current tree, because the root's containment rules allow Locality, Country, or Organization, but not Organizational Unit.

When you move a partition, eDirectory changes all references to the partition root object. Although the object's common name remains unchanged, the complete name of the container (and of all its subordinates) changes.

When you move a partition, you should choose the option to create an Alias object in place of the container you're moving. Doing so allows users to continue to log in to the network and find objects in the original directory location.

The Alias object that is created has the same common name as the moved container and references the new complete name of the moved container.

IMPORTANT: If you move a partition and do not create an Alias object in place of the moved partition, users who are unaware of the partition's new location cannot easily find that partition's objects in the directory tree, because they look for them in their original directory location.

This might also cause client workstations to fail at login if the workstation NAME CONTEXT parameter is set to the original location of the container in the directory tree.


Because the context of an object changes when you move it, users whose name context references the moved object need to update their NAME CONTEXT parameter so that it references the object's new name.

To automatically update users' NAME CONTEXT after moving a container object, use the NCUPDATE utility.

After moving the partition, if you don't want the partition to remain a partition, merge it with its parent partition.

Make sure your directory tree is synchronizing correctly before you move a partition. If you have any errors in synchronization in either the partition you want to move or the destination partition, do not perform a move partition operation. First, fix the synchronization errors.

To move a partition:

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Partition and Replicas > Move Partition.
- 3** Specify the name and context of the partition object you want to move in the Object Name field.
- 4** Specify the container name and context you want to move the partition to in the Move To field.
- 5** If you want to create an Alias in the old location for the partition being moved, select Create an Alias in Place of Moved Object.

This allows any operations that are dependent on the old location to continue uninterrupted until you can update those operations to reflect the new location.

- 6 Click OK.

Canceling Create or Merge Partition Operations

You can cancel a Create or Merge partition operation if the operation has not yet progressed past the stage at which the change is committed. Use this feature to back out of an operation, or if your eDirectory network returns eDirectory errors or fails to synchronize following a partition operation.

If replicas in your directory tree experience synchronization errors, an abort operation might not always solve the problem. However, you can use this feature as an initial troubleshooting option.

If a partition operation cannot be completed because a server is down (or otherwise unavailable), either make the server visible to the network so the operation can complete or attempt to abort the operation. If eDirectory cannot synchronize because the database is corrupted, you should abort any partition operation in progress.

Partition operations can take considerable time to fully synchronize across the network, depending on the number of replicas involved, the visibility of servers involved, and the existing wire traffic.

If you get an error that says a partition is busy, it doesn't mean that you should abort the operation. You can usually expect partition operations to complete within 24 hours depending on the size of the partition, connectivity issues, etc. If a particular operation fails to complete within this time frame, you should then attempt to abort the operation in progress.

Administering Replicas


Before you add or delete a replica, or change replica type, carefully plan target replica locations. See [“Guidelines for Replicating Your Tree” on page 75](#).





Adding a Replica

Add a replica to a server to provide your directory with

- ◆ Fault tolerance
- ◆ Faster access to data
- ◆ Faster access across a WAN link
- ◆ Access to objects in a set context (using bindery services)

To add a replica:

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click Partition and Replicas > Replica View.
- 3 Specify the name and context of the partition or server you want to replicate, then click OK.
- 4 Click Add Replica.
- 5 Specify the partition or server name and context.
- 6 Choose one of the following replica types:

Replica Type	Description
 Read-Write	Users will be able to both read and modify the contents of the new replica. Select this option if there are no modifiable replicas close enough to the users who manage the eDirectory objects in this partition.
 Read-Only	Users will be able to read but not modify the contents of the new replica. Select this option if there are no replicas close enough to the users who read but don't modify the eDirectory objects in this partition.
 Filtered Read-Write	Users will be able to both read and modify the contents of the new replica, and the contents will be limited to the types of eDirectory objects and properties specified in a filter.
 Filtered Read-Only	Users will be able to read but not modify the contents of the new replica, and the contents will be limited to the types of eDirectory objects and properties specified in a filter.

7 Click OK.

For more information, see [“Replica Types” on page 50](#).

Deleting a Replica

Deleting a replica removes the replica of the partition from a server.

If you want to remove a server from the directory tree, you can delete replicas from the server before removing the server. Deleting the replicas reduces the chance of having problems when removing the server.

You can also reduce synchronization traffic on the network by removing replicas. Keep in mind that you probably don't want more than six replicas of any partition.

You cannot delete a master replica or a subordinate reference.

If the replica you want to delete is a master, you have two options:



- ◆ Go to a server with another replica of the partition and make it the new master replica
This automatically changes the original master replica to a read/write replica, which you can then delete.
- ◆ Merge the partition with its parent partition
This merges the replicas of the partition with those of its parent and removes them from the servers they reside on. Merging removes partition boundaries, but not the objects. The objects continue to exist on each server which held a replica of the “joined” partition.

When you delete replicas, keep the following guidelines in mind:

- ◆ For fault tolerance, you should maintain at least three replicas of each partition on different servers.
- ◆ Deleting a replica deletes a copy of part of the directory database on the targeted server.
The database can still be accessed on other servers in the network, and the server that the replica was on still functions in eDirectory.

You cannot delete or manage subordinate reference replicas. They are created automatically on a server by eDirectory when the server contains a replica of a partition but not of that partition's child.

To delete a replica:

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Partition and Replicas > Replica View.
- 3** Specify the name and context of the partition or server that holds the replica you want to delete, then click OK.
- 4** Click  to the left of the replica you want to delete.
- 5** Click OK.

Changing a Replica Type


Change a replica type to control access to the replica information. For example, you might want to change an existing read/write replica to a read-only replica to prevent users from writing to the replica and modifying directory data.


You can change the type of a read/write or a read-only replica. You cannot change the type of a master replica, but a read/write or read-only can be changed to a master, which automatically changes the original master to a read/write replica.





Most replicas should be read/write. Read/write replicas can be written to by client operations. They send out information for synchronization when a change is made. Read-only replicas cannot be written to by client operations. However, they are updated when the replicas synchronize.

You cannot change the replica type of a subordinate reference. To place a replica of a partition on a server which currently has a subordinate reference requires an Add replica operation. A subordinate reference replica is not a complete copy of a partition. The placement and management of subordinate reference replicas is handled by eDirectory. They are created automatically on a server by eDirectory when the server contains a replica of a partition but not of that partition's child.

To change a replica type:

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Partition and Replicas > Replica View.
- 3** Specify the name and context of the partition or server that holds the replica you want to change, then click OK.
- 4** Click the replica type (in the Type column) of the replica you want to change.
- 5** Select a new replica type, then click OK.

Replica Type	Description
 Master	Users can both read and modify the contents of this replica, and the replica is the starting point for any future partitioning activity that affects this partition, such as creating or merging a subpartition. Only one master replica is allowed per partition.

Replica Type	Description
 Read-Write	Users can both read and modify the contents of the new replica. Select this option if there are no modifiable replicas close enough to the users who manage the eDirectory objects in this partition.
 Read-Only	Users can read but not modify the contents of the new replica. Select this option if there are no replicas close enough to the users who read but don't modify the eDirectory objects in this partition.
 Filtered Read-Write	Users can both read and modify the contents of the new replica, and the contents are limited to the types of eDirectory objects and properties specified in a filter.
 Filtered Read-Only	Users can read but not modify the contents of the new replica, and the contents are limited to the types of eDirectory objects and properties specified in a filter.

6 Click OK.

For more information, see [“Replica Types” on page 50](#).

Setting Up and Managing Filtered Replicas

Filtered replicas maintain a filtered subset of information from an eDirectory partition (objects or object classes along with a filtered set of attributes and values for those objects).

Administrators generally use the filtered replica capability to create an eDirectory server that holds a set of filtered replicas that contain only specific objects and attributes that they want synchronized.


To do this, iManager provides tools to create a filtered replica partition scope and filter. A scope is simply the set of partitions where you want replicas placed on a server; a replication filter contains the set of eDirectory classes and attributes you want to host on a server's set of filtered replicas. The result is an eDirectory server that can house a well-defined data set from many partitions in the tree.

The descriptions of the server's partition scope and replication filters are stored in eDirectory, and they can be managed through the Server object or the Partition and Replicas role in iManager.

- ♦ [“Using the Filtered Replica Wizard” on page 120](#)
- ♦ [“Defining a Partition Scope” on page 121](#)
- ♦ [“Setting Up a Server Filter” on page 122](#)

Using the Filtered Replica Wizard

The Filtered Replica Wizard guides you step-by-step through the setup of a server's replication filter and partition scope.

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Partition and Replicas > Filtered Replica Wizard.
- 3** Specify the server that you want to configure a filtered replica on, then click Next.

- 4 To define the classes and attributes for a filter set on the selected server, click Define the Filter Set.

The replication filter contains the set of eDirectory classes and attributes you want to host on this server's set of filtered replicas. For more information on defining a filter set, see [“Setting Up a Server Filter” on page 122](#).


- 5 Click Next.
- 6 To define the partition scope for this server, click Define the Partition Scope.
For more information on partition scopes, see [“Defining a Partition Scope” on page 121](#).
- 7 Click Next, then click Finish.

Defining a Partition Scope


A partition scope is the set of partitions where you want replicas placed on a server. The Replica View page in iManager provides a view of the hierarchy of partitions in the eDirectory tree. You can select individual partitions, a set of partitions of a given branch, or all of the partitions in the tree. You can then select the type of replicas of these partitions you want added to the server, or change existing replica types.

A server can hold both full replicas and filtered replicas. For more information, see [“Filtered Replicas” on page 53](#).


Viewing Replicas on an eDirectory Server

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click Partition and Replicas > Replica View.
- 3 Specify the name and context of server you want to view, then click OK to view the list of replicas on this server.

Adding a Filtered Replica to an eDirectory Server

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click Partition and Replicas > Replica View.
- 3 Specify the name and context of server you want to add a filtered replica to, then click OK.
- 4 Click Add Replica.
- 5 Specify the partition name and context.
- 6 Click Filtered Read-Write or Filtered Read-Only, then click OK.

Changing a Full Replica into a Filtered Replica

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click Partition and Replicas > Replica View.
- 3 Specify the name and context of the partition or server that holds the replica you want to change, then click OK.
- 4 Click the replica type (in the Type column) of the replica you want to change.
- 5 Click Filtered Read-Write or Filtered Read-Only, then click OK.

Setting Up a Server Filter


A server replication filter contains the set of eDirectory classes and attributes you want to host on a server's set of filtered replicas. You can set up a filter from any Server object. For filtered replicas, you can have only one filter per server. This means that any filter defined for an eDirectory server applies to all filtered replicas on that server. The filter, however, does not apply to full replicas.

A server's filter can be modified if required, but the operation generates a resynchronization of the replica and can thus be time consuming. Careful planning of the server's function is recommended.


You can set up or modify a server filter in either of the following ways:

- ♦ [“Using the Replica View” on page 122](#)
- ♦ [“Using the Server Object” on page 122](#)

Using the Replica View

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Partition and Replicas > Replica View.
- 3** Specify the name and context of the partition or server that holds the replica you want to change, then click OK.
- 4** Click Edit in the Filter column for the server or partition you want to modify.
- 5** Add the classes and attributes you want, then click OK.
- 6** Click Done.

Using the Server Object

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Administration > Modify Object.
- 3** Specify the name and context of the server that holds the replica you want to change, then click OK.
- 4** Click the Replica tab.
- 5** If no filter has been defined for this server, click The Filter is Empty to open the Edit Filter Dialog window, then add the classes and attributes you want.
or
Click Copy Filter From to browse for an object (such as another server) whose filter you want to copy.
- 6** To edit an existing filter, click any hyperlinked item in the filter to open the Edit Filter Dialog window, then add or remove the classes and attributes you want.


Viewing Partitions and Replicas

This section contains the following information:

- ◆ “Viewing the Partitions on a Server” on page 123
- ◆ “Viewing a Partition's Replicas” on page 123
- ◆ “Viewing Information about a Partition” on page 123
- ◆ “Viewing Partition Hierarchy” on page 124
- ◆ “Viewing Information about a Replica” on page 124

Viewing the Partitions on a Server

You can use Novell iManager to view which partitions are allocated to a server. You might want to view the partitions stored on a server if you are planning to remove a Server object from the directory tree. In this case, you can view the replicas you need to remove before removing the object.


- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Partition and Replicas > Replica View.
- 3** Enter the name and context of a Server object, then click OK.

Viewing a Partition's Replicas

This operation lets you identify


- ◆ Which servers the partition's replicas reside on
- ◆ Which server hosts the master replica of the partition
- ◆ Which servers have read/write, read-only, and subordinate reference replicas of the partition
- ◆ The state of each of the partition's replicas

To view a partition's replicas:

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Partition and Replicas > Replica View.
- 3** Enter the name and context of a partition, then click OK.

Viewing Information about a Partition

The most significant reason to view information about a partition is to see its synchronization information (last successful synchronization and last attempted synchronization).

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Partition and Replicas > View Partition Information.
- 3** Enter the name and context of a partition, then click OK.

Viewing Partition Hierarchy

You can easily view the partition hierarchy in iManager. You can expand container objects to view which partitions are parent, and which are child partitions.


Each container representing the root of a partition is marked with the following icon: .

Viewing Information about a Replica

The most significant reason to view information about a replica is to see its state. An eDirectory replica can be in various states depending on the partition or replication operations it is undergoing. The following table describes the replica states that you might see in iManager.

State	Means That the Replica Is
On	Currently not undergoing any partition or replication operations
New	Being added as a new replica on the server
Dying	Being deleted from the server
Dead	Done being deleted from the server
Master Start	Being changed to a master replica
Master Done	Done being changed to a master replica
Change Type	Being changed to a different type of replica
Locked	Locked in preparation for a partition move or repair operation
Transition Move	Starting into a partition move operation
Move	In the midst of a partition move operation
Transition Split	Starting into a partition split operation (creation of a child partition)
Split	In the midst of a partition split operation (creation of a child partition)
Join	Being merged into its parent partition
Transition On	About to return to an On state
Unknown	In a state not known to iManager

To view information about a replica:

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Partition and Replicas > Replica View.
- 3** Enter the name and context of a partition or server, then click OK.

6

Novell eDirectory Management Utilities

This chapter contains information on the following Novell® eDirectory™ utilities:

- ♦ “Novell Import Conversion Export Utility” on page 125
- ♦ “Index Manager” on page 155
- ♦ “Predicate Data” on page 159
- ♦ “eDirectory Service Manager” on page 160

Novell Import Conversion Export Utility

The Novell Import Conversion Export utility lets you

- ♦ Import data from LDIF files to an LDAP directory.
- ♦ Export data from the LDAP directory to an LDIF file.
- ♦ Migrate data between LDAP servers.
- ♦ Perform a schema compare and update.
- ♦ Load information into eDirectory using a template.
- ♦ Import schema from SCH files to an LDAP directory.

The Novell Import Conversion Export utility manages a collection of handlers that read or write data in a variety of formats. Source handlers read data; destination handlers write data. A single executable module can be both a source and a destination handler. The engine receives data from a source handler, processes the data, then passes the data to a destination handler.

For example, if you want to import LDIF data into an LDAP directory, the Novell Import Conversion Export engine uses an LDIF source handler to read an LDIF file and an LDAP destination handler to send the data to the LDAP directory server. See “[Troubleshooting LDIF Files](#)” on page 484 for more information on LDIF file syntax, structure, and debugging.

You can run the Novell Import Conversion Export client utility from the command line, from a snap-in to ConsoleOne®, or from the Import Convert Export Wizard in Novell iManager. The comma-delimited data handler, however, is available only in the command line utility and Novell iManager.

You can use the Novell Import Conversion Export utility in any of the following ways:

- ♦ “[Using the Novell iManager Import Convert Export Wizard](#)” on page 126
- ♦ “[Using the Command Line Interface](#)” on page 129

Both the wizard and the command line interface give you access to the Novell Import Conversion Export engine, but the command line interface gives you greater options for combining source and destination handlers.

The Novell Import Conversion Export utility replaces both the BULKLOAD and ZONEIMPORT utilities included with previous versions of NDS and eDirectory.


Using the Novell iManager Import Convert Export Wizard

The Import Convert Export Wizard lets you

- ◆ Import data from an LDIF, delimited text file, schema file, or LOAD file.
- ◆ Export data to an LDIF file.
- ◆ Migrate data between servers.

For information on using and accessing Novell iManager, see the *Novell iManager 2.0.x Administration Guide* (<http://www.novell.com/documentation/lg/imanager20/index.html>).

Importing Data from a File

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance > Import Convert Export Wizard.
- 3** Click Import Data from File on Disk, then click Next.
- 4** Select the type of file you want to import.
- 5** Specify the name of the file containing the data you want to import, specify the appropriate options, then click Next.


The options on this page depend on the type of file you selected. Click Help for more information on the available options.

- 6** Specify the LDAP server where the data will be imported.
- 7** Add the appropriate options, as described in the following table:

Option	Description
Server DNS name/IP address	DNS name or IP address of the destination LDAP server
Port	Integer port number of the destination LDAP server
DER File	Name of the DER file containing a server key used for SSL authentication
Login method	Authenticated Login or Anonymous Login (for the entry specified in the User DN field)
User DN	Distinguished name of the entry that should be used when binding to the server-specified bind operation
Password	Password attribute of the entry specified in the User DN field

- 8** Click Next, then click Finish.

Exporting Data to a File

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance > Import Convert Export Wizard.
- 3** Click Export Data to a File on Disk, then click Next.
- 4** Specify the LDAP server holding the entries you want to export.

Use the Advanced Settings to configure additional options for the LDAP source handler. Click Help for more information on the available options.

- 5** Add the appropriate options, as described in the following table:

Option	Description
Server DNS name/IP address	DNS name or IP address of the source LDAP server
Port	Integer port number of the source LDAP server
DER File	Name of the DER file containing a server key used for SSL authentication
Login method	Authenticated Login or Anonymous Login (for the entry specified in the User DN field)
User DN	Distinguished name of the entry that should be used when binding to the server-specified bind operation
Password	Password attribute of the entry specified in the User DN field

- 6** Click Next.

- 7** Specify the search criteria (described below) for the entries you want to export.

Option	Description
Base DN	Base distinguished name for the search request If this field is left empty, the base DN defaults to "" (empty string).
Scope	Scope of the search request
Filter	RFC 1558-compliant search filter The default is objectclass=*
Attributes	Attributes you want returned for each search entry


- 8** Click Next.

- 9** Select the export file type.

The exported file is saved in a temporary location. You can download this file at the conclusion of the Wizard.

- 10** Click Next, then click Finish.

Migrating Data between LDAP Servers

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance > Import Convert Export Wizard.
- 3** Click Migrate Data Between Servers, then click Next.
- 4** Specify the LDAP server holding the entries you want to migrate.

Use the Advanced Settings to configure additional options for the LDAP source handler. Click Help for more information on the available options.

- 5** Add the appropriate options, as described in the following table:

Option	Description
Server DNS name/IP address	DNS name or IP address of the source LDAP server
Port	Integer port number of the source LDAP server
DER file	Name of the DER file containing a server key used for SSL authentication
Login method	Authenticated Login or Anonymous Login (for the entry specified in the User DN field)
User DN	Distinguished name of the entry that should be used when binding to the server-specified bind operation
Password	Password attribute of the entry specified in the User DN field

- 6** Click Next.

- 7** Specify the search criteria (described below) for the entries you want to migrate:

Option	Description
Base DN	Base distinguished name for the search request If this field is left empty, the base DN defaults to " " (empty string).
Scope	Scope of the search request
Filter	RFC 2254-compliant search filter The default is objectclass=*
Attributes	Attributes you want returned for each search entry

- 8** Click Next.

- 9** Specify the LDAP server where the data will be migrated.

- 10** Click Next, then click Finish.

NOTE: Ensure that the schema is consistent across LDAP Services.

Using Port Numbers with More Than Four Digits

To enter an LDAP port number with more than four digits, such as 10389, with ICE, change the following entry in the c:\Program Files\Novell\Tomcat\webapps\nps\portal\modules\ICEWiz\skins\default\devices\default\ICEWizProfile_inc.jsp file:

```
<input type=text name="<%= c.var("PROFILE.SERVER_PORT") %>" value="<%=
c.var(c.var("PROFILE.SERVER_PORT")) %>" size=8 maxlength=4/>
```

to

```
<input type=text name="<%= c.var("PROFILE.SERVER_PORT") %>" value="<%=
c.var(c.var("PROFILE.SERVER_PORT")) %>" size=8 maxlength=5/>
```


Using the Command Line Interface

You can use the command line version of the Novell Import Conversion Export utility to perform the following:

- ◆ LDIF imports
- ◆ LDIF exports
- ◆ Comma-delimited data imports
- ◆ Comma-delimited data exports
- ◆ Data migration between LDAP servers
- ◆ Schema compare and update
- ◆ Load information into eDirectory using a template
- ◆ Schema imports

The Novell Import Convert Export Wizard is installed as part of Novell iManager. Both a Win32* version (`ice.exe`) and a NetWare[®] version (`ice.nlm`) are included in the installation. On Linux, Solaris, AIX, and HP-UX systems, the Import/Export utility is included in the NOVLicense package.

Novell Import Conversion Export Syntax

The Novell Import Conversion Export utility is launched with the following syntax:

```
ice general_options
-S[LDIF | LDAP | DELIM | LOAD | SCH] source_options
-D[LDIF | LDAP | DELIM] destination_options
```

or when using the schema cache:

```
ice -C schema_options
-S[LDIF | LDAP] source_options
-D[LDIF | LDAP] destination_options
```

When performing an update using the schema cache, an LDIF file is not a valid destination.

General options are optional and must come before any source or destination options. The -S (source) and -D (destination) handler sections can be placed in any order.

The following is a list of the available source and destination handlers:

- ◆ [“LDIF Source Handler Options” on page 131](#)
- ◆ [“LDIF Destination Handler Options” on page 132](#)
- ◆ [“LDAP Source Handler Options” on page 132](#)
- ◆ [“LDAP Destination Handler Options” on page 134](#)
- ◆ [“DELIM Source Handler Options” on page 135](#)
- ◆ [“DELIM Destination Handler Options” on page 136](#)
- ◆ [“SCH Source Handler Options” on page 137](#)
- ◆ [“LOAD Source Handler Options” on page 137](#)

General Options

General options affect the overall processing of the Novell Import Conversion Export engine.

Option	Description
-C	Specifies that you are using the schema cache to perform schema compare and update.
-l <i>log_file</i>	Specifies a filename where output messages (including error messages) are logged. If this option is not used, error messages are sent to ice.log. If you omit this option on Linux, Solaris, AIX, or HP-UX systems, error messages will not be logged.
-o	Overwrites an existing log file. If this flag is not set, messages are appended to the log file instead.
-e <i>LDIF_error_log_file</i>	Specifies a filename where entries that fail are output in LDIF format. This file can be examined, modified to correct the errors, then reapplied to the directory.
-p <i>URL</i>	Specifies the location of an XML placement rule to be used by the engine. Placement rules let you change the placement of an entry. See “Conversion Rules” on page 144 for more information.
-c <i>URL</i>	Specifies the location of an XML creation rule to be used by the engine. Creation rules let you supply missing information that might be needed to allow an entry to be created successfully on import. For more information, see “Conversion Rules” on page 144 .
-s <i>URL</i>	Specifies the location of an XML schema mapping rule to be used by the engine. Schema mapping rules let you map a schema element on a source server to a different but equivalent schema element on a destination server. For more information, see “Conversion Rules” on page 144 .
-b (NetWare only)	Specifies to not pause for input at the ICE console screen at the end of execution.
-h or -?	Displays command line help.

Schema Options

The schema options let you use the schema cache to perform schema compare and update operations.

Option	Description
-C -a	Updates the destination schema (adds missing schema).
-C -c <i>filename</i>	Outputs the destination schema to the specified file.
-C -n	Disables schema pre-checking.

Source Handler Options

The source handler option (-S) determines the source of the import data. Only one of the following can be specified on the command line.

Option	Description
-SLDIF	Specifies that the source is an LDIF file. For a list of supported LDIF options, see “LDIF Source Handler Options” on page 131.
-SLDAP	Specifies that the source is an LDAP server. For a list of supported LDAP options, see “LDAP Source Handler Options” on page 132
-SDELIM	Specifies that the source is a comma-delimited data file. For a list of supported DELIM options, see “DELIM Source Handler Options” on page 135.
-SSCH	Specifies that the source is a schema file. For a list of supported SCH options, see “SCH Source Handler Options” on page 137
-SLOAD	Specifies that the source is a DirLoad template. For a list of supported LOAD options, see “LOAD Source Handler Options” on page 137.

Destination Handler Options

The destination handler option (-D) specifies the destination of the export data. Only one of the following can be specified on the command line.

Option	Description
-DLDIF	Specifies that the destination is an LDIF file. For a list of supported options, see “LDIF Destination Handler Options” on page 132.
-DLdap	Specifies that the destination is an LDAP server. For a list of supported options, see “LDAP Destination Handler Options” on page 134.
-DDELIM	Specifies that the destination is a comma-delimited file. For a list of supported options, see “DELIM Destination Handler Options” on page 136.

LDIF Source Handler Options

The LDIF source handler reads data from an LDIF file, then sends it to the Novell Import Conversion Export engine.

Option	Description
-f <i>LDIF_file</i>	Specifies a filename containing LDIF records read by the LDIF source handler and sent to the engine. If you omit this option on Linux, Solaris, AIX, or HP-UX systems, the input will be taken from stdin.
-a	If the records in the LDIF file are content records (that is, they contain no changetypes), they will be treated as records with a changetype of add.
-c	Prevents the LDIF source handler from stopping on errors. This includes errors on parsing LDIF and errors sent back from the destination handler. When this option is set and an error occurs, the LDIF source handler reports the error, finds the next record in the LDIF file, then continues.
-n	Does not perform update operations, but prints what would be done. When this option is set, the LDIF source handler parses the LDIF file but does not send any records to the Novell Import Conversion Export engine (or to the destination handler).
-m	If the records in the LDIF file are content records (that is, they contain no changetypes), they will be treated as records with a changetype of modify.
-x	If the records in the LDIF file are content records (that is, they contain no changetypes), they will be treated as records with a changetype of delete.
-R <i>value</i>	Specifies the range of records to be processed.
-v	Enables the verbose mode of the handler.

LDIF Destination Handler Options

The LDIF destination handler receives data from the Novell Import Conversion Export engine and writes it to an LDIF file.

Option	Description
-f <i>LDIF_file</i>	Specifies the filename where LDIF records can be written. If you omit this option on Linux, Solaris, AIX, or HP-UX systems, the output will go to stdout.
-B	Do not suppress printing of binary values.
-b	Do not base64 encode LDIF data.

LDAP Source Handler Options

The LDAP source handler reads data from an LDAP server by sending a search request to the server. It then sends the search entries it receives from the search operation to the Novell Import Conversion Export engine.

Option	Description
-s <i>server_name</i>	Specifies the DNS name or IP address of the LDAP server that the handler will send a search request to. The default is the local host.

Option	Description
<i>-p port</i>	Specifies the integer port number of the LDAP server specified by <i>server_name</i> . The default is 389. For secure operations, the default port is 636.
<i>-d DN</i>	Specifies the distinguished name of the entry that should be used when binding to the server-specified bind operation.
<i>-w password</i>	Specifies the password attribute of the entry specified by <i>DN</i> .
<i>-W</i>	Prompts for the password of the entry specified by <i>DN</i> . This option is applicable only for Linux, Solaris, AIX, and HP-UX.
<i>-F filter</i>	Specifies an RFC 1558-compliant search filter. If you omit this option, the search filter defaults to <code>objectclass=*</code> .
<i>-n</i>	Does not actually perform a search, but shows what search would be performed.
<i>-a attribute_list</i>	Specifies a comma-separated list of attributes to retrieve as part of the search. In addition to attribute names, there are three other values: <ul style="list-style-type: none"> ◆ Get no attributes (1.1) ◆ All user attributes (*) ◆ An empty list gets all nonoperational attributes If you omit this option, the attribute list defaults to the empty list.
<i>-o attribute_list</i>	Specifies a comma-separated list of attributes to be omitted from the search results received from the LDAP server before they are sent to the engine. This option is useful in cases where you want to use a wildcard with the <i>-a</i> option to get all attributes of a class and then remove a few of them from the search results before passing the data on to the engine. For example, <i>-a* -o telephoneNumber</i> searches for all user-level attributes and filters the telephone number from the results.
<i>-R</i>	Specifies to not automatically follow referrals. The default is to follow referrals with the name and password given with the <i>-d</i> and <i>-w</i> options.
<i>-e value</i>	Specifies which debugging flags should be enabled in the LDAP client SDK. For more information, see "Using LDAP SDK Debugging Flags" on page 495 .
<i>-b base_DN</i>	Specifies the base distinguished name for the search request. If this option is omitted, the base DN defaults to "" (empty string).
<i>-c search_scope</i>	Specifies the scope of the search request. Valid values are the following: <ul style="list-style-type: none"> ◆ One: Searches only the immediate children of the base object. ◆ Base: Searches only the base object entry itself. ◆ Sub: Searches the LDAP subtree rooted at and including the base object. If you omit this option, the search scope defaults to Sub.

Option	Description
-r <i>deref_aliases</i>	<p>Specifies the way aliases should be dereferenced during the search operation. Values include the following:</p> <ul style="list-style-type: none"> ◆ Never: Prevents the server from dereferencing aliases. ◆ Always: Causes aliases to be dereferenced when locating the base object of the search and when evaluating entries that match the search filter. ◆ Search: Causes aliases to be dereferenced when applying the filter to entries within the scope of the search after the base object has been located, but not when locating the base object itself. ◆ Find: Causes aliases to be dereferenced when locating the base object of the search, but not when actually evaluating entries that match the search filter. <p>If you omit this option, the alias dereferencing behavior defaults to Never.</p>
-l <i>time_limit</i>	Specifies a time limit (in seconds) for the search.
-z <i>size_limit</i>	Specifies the maximum number of entries to be returned by the search.
-V <i>version</i>	Specifies the LDAP protocol version to be used for the connection. It must be 2 or 3. If this option is omitted, the default is 3.
-v	Enables verbose mode of the handler.
-L <i>filename</i>	Specifies a file in DER format containing a server key used for SSL authentication.
-A	Retrieves attribute names only. Attribute values are not returned by the search operation.
-t	Prevents the LDAP handler from stopping on errors.
-m	LDAP operations will be modifies.
-x	LDAP operations will be deletes.
-k	Uses SSL to connect.
-M	Enables the Manage DSA IT control.
-MM	Enables the Manage DSA IT control, and makes it critical.

LDAP Destination Handler Options

The LDAP destination handler receives data from the Novell Import Conversion Export engine and sends it to an LDAP server in the form of update operations to be performed by the server.

For information about hashed password in an LDIF file, see [“Hashed Password Representation in LDIF Files” on page 490](#).

Option	Description
-s <i>server_name</i>	Specifies the DNS name or IP address of the LDAP server that the handler will send a search request to. The default is the local host.

Option	Description
-p <i>port</i>	Specifies the integer port number of the LDAP server specified by <i>server_name</i> . The default is 389. For secure operations, the default port is 636.
-d <i>DN</i>	Specifies the distinguished name of the entry that should be used when binding to the server-specified bind operation.
-w <i>password</i>	Specifies the password attribute of the entry specified by <i>DN</i> .
-W	Prompts for the password of the entry specified by <i>DN</i> . This option is applicable only for Linux, Solaris, AIX, and HP-UX.
-B	Use this option if you do not want to use asynchronous LDAP Bulk Update/Replication Protocol (LBURP) requests for transferring update operations to the server. Instead, use standard synchronous LDAP update operation requests. For more information, see “LDAP Bulk Update/Replication Protocol” on page 152 .
-F	Allows the creation of forward references. When an entry is going to be created before its parent exists, a placeholder called a <i>forward reference</i> is created for the entry's parent to allow the entry to be successfully created. If a later operation creates the parent, the forward reference is changed into a normal entry.
-I	Stores password values using the simple password method of the Novell Modular Authentication Service (NMASTM). Passwords are kept in a secure location in the directory, but key pairs are not generated until they are actually needed for authentication between servers.
-e <i>value</i>	Specifies which debugging flags should be enabled in the LDAP client SDK. For more information, see “Using LDAP SDK Debugging Flags” on page 495 .
-V <i>version</i>	Specifies the LDAP protocol version to be used for the connection. It must be 2 or 3. If this option is omitted, the default is 3.
-L <i>filename</i>	Specifies a file in DER format containing a server key used for SSL authentication.
-k	Uses SSL to connect.
-M	Enables the Manage DSA IT control.
-MM	Enables the Manage DSA IT control, and makes it critical.

DELIM Source Handler Options

The DELIM source handler reads data from a comma-delimited data file, then sends it to the destination handler.

Option	Description
-f <i>filename</i>	Specifies a filename containing comma-delimited records read by the DELIM source handler and sent to the destination handler.

Option	Description
-F <i>value</i>	Specifies a filename containing the attribute data order for the file specified by -f. If this option is not specified, you must enter this information directly using -t. See “Performing a Comma-Delimited Import” on page 141 for more information.
-t <i>value</i>	Comma-delimited list of attributes specifying the attribute data order for the file specified by -f. Either this option or -F must be specified. See “Performing a Comma-Delimited Import” on page 141 for more information.
-c	Prevents the DELIM source handler from stopping on errors. This includes errors on parsing comma-delimited data files and errors sent back from the destination handler. When this option is set and an error occurs, the DELIM source handler reports the error, finds the next record in the comma-delimited data file, then continues.
-n <i>value</i>	Specifies the LDAP naming attribute for the new object. This attribute must be contained in the attribute data you specify using -F or -t.
-l <i>value</i>	Specifies the path to append the RDN to (such as o=myCompany). If you are passing the DN, this value is not necessary.
-o <i>value</i>	Comma-delimited list of object classes (if none is contained in your input file) or additional object classes such as auxiliary classes. The default value is inetorgperson.
-i <i>value</i>	Comma-delimited list of columns to skip. This value is an integer specifying the number of the column to skip. For example, to skip the third and fifth columns, specify i3,5.
-d <i>value</i>	Specifies the delimiter. The default delimiter is a comma (,). The following values are special case delimiters: [q] = quote (a single " as the delimiter) [t] = tab For example, to specify a tab as a delimiter, you would pass -d[t].
-q <i>value</i>	Specifies the secondary delimiter. The default secondary delimiter is single quotes ('). The following values are special case delimiters: [q] = quote (a single " as the delimiter) [t] = tab For example, to specify a tab as a delimiter, you would pass -d[t].
-v	Runs in verbose mode.

DELIM Destination Handler Options

The DELIM destination handler receives data from the source handler and writes it to a comma-delimited data file.

Option	Description
<i>-f filename</i>	Specifies the filename where comma-delimited records can be written.
<i>-F value</i>	Specifies a filename containing the attribute data order for the source data. If this option is not specified, you must enter this information directly using <i>-t</i> .
<i>-t value</i>	Comma-delimited list of attributes specifying the attribute data order for the source data. Either this option or <i>-F</i> must be specified.
<i>-l value</i>	Can be either RDN or DN. Specifies whether the driver should place the entire DN or just the RDN in the data. RDN is the default value.
<i>-d value</i>	Specifies the delimiter. The default delimiter is a comma (,). The following values are special case delimiters: [q] = quote (a single " as the delimiter) [t] = tab For example, to specify a tab as a delimiter, you would pass <i>-d[t]</i> .
<i>-q value</i>	Specifies the secondary delimiter. The default secondary delimiter is single quotes ('). The following values are special case delimiters: [q] = quote (a single " as the delimiter) [t] = tab For example, to specify a tab as a delimiter, you would pass <i>-d[t]</i> .
<i>-n value</i>	Specifies a naming attribute to be appended during import, for example, <i>cn</i> .

SCH Source Handler Options

The SCH handler reads data from a legacy NDS or eDirectory schema file (files with a *.sch extension), then sends it to the Novell Import Conversion Export engine. You can use this handler to implement schema-related operations on an LDAP Server, such as extensions using a *.sch file as input.

The SCH handler is a source handler only. You can use it to import *.sch files into an LDAP server, but you cannot export *.sch files.

The options supported by the SCH handler are shown in the following table.

Option	Description
<i>-f filename</i>	Specifies the full path name of the *.sch file.
<i>-c</i>	(Optional) Prevents the SCH handler from stopping on errors.
<i>-v</i>	(Optional) Run in verbose mode.

LOAD Source Handler Options

The DirLoad handler generates eDirectory information from commands in a template. This template file is specified with the *-f* argument and contains the attribute specification information and the program control information.

Option	Description
<code>-f filename</code>	Specifies the template file containing all attribute specification and all control information for running the program.
<code>-c</code>	Continues to the next record if an error is reported.
<code>-v</code>	Runs in verbose mode.
<code>-r</code>	Changes the request to a delete request so the data is deleted instead of added. This allows you to remove records that were added using a DirLoad template.
<code>-m</code>	Indicates that modify requests will be in the template file.

Attribute Specifications determines the context of new objects.

See the following sample attribute specification file:

```
givenname: $R(first.txt)
initials: $R(initial.txt)
sn: $R(last.txt)
dn:cn=$A(givenname,%1s)$A(initials,%1s)$A(sn),ou=dev,ou=ds,o=novell
objectclass: inetorgperson
telephonenumber: 1-800-$N(1-999,%03d)-$C(%04d)
title: $R(titles.txt)
locality: Our location
```

The format of the attribute specification file resembles an LDIF file, but allows some powerful constructs to be used to specify additional details and relationships between the attributes.

Unique Numeric Value inserts a numeric value that is unique for a given object into an attribute value.

Syntax: `$C[(<format>)]`

The optional `<format>` specifies a print format that is to be applied to the value. Note that if no format is specified, the parenthesis cannot be used either:

```
$C
$C(%d)
$C(%04d)
```

The plain `$C` inserts the current numeric value into an attribute value. This is the same as `$C(%d)` because “%d” is the default format that the program uses if none was specified. The numeric value is incremented after each object, so if you use `$C` multiple times in the attribute specification, the value is the same within a single object. The starting value can be specified in the settings file by using the `!COUNTER=value` syntax.

Random Numeric Value inserts a random numeric value into an attribute value using the following syntax:

`$N(<low>-<high>[,<format>])`

`<low>` and `<high>` specify the lower and upper bounds, respectively, that are used as a random number is generated. The optional `<format>` specifies a print format that is to be applied to a value from the list.

```
$N(1-999)
$N(1-999,%d)
$N(1-999,%03d)
```

Random String Value From a List inserts a randomly selected string from a specified list into an attribute value using the following syntax:

```
$R(<filename[,<format]])
```

The *<filename* specifies a file that contains a list of values. This can be an absolute or relative path to a file. Several files containing the lists are included with this package. The values are expected to be separated by a newline character.

The optional *<format* specifies a print format that is to be applied to a value from the list.

```
$A(givename)
$A(givename,%s)
$A(givename,%.1s)
```

It is important to note that no forward references are allowed. Any attribute whose value you are going to use must precede the current attribute in the attribute specification file. In the example below, the cn as part of the dn is constructed from givename, initials, and sn; therefore, these attributes must precede the dn in the settings file.

```
givename: $R(first.txt)
initials: $R(initial.txt)
sn: $R(last.txt)
dn:o=novell,ou=dev,ou=ds,cn=$A(givename,%.1s)$A(initials,%.1s)$A(sn)
```

The dn receives special handling in the LDIF file: no matter what the location of dn is in the settings, it will be written first (as per LDIF syntax) to the LDIF file. All other attributes are written in the order they appear.

Control Settings provide some additional controls for the object creation. All controls have an exclamation point (!) as the first character on the line to separate them from attribute settings. The controls can be placed anywhere in the file.

```
!COUNTER=300
!OBJECTCOUNT=2
!CYCLE=title
!UNICYCLE=first,last
!CYCLE=ou,BLOCK=10
```

- ◆ Counter

Provides the starting value for the unique counter value. The counter value is inserted to any attribute with the \$C syntax.

- ◆ Object Count

OBJECTCOUNT determines how many objects are created from the template.

- ◆ Cycle

CYCLE can be used to modify the behavior of pulling random values from the files (\$R-syntax). This setting has three different values.

```
!CYCLE=title
```

Anytime the list named “title” is used, the next value from the list is pulled rather than randomly selecting a value. After all values have been consumed in order, the list starts from the beginning again.

```
!CYCLE=ou,BLOCK=10
```

Each value from list “ou” is to be used 10 times before moving to the next value.

The most interesting variant of the CYCLE control setting is UNICYCLE. It specifies a list of sources that are cycled through in left-to-right order, allowing you to create guaranteed unique values if desired. If this control is used, the OBJECTCOUNT control is used only to limit the number of objects to the maximum number of unique objects that can be created from the lists. In other words, if the lists that are part of UNICYCLE can produce 15000 objects, then OBJECTCOUNT can be used to reduce that number, but not to increase it.

For example, assume that the givenname file contains two values (Doug and Karl) and the sn file contains three values (Hoffman, Schultz, and Grieger). With the control setting !UNICYCLE=givenname,sn and attribute definition cn: \$R(givenname) \$R(sn), the following cns are created:

```
cn: Doug Hoffmancn
cn: Karl Hoffmancn
cn: Doug Schultzcncn
cn: Karl Schultzcncn
cn: Doug Griegercncn
cn: Karl Grieger
```

Examples

Listed below are sample commands that can be used with the Novell Import Conversion Export command line utility for the following functions:

- ◆ [“Performing an LDIF Import” on page 140](#)
- ◆ [“Performing an LDIF Export” on page 140](#)
- ◆ [“Performing a Comma-Delimited Import” on page 141](#)
- ◆ [“Performing a Comma-Delimited Export” on page 141](#)
- ◆ [“Performing a Data Migration between LDAP Servers” on page 141](#)
- ◆ [“Performing a Schema Import” on page 141](#)
- ◆ [“Performing a LOAD File Import” on page 142](#)

Performing an LDIF Import

To perform an LDIF import, combine the LDIF source and LDAP destination handlers, for example:

```
ice -S LDIF -f entries.ldif -D LDAP -s server1.acme.com -p 389 -d
cn=admin,c=us -w secret
```

This command line reads LDIF data from entries.ldif and sends it to the LDAP server server1.acme.com at port 389 using the identity cn=admin,c=us, and the password “secret.”

Performing an LDIF Export

To perform an LDIF export, combine the LDAP source and LDIF destination handlers. For example:

```
ice -S LDAP -s server1.acme.com -p 389 -d cn=admin,c=us -w password -F
objectClass=* -c sub -D LDIF -f server1.ldif
```

This command line performs a subtree search for all objects in the server `server1.acme.com` at port 389 using the identity `cn=admin,c=us` and the password “password” and outputs the data in LDIF format to `server1.ldif`.

Performing a Comma-Delimited Import

To perform a comma-delimited import, use a command similar to the following:

```
ice -S DELIM -f/tmp/in.csv -F /tmp/order.csv -ncn -lo=acme -D LDAP -s
server1.acme.com -p389 -d cn=admin,c=us -w secret
```

This command reads comma-delimited values from the `/tmp/in.csv` file and reads the attribute order from the `/tmp/order.csv` file. For each attribute entry in `in.csv`, the attribute type is specified in `order.csv`. For example, if `in.csv` contains

```
pat,pat,engineer,john
```

then `order.csv` would contain

```
dn,cn,title,sn
```

The information in `order.csv` could be input directly using the `-t` option.

The data is then sent to the LDAP server `server1.acme.com` at port 389 using the identity `cn=admin,c=us`, and password “secret”.

This example specifies that `cn` should become the new DN for this object using the `-n` option, and this object was added to the organization container `acme` using the `-l` option.

Performing a Comma-Delimited Export

To perform a comma-delimited export, use a command similar to the following:

```
ice -S LDAP -s server1.acme.com -p 389 -d cn=admin,c=us -w password -l
objectClass=* -c sub -D DELIM -f /tmp/server1.csv -F order.csv
```

This command line performs a subtree search for all objects in the server `server1.acme.com` at port 389 using the identity `cn=admin,c=us` and the password “password” and outputs the data in comma-delimited format to the `/tmp/server1.csv` file.

Performing a Data Migration between LDAP Servers

To perform a data migration between LDAP servers, combine the LDAP source and LDAP destination handlers. For example:

```
ice -S LDAP -s server1.acme.com -p 389 -d cn=admin,c=us -w password -F
objectClass=* -c sub -D LDAP -s server2.acme.com -p 389 -d cn=admin,c=us -w
secret
```

This particular command line performs a subtree search for all objects in the server `server1.acme.com` at port 389 using the identity `cn=admin,c=us` and the password “password” and sends it to the LDAP server `server2.acme.com` at port 389 using the identity `cn=admin,c=us` and the password “secret.”

Performing a Schema Import

To perform a schema file import, use a command similar to the following:

```
ice -S SCH -f $HOME/myfile.sch -D LDAP -s myserver -d cn=admin,o=novell -w
passwd
```

This command line reads schema data from myfile.sch and sends it to the LDAP server myserver using the identity cn=admin,o=novell and the password "passwd."

Performing a LOAD File Import

To perform a LOAD file import, use a command similar to the following:

```
ice -S LOAD -f attrs -D LDIF -f new.ldf
```

In this example, the contents of the attribute file attrs is as follows:

```
#####
# DirLoad 1.00
#####
!COUNTER=300
!OBJECTCOUNT=2
#-----
# ATTRIBUTE TEMPLATE
#-----

objectclass: inetorgperson
givenname: $R(first.txt)
initials: $R(initial.txt)
sn: $R(last.txt)
dn: cn=$A(givenname,%1s)$A(initials,%1s)$A(sn),ou=$R(ou),ou=dev,o=novell,
telephonenumber: 1-800-$N(1-999,%03d)-$C(%04d)
title: $R(titles)
```

Running the previous command from a command prompt produces the following LDIF file:

```
version: 1
dn: cn=JohnBBill,ou=ds,ou=dev,o=novell
changetype: add
objectclass: inetorgperson
givenname: John
initials: B
sn: Bill
telephonenumber: 1-800-290-0300
title: Amigo

dn: cn=BobJAmy,ou=ds,ou=dev,o=novell
changetype: add
```

```

objectclass: inetorgperson
givenname: Bob
initials: J
sn: Amy
telephonenumber: 1-800-486-0301
title: Pomo

```

Running the following command from a command prompt sends the data to an LDAP server via the LDAP Handler:

```
ice -S LOAD -f attrs -D LDAP -s www.novell.com -d cn=admin,o=novell -w admin
```

If the previous template file is used, but the following command line is used, all of the records that were added with the above command will be deleted.

```
ice -S LOAD -f attrs -r -D LDAP -s www.novell.com -d cn=admin,o=novell -w admin
```

If you want to use `-m` to modify, the following is an example of how to modify records:

```

# =====
# DirLoad 1.00
# =====
!COUNTER=300
!OBJECTCOUNT=2
#-----
# ATTRIBUTE TEMPLATE
# -----
dn: cn=$R(first),%.1s) ($R(initial),%.1s) $R(last),ou=$R(ou),ou=dev,o=novell
delete: givenname
add: givenname
givenname: test1
replace: givenname
givenname: test2
givenname: test3

```

If the following command line is used where the `attrs` file contains the data above:

```
ice -S LOAD -f attrs -m -D LDIF -f new.ldf
```

then the results would be the following LDIF data:

```
version: 1
```

```
dn: cn=BillTSmith,ou=ds,ou=dev,o=novell
changetype: modify
delete: givenname
-
add: givenname
givenname: test1
-
replace: givenname
givenname: test2
givenname: test3
-
dn: cn=JohnAWilliams,ou=ldap,ou=dev,o=novell
changetype: modify
delete: givenname
-
add: givenname
givenname: test1
-
replace: givenname
givenname: test2
givenname: test3
-
```

Conversion Rules

The Novell Import Conversion Export engine lets you specify a set of rules that describe processing actions to be taken on each record received from the source handler and before the record is sent on to the destination handler. These rules are specified in XML (either in the form of an XML file or XML data stored in the directory) and solve the following problems when importing entries from one LDAP directory to another:


- ◆ Missing information
- ◆ Hierarchical differences
- ◆ Schema differences

There are three types of conversion rules:

Rule	Description
Placement	<p>Changes the placement of an entry.</p> <p>For example, if you are importing a group of users in the l=San Francisco, c=US container but you want them to be in the l=Los Angeles, c=US container when the import is complete, you could use a placement rule to do this.</p> <p>For information on the format of these rules, see “Placement Rules” on page 149.</p>
Creation	<p>Supplies missing information that might be needed to allow an entry to be created successfully on import.</p> <p>For example, assume that you have exported LDIF data from a server whose schema requires only the cn (commonName) attribute for user entries, but the server that you are importing the LDIF data to requires both the cn and sn (surname) attributes. You could use the creation rule to supply a default sn value, (such as " ") for each entry as it is processed by the engine. When the entry is sent to the destination server, it will have the required sn attribute and can be added successfully.</p> <p>For information on the format of these rules, see “Create Rules” on page 147.</p>
Schema Mapping	<p>If, when you are transferring data between servers (either directly or using LDIF), there are schema differences in the servers, you can use Schema Mapping to</p> <ul style="list-style-type: none"> ♦ Extend the schema on the destination server to accommodate the object classes and attribute types in entries coming from the source server. ♦ Map a schema element on the source server to a different but equivalent schema element on the destination server. <p>For information on the format of these rules, see “Schema Mapping Rules” on page 146.</p>

You can enable conversion rules in both the Novell eDirectory Import/Export Wizard and the command line interface. For more information on XML rules, see [“Using XML Rules” on page 146](#).

Using the Novell eDirectory Import Convert Export Wizard

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance > Import Convert Export Wizard.
- 3** Select the task you want to perform.
- 4** Under Advanced Settings, choose from the following options:

Option	Description
Schema Rules	Specifies the location of an XML schema mapping rule to be used by the engine.
Placement Rules	Specifies the location of an XML placement rule to be used by the engine.
Creation Rules	Specifies the location of an XML creation rule to be used by the engine.

- 5** Click Next.
- 6** Follow the online instructions to finish your selected task.

Using the Command Line Interface

You can enable conversion rules with the `-p`, `-c`, and `-s` general options on the Novell Import Conversion Export executable. For more information, see “General Options” on page 129.

Option	Description
<code>-p URL</code>	Location of an XML placement rule to be used by the engine.
<code>-c URL</code>	Location of an XML creation rule to be used by the engine.
<code>-s URL</code>	Location of an XML schema mapping rule to be used by the engine.

For all three options, *URL* must be one of the following:

- ◆ A URL of the following format:

```
file://[path/]filename
```

The file must be on the local file system.

- ◆ An RFC 2255-compliant LDAP URL that specifies a base-level search and an attribute list consisting of a single attribute description for a singled-valued attribute type.

Using XML Rules

The Novell Import Conversion Export conversion rules use the same XML format as DirXML[®]. For more information on DirXML, see the *DirXML Administration Guide* (<http://www.novell.com/documentation/lg/dirxml11a/index.html>).

Schema Mapping Rules

The `<attr-name-map>` element is the top-level element for the schema mapping rules. Mapping rules determine how the import schema interacts with the export schema. They associate specified import class definitions and attributes with corresponding definitions in the export schema.

Mapping rules can be set up for attribute names or class names.

- ◆ For an attribute mapping, the rule must specify that it is an attribute mapping, a name space (`nds-name` is the tag for the source name), the name in the eDirectory name space, then the other name space (`app-name` is the tag for the destination name) and the name in that name space. It can specify that the mapping applies to a specific class or it can be applied to all classes with the attribute.
- ◆ For a class mapping, the rule must specify that it is a class mapping rule, a name space (eDirectory or the application), the name in that name space, then the other name space and the name in that name space.

The following is the formal DTD definition of schema mapping rules:

```
<!ELEMENT attr-name-map (attr-name | class-name)*>

<!ELEMENT attr-name (nds-name, app-name)>
<!ATTLIST attr-name
    class-name    CDATA    #IMPLIED>

<!ELEMENT class-name (nds-name, app-name)>

<!ELEMENT nds-name (#PCDATA)>

<!ELEMENT app-name (#PCDATA)>
```

You can have multiple mapping elements in the file. Each element is processed in the order that it appears in the file. If you map the same class or attribute more than once, the first mapping takes precedence.

The following examples illustrate how to create a schema mapping rule.

Schema Rule 1: The following rule maps the source's surname attribute to the destination's sn attribute for the inetOrgPerson class.

```
<attr-name-map>
  <attr-name class-name="inetOrgPerson">
    <nds-name>surname</nds-name>
    <app-name>sn</app-name>
  </attr-name>
</attr-name-map>
```

Schema Rule 2: The following rule maps the source's inetOrgPerson class definition to the destination's User class definition.

```
<attr-name-map>
  <class-name>
    <nds-name>inetOrgPerson</nds-name>
    <app-name>User</app-name>
  </class-name>
</attr-name-map>
```

Schema Rule 3: The following example contains two rules. The first rule maps the source's Surname attribute to the destination's sn attribute for all classes that use these attributes. The second rule maps the source's inetOrgPerson class definition to the destination's User class definition.

```
<attr-name-map>
  <attr-name>
    <nds-name>surname</nds-name>
    <app-name>sn</app-name>
  </attr-name>
  <class-name>
    <nds-name>inetOrgPerson</nds-name>
    <app-name>User</app-name>
  </class-name>
</attr-name-map>
```

Example Command: If the schema rules are saved to an srl.xml file, the following command instructs the utility to use the rules while processing the lentry.ldf file and to send the results to a destination file, outt1.ldf.

```
ice -o -sfile://srl.xml -SLDIF -flentry.ldf -c -DLDIF
-foutt1.ldf
```

Create Rules

Create rules specify the conditions for creating a new entry in the destination directory. They support the following elements:

- ♦ **Required Attributes** specifies that an add record must have values for all of the required attributes, or else the add fails. The rule can supply a default value for a required attribute. If a record does not have a value for the attribute, the entry is given the default value. If the record has a value, the record value is used.
- ♦ **Matching Attributes** specifies that an add record must have the specific attributes and match the specified values, or else the add fails.

- ◆ **Templates** specifies the distinguished name of a Template object in eDirectory. The Novell Import Conversion Export utility does not currently support specifying templates in create rules.

The following is the formal DTD definition for create rules:

```
<!ELEMENT create-rules (create-rule)*>

<!ELEMENT create-rule (match-attr*,
                      required-attr*,
                      template?) >

<!ATTLIST create-rule
  class-name      CDATA      #IMPLIED
  description     CDATA      #IMPLIED>

<!ELEMENT match-attr (value)+ >
<!ATTLIST match-attr
  attr-name      CDATA      #REQUIRED>

<!ELEMENT required-attr (value)*>
<!ATTLIST required-attr
  attr-name      CDATA      #REQUIRED>

<!ELEMENT template EMPTY>
<!ATTLIST template
  template-dn    CDATA      #REQUIRED>
```

You can have multiple create rule elements in the file. Each rule is processed in the order that it appears in the file. If a record does not match any of the rules, that record is skipped and the skipping does not generate an error.

The following examples illustrate how to format create rules.

Create Rule 1: The following rule places three conditions on add records that belong to the inetOrgPerson class. These records must have givenName and Surname attributes. They should have an L attribute, but if they don't, the create rule supplies a default value of Provo for them.

```
<create-rules>
  <create-rule class-name="inetOrgPerson">
    <required-attr attr-name="givenName"/>
    <required-attr attr-name="surname"/>
    <required-attr attr-name="L">
      <value>Provo</value>
    </required-attr>
  </create-rule>
</create-rules>
```

Create Rule 2: The following create rule places three conditions on all add records, regardless of their base class:

- ◆ The record must contain a givenName attribute. If it doesn't, the add fails.
- ◆ The record must contain a Surname attribute. If it doesn't, the add fails.
- ◆ The record must contain an L attribute. If it doesn't, the attribute is set to a value of Provo.

```
<create-rules>
  <create-rule>
    <required-attr attr-name="givenName"/>
    <required-attr attr-name="Surname"/>
    <required-attr attr-name="L">
```

```

        <value>Provo</value>
      </required-attr>
    </create-rule>
  </create-rules>

```

Create Rule 3: The following create rule places two conditions on all records, regardless of base class:

- ◆ The rule checks to see if the record has a uid attribute with a value of ratuid. If it doesn't, the add fails.
- ◆ The rule checks to see if the record has an L attribute. If it does not have this attribute, the L attribute is set to a value of Provo.

```

<create-rules>
  <create-rule>
    <match-attr attr-name="uid">
      <value>cn=ratuid</value>
    </match-attr>
    <required-attr attr-name="L">
      <value>Provo</value>
    </required-attr>
  </create-rule>
</create-rules>

```

Example Command: If the create rules are saved to an crl.xml file, the following command instructs the utility to use the rules while processing the lentry.ldf file and to send the results to a destination file, outt1.ldf.

```

ice -o -cfile://crl.xml -SLDIF -flentry.ldf -c -DLDIF
-foutt1.ldf

```

Placement Rules

Placement rules determine where an entry is created in the destination directory. They support the following conditions for determining whether the rule should be used to place an entry:

- ◆ **Match Class:** If the rule contains any match class elements, an objectClass specified in the record must match the class-name attribute in the rule. If the match fails, the placement rule is not used for that record.
- ◆ **Match Attribute:** If the rule contains any match attribute elements, the record must contain an attribute value for each of the attributes specified in the match attribute element. If the match fails, the placement rule is not used for that record.
- ◆ **Match Path:** If the rule contains any match path elements, a portion of the record's dn must match the prefix specified in the match path element. If the match fails, the placement rule is not used for that record.

The last element in the rule specifies where to place the entry. The placement rule can use zero or more of the following:

- ◆ **PCDATA** uses parsed character data to specify the DN of a container for the entries.
- ◆ **Copy the Name** specifies that the naming attribute of the old DN is used in the entry's new DN.
- ◆ **Copy the Attribute** specifies the naming attribute to use in the entry's new DN. The specified naming attribute must be a valid naming attribute for the entry's base class.
- ◆ **Copy the Path** specifies that the source DN should be used as the destination DN.

- ◆ **Copy the Path Suffix** specifies that the source DN, or a portion of its path, should be used as the destination DN. If a match-path element is specified, only the part of the old DN that does not match the prefix attribute of the match-path element is used as part of the entry's DN.

The following is the formal DTD definition for the placement rule:

```
<!ELEMENT placement-rules (placement-rule*)>
<!ATTLIST placement-rules
    src-dn-format      (%dn-format;)      "slash"
    dest-dn-format     (%dn-format;)      "slash"
    src-dn-delims      CDATA              #IMPLIED
    dest-dn-delims     CDATA              #IMPLIED>

<!ELEMENT placement-rule (match-class*,
                           match-path*,
                           match-attr*,
                           placement)>
<!ATTLIST placement-rule
    description        CDATA              #IMPLIED>

<!ELEMENT match-class   EMPTY>
<!ATTLIST match-class
    class-name         CDATA              #REQUIRED>

<!ELEMENT match-path    EMPTY>
<!ATTLIST match-path
    prefix              CDATA              #REQUIRED>

<!ELEMENT match-attr    (value)+ >
<!ATTLIST match-attr
    attr-name           CDATA              #REQUIRED>

<!ELEMENT placement     (#PCDATA |
                           copy-name |
                           copy-attr |
                           copy-path |
                           copy-path-suffix)* >
```

You can have multiple placement-rule elements in the file. Each rule is processed in the order that it appears in the file. If a record does not match any of the rules, that record is skipped and the skipping does not generate an error.

The following examples illustrate how to format placement rules. The `src-dn-format="ldap"` and `dest-dn-format="ldap"` attributes set the rule so that the name space for the dn in the source and destination is LDAP format.

The Novell Import Conversion Export utility supports source and destination names only in LDAP format.

Placement Example 1: The following placement rule requires that the record have a base class of `inetOrgPerson`. If the record matches this condition, the entry is placed immediately subordinate to the test container and the left-most component of its source dn is used as part of its dn.

```
<placement-rules src-dn-format="ldap" dest-dn-format="ldap">
  <placement-rule>
    <match-class class-name="inetOrgPerson"></match-class>
    <placement>cn=<copy-name/>,o=test</placement>
  </placement-rule>
</placement-rules>
```

With this rule, a record with a base class of inetOrgPerson and with the following dn:

```
dn: cn=Kim Jones, ou=English, ou=Humanities, o=UofZ
```

would have the following dn in the destination directory:

```
dn: cn=Kim Jones, o=test
```

Placement Example 2: The following placement rule requires that the record have an sn attribute. If the record matches this condition, the entry is placed immediately subordinate to the test container and the left-most component of its source dn is used as part of its dn.

```
<placement-rules src-dn-format="ldap" dest-dn-format="ldap">
  <placement-rule>
    <match-attr attr-name="sn"></match-attr>
    <placement>cn=<copy-name/>,o=test</placement>
  </placement-rule>
</placement-rules>
```

With this rule, a record with the following dn and sn attribute:

```
dn: cn=Kim Jones, ou=English, ou=Humanities, o=UofZ
sn: Jones
```

would have the following dn in the destination directory:

```
dn: cn=Kim Jones, o=test
```

Placement Example 3: The following placement rule requires the record to have an sn attribute. If the record matches this condition, the entry is placed immediately subordinate to the test container and its sn attribute is used as part of its dn. The specified attribute in the copy-attr element must be a naming attribute of the entry's base class.

```
<placement-rules src-dn-format="ldap" dest-dn-format="ldap">
  <placement-rule>
    <match-attr attr-name="sn"></match-attr>
    <placement>cn=<copy-attr attr-name="sn"/>,o=test</placement>
  </placement-rule>
</placement-rules>
```

With this rule, a record with the following dn and sn attribute:

```
dn: cn=Kim Jones, ou=English, ou=Humanities, o=UofZ
sn: Jones
```

would have the following dn in the destination directory:

```
dn: cn=Jones, o=test
```

Placement Example 4: The following placement rule requires the record to have an sn attribute. If the record matches this condition, the source dn is used as the destination dn.

```
<placement-rules src-dn-format="ldap" dest-dn-format="ldap">
  <placement-rule>
    <match-attr attr-name="sn"></match-attr>
    <placement><copy-path/></placement>
  </placement-rule>
</placement-rules>
```

Placement Example 5: The following placement rule requires the record to have an sn attribute. If the record matches this condition, the entry's entire DN is copied to the test container.

```
<placement-rules src-dn-format="ldap" dest-dn-format="ldap">
  <placement-rule>
    <match-attr attr-name="sn"></match-attr>
```

```

        <placement><copy-path-suffix/>,o=test</placement>
    </placement-rule>
</placement-rules>

```

With this rule, a record with the following dn and sn attribute:

```

dn: cn=Kim Jones, ou=English, ou=Humanities, o=UofZ
sn: Jones

```

would have the following dn in the destination directory:

```

dn: cn=Kim Jones, ou=English, ou=Humanities, o=UofZ, o=test

```

Placement Example 6: The following placement rule requires the record to have an sn attribute. If the record matches this condition, the entry's entire DN is copied to the neworg container.

```

<placement-rules>
<placement-rule>
  <match-path prefix="o=engineering"/>
  <placement><copy-path-suffix/>o=neworg</placement>
</placement-rule>
</placement-rules>

```

For example:

```

dn: cn=bob,o=engineering

```

becomes

```

dn: cn=bob,o=neworg

```

Example Command: If the placement rules are saved to a pr1.xml file, the following command instructs the utility to use the rules while processing the 1entry.ldf file and to send the results to a destination file, foutt1.ldf.

```

ice -o -pfile://pr1.xml -SLDIF -f1entry.ldf -c -DLDIF
-foutt1.ldf

```

LDAP Bulk Update/Replication Protocol

The Novell Import Conversion Export utility uses the LDAP Bulk Update/Replication Protocol (LBURP) to send asynchronous requests to an LDAP server. This guarantees that the requests are processed in the order specified by the protocol and not in an arbitrary order influenced by multiprocessor interactions or the operating system's scheduler.

LBURP also lets the Novell Import Conversion Export utility send several update operations in a single request and receive the response for all of those update operations in a single response. This adds to the network efficiency of the protocol.

LBURP works as follows:

1. The Novell Import Conversion Export utility binds to an LDAP server.
2. The server sends a bind response to the client.
3. The client sends a start LBURP extended request to the server.
4. The server sends a start LBURP extended response to the client.
5. The client sends zero or more LBURP operation extended requests to the server.

These requests can be sent asynchronously. Each request contains a sequence number identifying the order of this request relative to other requests sent by the client over the same connection. Each request also contains at least one LDAP update operation.


6. The server processes each of the LBURP operation extended requests in the order specified by the sequence number and sends an LBURP operation extended response for each request.
7. After all of the updates have been sent to the server, the client sends an end LBURP extended request to the server.
8. The server sends an end LBURP extended response to the client.

The LBURP protocol lets Novell Import Conversion Export present data to the server as fast as the network connection between the two will allow. If the network connection is fast enough, this lets the server stay busy processing update operations 100% of the time because it never has to wait for Novell Import Conversion Export to give it more work to do.

The LBURP processor in eDirectory also commits update operations to the database in groups to gain further efficiency in processing the update operations. LBURP can greatly improve the efficiency of your LDIF imports over a traditional synchronous approach.

LBURP is enabled by default, but you can choose to disable it during an LDIF import.

To enable or disable LBURP during an LDIF import:

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance > Import Convert Export Wizard.
- 3 Click Import Data From File on Disk, then click Next.
- 4 Select LDIF from the File Type drop-down list, then specify the name of the LDIF file containing the data you want to import.
- 5 Click Next.
- 6 Specify the LDAP server where the data will be imported and the type of login (anonymous or authenticated).
- 7 Under Advanced Setting, select Use LBURP.
- 8 Click Next, then follow the online instructions to complete the remainder of the LDIF Import Wizard.

IMPORTANT: Because LBURP is a relatively new protocol, eDirectory servers earlier than version 8.5 (and most non-eDirectory servers) do not support it. If you are using the Novell eDirectory Import/Export Wizard to import an LDIF file to one of these servers, you must disable the LBURP option for the LDIF import to work.

You can use the command line option to enable or disable LBURP during an LDIF import. For more information, see “-B” on page 135.

Migrating the Schema between LDAP Directories

Refer to *NetWare Application Notes* (<http://www.developer.novell.com/research>) on the Novell Developer Portal for more information about migrating the schema between LDAP directories.

Improving the Speed of LDIF Imports

In cases where you have thousands or even millions of records in a single LDIF file you are importing, consider the following:

- ◆ [“Importing Directly to a Server with a Read/Write Replica” on page 154](#)
- ◆ [“Using LBURP” on page 154](#)
- ◆ [“Configuring the Database Cache” on page 154](#)
- ◆ [“Using Simple Passwords” on page 154](#)
- ◆ [“Using Indexes Appropriately” on page 155](#)

Importing Directly to a Server with a Read/Write Replica

If it's possible to do so, select a destination server for your LDIF import that has read/write replicas containing all the entries represented in the LDIF file. This will maximize network efficiency.

Avoid having the destination server chain to other eDirectory servers for updates. This can severely reduce performance. However, if some of the entries to be updated are only on eDirectory servers that are not running LDAP, you might need to allow chaining to import the LDIF file.

For more information on replicas and partition management, see [Chapter 5, “Managing Partitions and Replicas,” on page 113](#).

Using LBURP

Novell Import Conversion Export maximizes network and eDirectory server processing efficiency by using LBURP to transfer data between the wizard and the server. Using LBURP during an LDIF import greatly improves the speed of your LDIF import.

For more information on LBURP, see [“LDAP Bulk Update/Replication Protocol” on page 152](#).

Configuring the Database Cache


The amount of database cache available for use by eDirectory has a direct bearing on the speed of LDIF imports, especially as the total number of entries on the server increases. When doing an LDIF import, you might want to allocate the maximum memory possible to eDirectory during the import. After the import is complete and the server is handling an average load, you can restore your previous memory settings. This is particularly important if the import is the only activity taking place on the eDirectory server.

For more information on configuring the eDirectory database cache, see [Chapter 15, “Maintaining Novell eDirectory,” on page 427](#).

Using Simple Passwords

Novell eDirectory uses public and private key pairs for authentication. Generating these keys is a very CPU-intensive process. With eDirectory 8.7.3, you can choose to store passwords using the simple password feature of Novell Modular Authentication Service (NMAST[™]). When you do this, passwords are kept in a secure location in the directory, but key pairs are not generated until they are actually needed for authentication between servers. This greatly improves the speed for loading an object that has password information.

To enable simple passwords during an LDIF import:

- 1 In Novell iManager, click the Roles and Tasks button .

- 2** Click eDirectory Maintenance > Import Convert Export Wizard.
- 3** Click Import Data From File on Disk, then click Next.
- 4** Select LDIF from the File Type drop-down list, then enter the name of the LDIF file containing the data you want to import.
- 5** Click Next.
- 6** Specify the LDAP server where the data will be imported and the type of login (anonymous or authenticated).
- 7** Under Advanced Setting, select Store NMAS Simple Passwords/Hashed Passwords.
- 8** Click Next, then follow the online instructions to complete the remainder of the LDIF import wizard.

If you choose to store passwords using simple passwords, you must use an NMAS-aware Novell Client™ to log in to the eDirectory tree and access traditional file and print services. NMAS must also be installed on the server. LDAP applications binding with name and password will work seamlessly with the simple password feature.

For more information on NMAS, see the *Novell Modular Authentication Service Administration Guide* (<http://www.novell.com/documentation/lg/nmas23/index.html>).

Using Indexes Appropriately

Having unnecessary indexes can slow down your LDIF import because each defined index requires additional processing for each entry having attribute values stored in that index. You should make sure that you don't have unnecessary indexes before you do an LDIF import, and you might want to consider creating some of your indexes after you have finished loading the data reviewed predicate statistics to see where they are really needed.

For more information on tuning indexes, see “[Index Manager](#)” on page 155.

Index Manager

Index Manager is an attribute of the Server object that lets you manage database indexes. These indexes are used by eDirectory to significantly improve query performance.

Novell eDirectory ships with a set of indexes that provide basic query functionality. These default indexes are for the following attributes:

CN	Aliased Object Name
dc	Obituary
Given Name	Member
Surname	Reference
uniqueID	Equivalent to Me
GUID	NLS: Common Certificate
cn_SS	Revision
uniqueID_SS	extensionInfo
ldapAttributeList	ldapClassList


You can also create customized indexes to further improve eDirectory performance in your environment. For example, if your organization has implemented a new LDAP application that looks up an attribute not indexed by default, it might be useful to create an index for that attribute.

NOTE: Although indexes improve search performance, additional indexes also add to directory update time. As a general rule, create new indexes only if you suspect performance issues are related to a particular directory lookup.

Using Novell iManager, you can create or delete indexes. You can also view and manage the properties of an index, including the index name, state, type, rule, and attribute indexed.

Use the Predicate Statistics data, available only in ConsoleOne, to know what additional indexes might be valuable for your environment. See [“Predicate Data” on page 159](#).

Creating an Index

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance > Index Management.
- 3 Select a server from the list of available servers.
- 4 On the Modify Indexes page, click Create.
- 5 Enter the Index Name.

If you do not enter an index name, the attribute is automatically assigned as the index name.

IMPORTANT: The \$ character is used as a delimiter for attribute values. If you use the \$ character in your index name, you must use a preceding backslash (/) character to escape the \$ character when working with indexes via LDAP.


- 6 Select an attribute.
- 7 Select the index rule.
 - ♦ **Value** matches the entire value or the first part of the value of an attribute. For example, value matching could be used to find entries with a LastName that is equal to “Jensen” and entries with a LastName that begins with “Jen.”
 - ♦ **Presence** requires only the presence of an attribute rather than specific attribute values. A query to find all entries with a Login Script attribute would use a presence index.
 - ♦ **Substring** matches a subset of the attribute value string. For example, a query to find a LastName with “der” would return matches for Derington, Anderson, and Lauder.

A substring index is the most resource-intensive index to create and maintain.

- 8 Click OK to update the index table.
- 9 Click Apply to restart Limber as a background process and initiate the change.

Deleting an Index


Indexes might outlive their usefulness. You can delete user-defined and auto-created indexes that are no longer a benefit. Use Predicate Statistics to help you know which indexes might be less useful. See [“Predicate Data” on page 159](#) for more information.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance > Index Management.
- 3 Select a server from the list of available servers.

- 4** On the Modify Indexes page, select the user- or auto-added index you want to delete.
- 5** Click Delete to update the index table.
- 6** Click Apply to restart Limber as a background process and initiate the change.

Taking an Index Offline

During peak times you might want to tune performance by temporarily taking indexes offline. For example, to achieve additional bulk-load speed, you might want to suspend all of the user-defined indexes. Because each object addition or modification requires updating defined indexes, having all indexes active might slow down bulk-loading of data. After the bulk-load is completed, the indexes can be brought online again.

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance > Index Management.
- 3** Select a server from the list of available servers.
- 4** On the Modify Indexes page, select the indexes you want to take offline, then click Change State.


The index state changes from Online to Offline in the display table. An index can be in any of the following states:

- ♦ **Online:** Currently running.
 - ♦ **Offline:** Suspended; can be started again by clicking Bring Online.
 - ♦ **New:** Waiting to move to Online.
 - ♦ **Deleted:** Waiting to be removed from the index table.
- 5** Click Apply.

Managing Indexes on Other Servers

If you've found a particular index to be useful on one server and you see the need for this index on another server, you can copy the index definition from one server to another. In reviewing predicate data, you might also find just the opposite case: an index that was meeting a need for several servers is no longer useful on one of these servers. In that case, you could delete the index from the single server that isn't benefitting from the index.

Index Manager allows you to target a single instance of an index without impacting all instances.

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance > Index Management.
- 3** Select a server from the list of available servers.
- 4** To copy an index definition to another server on the same tree, click Modify Index Location.
- 5** Select the index definition you want to copy.

When you select an index, servers in the tree providing that index are listed.
- 6** Use the columns provided to move a copy of the index to the desired server.
- 7** Click Apply.

Using the Novell Import Conversion Export Utility to Manage Indexes

You can use the Novell Import Conversion Export utility to create or delete indexes.

You must use an LDIF file to create or delete indexes. After the LDIF file is imported, you can trigger Limber to initiate the indexing activity; otherwise, indexing takes place when Limber triggers automatically.

To specify an index in an LDIF file, you must supply values, because the following cases ignore strings that are separated by a dollar (\$) sign.

Order	String	Description
1	Index version	Reserved for future use. In eDirectory, this should always be set to zero (0).
2	Index name	Specifies the user-defined name for the index, such as .Family Name. or .Zip Code. The string should not contain the dollar (\$) sign.
3	Index state	<p>Specifies the state of the index. When defining an index, this field should be set to 2 (online). eDirectory supports the following values:</p> <ul style="list-style-type: none"> ♦ 0 - Suspended, which indicates the index is not used in queries and is not updated. ♦ 1 - Bringing Online, which indicates the index is in the process of being created. ♦ 2 - Online, which indicates the index is up and working. ♦ 3 - Pending Creation, which indicates the index has been defined and is waiting for the background process to run. <p>The background process changes the state after the building begins.</p>
4	Index rule	<p>Specifies the type of matching:</p> <ul style="list-style-type: none"> ♦ 0 - Value Matching, which optimizes queries that involve the entire value or the first part of the value. For example, a query for all entries with a surname equal to Jensen or beginning with Jen. ♦ 1 - Presence Matching, which optimizes queries that involve only the presence of an attribute. For example, a query for all entries with a surname attribute. ♦ 2 - Substring Matching, which optimizes queries that involve a match of a few characters. For example, a query for all entries with a surname containing .der. This query returns entries with the surnames of Derington, Anderson, and Lauder.

Order	String	Description
5	Index type	<p>Specifies who created the index. When defining an index, you must set this value to 0. eDirectory supports the following values:</p> <ul style="list-style-type: none"> ◆ 0 - User Defined ◆ 1 - Added on Attribute Creation ◆ 2 - Required for Operation ◆ 3 - System Index
6	Index value state	<p>Specifies the source of the index. When defining an index, set this string to 1. eDirectory supports the following values:</p> <ul style="list-style-type: none"> ◆ 0 - Uninitialized ◆ 1 - Added from Server ◆ 2 - Added from Local DIB ◆ 3 - Deleted from Local DIB ◆ 4 - Modified from Local DIB
7	Attribute name	<p>Specifies the NDS name for the attribute. Many attributes in eDirectory have both an LDAP name and an NDS name. This string requires the NDS name.</p>

Example LDIF File to Create Indexes

```
dn: cn=testServer-NDS,o=Novell
changetype: modify
add: indexDefinition
indexDefinition: 0$indexName$2$2$0$1$attributeName
```

Example LDIF File to Delete Indexes

```
dn: cn=osg-nw5-7, o=Novell
changetype: modify
delete: indexDefinition
indexDefinition: 0$indexName$2$2$0$1$attributeName
```

Predicate Data

Predicate data is a server-specific history of the objects people search for. This data and its collection are managed through the `ndsPredicateStats` object, which is created at the time of eDirectory install. The `ndsPredicateStats` object name is the server name with a `-PS` appended.

You can use predicate data to identify most frequently searched for objects, then create indexes to improve the speed of future information access.

Managing Predicate Data

The Predicate Statistics feature is not intended to run all the time. Collecting predicate statistics affects search performance. Also, lengthy accumulation of statistics can result in large databases. Use Predicate Statistics if you suspect performance issues are related to a particular directory lookup.

Use the Predicate Data properties page in ConsoleOne to manage the collection of data.

- 1 In ConsoleOne, right-click the Server object.
- 2 Click Properties > Predicate Data > Properties.
- 3 Specify the appropriate configuration for the ndsPredicateStats object.

Update Interval sets the number of seconds to wait before refreshing the data display and writing data to disk.

Advanced > Enable specifies whether the collection process should run in the background or should be turned off. If you turn off data collection, the most recently collected data will either be released from memory or, if you've selected Write to Disk, will be moved to disk.

Advanced > Write to Disk determines storage location of predicate data, either always in memory or moving from memory to disk as specified in the Update Interval.

Advanced > Display Value Text determines whether the data display will be abbreviated or complete. The abbreviated display provides enough information to determine which predicates are good candidates for indexes.

- 4 Click OK to update the object configuration.

eDirectory Service Manager

The eDirectory Service Manager provides information about available eDirectory services and their states. You can also use the Service Manager to start and stop these services.

Service Manager manages only eDirectory services. This is done with the help of the dsservcfg.xml configuration file, which lists the services to be managed on various platform. It also lets you add or remove services from the list.

You can access the eDirectory Service Manager through the following methods:

- ♦ [“Using the eMBox Client Service Manager eMTool” on page 160](#)
- ♦ [“Using the Service Manager Plug-In to Novell iManager” on page 161](#)

Using the eMBox Client Service Manager eMTool

The eDirectory Management Toolbox (eMBox) Client is a command line Java client that gives you remote access to the eDirectory Service Manager eMTool. The emboxclient.jar file is installed on your server as part of eDirectory. You can run it on any machine with a JVM. For more information on the eMBox Client, see [“Using the eMBox Command Line Client” on page 465](#).

To use the eMBox Client Service Manager eMTool:

- 1 Run the eMBox Client in interactive mode by entering the following at the command line:

```
java -cp path_to_the_file/emboxclient.jar embox -i
```


(If you have already put the `emboxclient.jar` file in your class path, you only need to enter `java embox -i`.)

The eMBox Client prompt appears:

```
eMBox Client>
```

- 2 Log in to the server that will run Service Manager by entering the following:

```
login -sserver_name_or_IP_address -pport_number
      -username.context -wpassword -n
```

The port number is usually 80 or 8008, unless you have a Web server that is already using the port. The `-n` option opens a nonsecure connection.

The eMBox Client indicates whether the login is successful.

- 3 Enter one of the following Service Manager commands:

Command	Description
<code>service.serviceList</code>	Lists the available eDirectory services.
<code>service.serviceStart -nModule_name</code>	Starts the specified eDirectory service.
<code>service.serviceStop -nModule_name</code>	Stops the specified eDirectory service.
<code>service.serviceInfo -nModule_name</code>	Displays information for the specified service.

You can also use the `list -tservice` command in the eMBox Client to list the Service Manager options with details. See [“Listing eMTools and Their Services” on page 469](#) for more information.


- 4 Log out from the eMBox Client by entering the following command:



```
logout
```




- 5 Exit the eMBox Client by entering the following command:

```
exit
```

Using the Service Manager Plug-In to Novell iManager

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance > Service Manager.
- 3 Specify the server you want to manage, then click OK.
- 4 Authenticate to the selected server, then click OK.
- 5 Use the following icons to check the status of any eDirectory service, or to start or stop a service:

Icon	Description
	A service is running.
	A service is stopped.

Icon	Description
	Starts a service.
	Stops a service.
	A service is running but you can't stop it.

7

Using Novell iMonitor 2.1

Novell® iMonitor provides cross-platform monitoring and diagnostic capability to all servers in your eDirectory™ tree. This utility lets you monitor your servers from any location on your network where a Web browser is available.

iMonitor lets you look at the eDirectory environment in depth on a partition, replica, or server basis. You can also examine what tasks are taking place, when they are happening, what their results are, and how long they are taking.

iMonitor provides a Web-based alternative or replacement for many of the Novell traditional server-based eDirectory tools such as DSBrowse, DSTrace, DSDiag, and the diagnostic features available in DSRepair. Because of this, iMonitor's features are primarily server focused, meaning that they focus on the health of individual eDirectory agents (running instances of the directory service) rather than the entire eDirectory tree.

iMonitor 2.1 provides the following features:

- ◆ eDirectory health summary
 - ◆ Synchronization information
 - ◆ Known servers
 - ◆ Agent configuration
- ◆ eDirectory health checks
- ◆ Hyperlinked DS Trace
- ◆ Agent configuration
- ◆ Agent activity and verb statistics
- ◆ Reports
- ◆ Agent information
- ◆ Error information
- ◆ Object/schema browser
- ◆ DirXML® monitor
- ◆ Search
- ◆ Partition list
- ◆ Agent process status
- ◆ Background process schedule
- ◆ DSRepair
- ◆ Connection monitor

The information you can view in iMonitor is based the following factors:

- ◆ The identity you have established
Your identity's eDirectory rights are applied to every request you make in iMonitor. For example, you must log in as the Administrator of the server or a console operator on the server where you are trying to access the DSRepair page.
- ◆ The eDirectory agent version you are monitoring
Newer versions of NDS[®] and eDirectory will have features and options that older versions do not.

The information you view in iMonitor immediately shows what is happening on your server.

This chapter gives information on the following topics:

- ◆ [“System Requirements” on page 164](#)
- ◆ [“Accessing iMonitor” on page 165](#)
- ◆ [“iMonitor Architecture” on page 165](#)
- ◆ [“iMonitor Features” on page 171](#)
- ◆ [“Ensuring Secure iMonitor Operations” on page 187](#)

System Requirements

To use iMonitor 2.1, you need

- ◆ Internet Explorer 5.5 or later or Netscape 7.02 or later
- ◆ Novell eDirectory 8.7.1 or later

Platforms

The iMonitor 2.1 utility runs on the following platforms:

- ◆ NetWare[®] 5.1 Support Pack 4 or later
Novell iMonitor is placed in autoexec.ncf.
- ◆ Windows NT, 2000, and 2003 Server (No SSL)
- ◆ Linux
- ◆ Solaris
- ◆ AIX
- ◆ HP-UX

For NetWare and Windows, iMonitor loads automatically when eDirectory runs. On Linux, Solaris, AIX, and HP-UX, iMonitor can be loaded using the `ndsmonitor -l` command. It can also be loaded automatically by adding `[ndsmonitor]` in the `/etc/ndsicon.conf` file before starting the eDirectory Server.

eDirectory Versions That Can Be Monitored

You can use iMonitor to monitor the following versions of NDS and eDirectory:

- ♦ All versions of NDS and eDirectory for NetWare 4.11 or later
- ♦ All versions of NDS and eDirectory for Windows
- ♦ All versions of NDS and eDirectory for Linux, Solaris, AIX, and HP-UX

Accessing iMonitor

1 Ensure that the iMonitor executable is running on the eDirectory server.

2 Open your Web browser.

3 In the address (URL) field, enter

`http://server's_TCPIP_address:httpstack_port/nds`

for example:

`http://192.10.10.1/xyz`

DNS names can be used anywhere a server's IP or IPX™ address or distinguished name could be used in iMonitor. For example, when you have configured DNS, then

`http://prv-gromit.provo.novell.com/nds?server=prv-igloo.provo.novell.com`

is equivalent to

`http://prv-gromit.provo.novell.com/nds?server=IP_or_IPX_address`

or

`http://prv-gromit.provo.novell.com/nds?server=/cn=prv-igloo,ou=ds,ou=dev,o=novell,t=novell_inc`

If an eDir HTTPS stack is available, you can use iMonitor through HTTPS.

4 Specify a user name, context, and password.

To have access to all of the features, log in as Administrator with the fully distinguished name, or as an administrator equivalent.

5 Click Login.

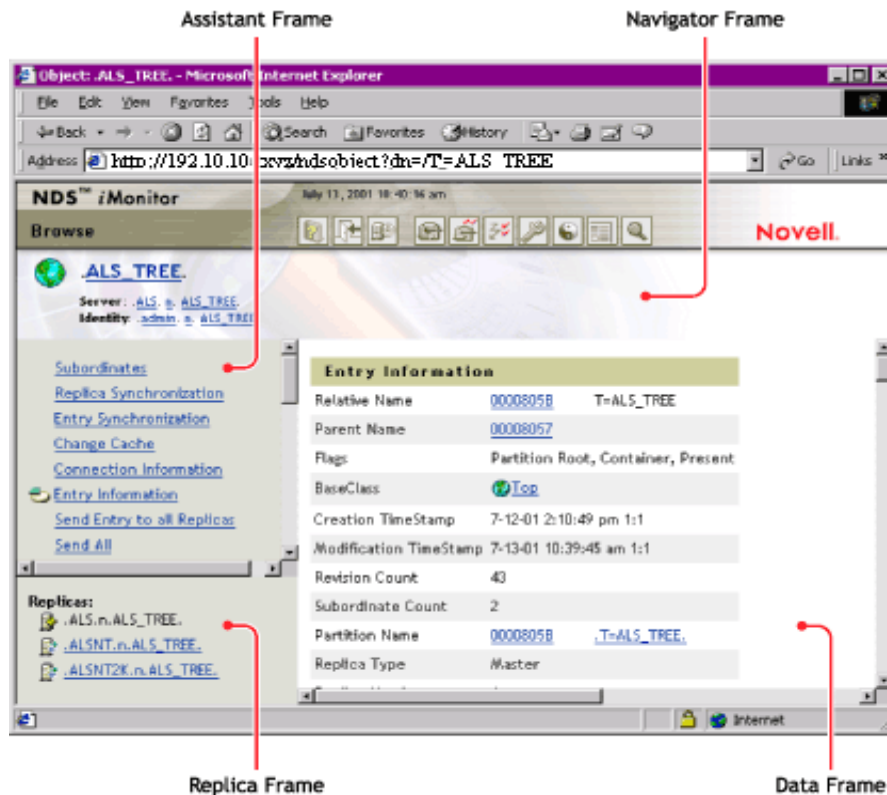
iMonitor Architecture

- ♦ [“Anatomy of an iMonitor Page” on page 166](#)
- ♦ [“Modes of Operation” on page 166](#)
- ♦ [“iMonitor Features Available on Every Page” on page 168](#)
- ♦ [“NetWare Remote Manager Integration” on page 168](#)
- ♦ [“Configuration Files” on page 168](#)

Anatomy of an iMonitor Page

Each iMonitor page is divided into four frames or sections: the Navigator frame, the Assistant frame, the Data frame, and the Replica frame.

Figure 26 iMonitor Frames



Navigator Frame: Located across the top of the page. This frame shows the server name where the data is being read from, your identity, and the icons you can click to link to other screens, including online help, login, server portal, and other iMonitor pages.

Assistant Frame: Located at the left side of the page. This frame contains additional navigational aids, such as links to other pages, items that help you navigate data in the Data frame, or other items to assist you with obtaining or interpreting the data on a given page.

Data Frame: Shows the detailed information about your servers that you request by clicking one of the links listed above. This is the only page you will see if your Web browser does not support frames.

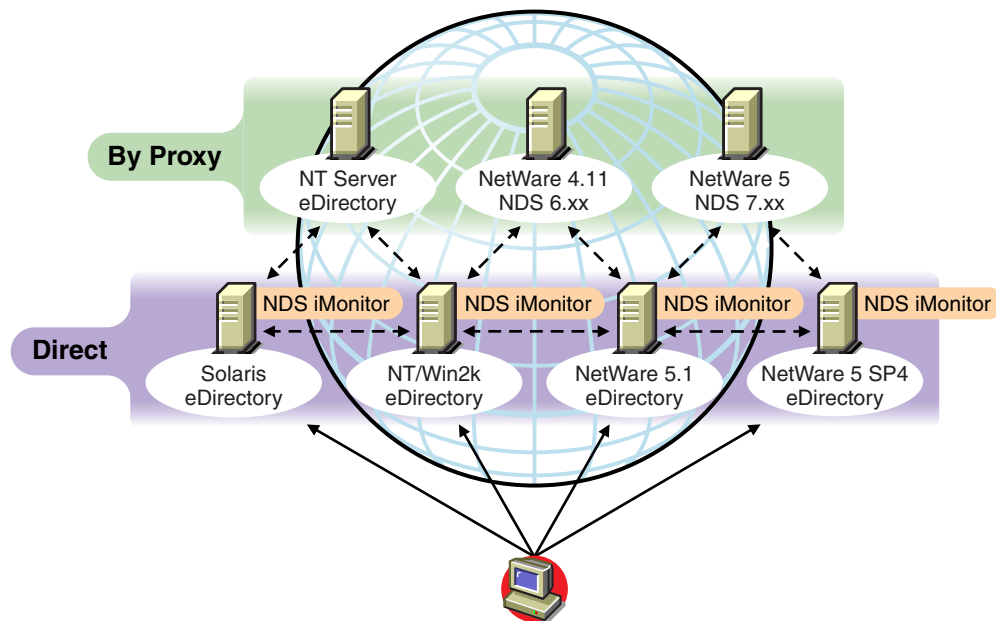
Replica Frame: Lets you determine which replica you are currently viewing and provides links to view the same information from another replica or server's point of view. This frame appears only when you view pages where another replica of the requested data exists or where another replica might have a different view of the information being presented in the Data frame.

Modes of Operation

Novell iMonitor can be used in two different modes of operation: Direct mode and Proxy mode. No configuration changes are necessary to move between these modes. Novell iMonitor

automatically moves between these modes, but you should understand them in order to successfully and easily navigate the eDirectory tree.

Figure 27 Modes of Operation



Direct Mode: Use this mode when your Web browser is pointed directly at an address or DNS name on a machine running the iMonitor executable and reading information only on that machine's local eDirectory DIB.

Some iMonitor features are server-centric; that is, they are available only to the iMonitor running on that machine. These features use local API sets that cannot be accessed remotely. Server-centric features in iMonitor include the DSTrace, DSRRepair, and Background Process Schedule pages. When using Direct mode, all iMonitor features will be available on that machine.

Key features of Direct mode:

- ◆ Full server-centric feature set
- ◆ Reduced network bandwidth (faster access)
- ◆ Access by proxy still available for all versions of eDirectory

Proxy Mode: Use this mode when your Web browser is pointed at an iMonitor running on one machine, but is gathering information from another machine. Because iMonitor uses traditional eDirectory non-server-centric protocols for non-server-centric features, all previous versions of eDirectory beginning with NDS 6.x can be monitored and diagnosed. However, server-centric features use APIs that cannot be accessed remotely.

If you are in Proxy mode and want to switch to Direct mode for a different server, you can do so as long as the server has a version of eDirectory in which iMonitor has shipped. If the server you are gathering information on by proxy has iMonitor running, you will see an additional icon button in the Navigator frame. When you move the mouse pointer over the icon, you will see a link to the remote iMonitor on the remote server. If the server you are gathering information on by proxy is an earlier version of eDirectory, no additional icon is shown and you will always need to gather information on that server by proxy until it is upgraded to a version of eDirectory that includes iMonitor.

Key features of Proxy mode:

- ◆ Not every server in the tree must be running iMonitor in order to use most iMonitor features
- ◆ Only one server must be upgraded
- ◆ There is a single point of access for dial-in
- ◆ You can access iMonitor over a slower speed link while iMonitor accesses eDirectory information over higher speed links
- ◆ Previous NDS version information is accessible
- ◆ Server-centric features are available only where iMonitor is installed

iMonitor Features Available on Every Page

You can link to the Agent Summary, Agent Information, Agent Configuration, Trace Configuration, DSRepair, Reports, and Search pages from any iMonitor page by using the icons in the Navigator frame. You can also log in or link to the Novell Support Web page from any iMonitor page.

Login/Logout: The Login button is available if you are not logged in. A Logout button, which closes your browser window, is displayed if you are logged in. Unless all browser windows are closed, your iMonitor session remains open, and you will not need to log in again. You can see your login status on any page by looking at Identity in the Navigator frame.

Support Connection Link: The Novell logo in the upper right corner is a link to the Novell Support Connection Web page. This provides a direct link to the Novell Web site for current server patch kits, updates, and product-specific support.

NetWare Remote Manager Integration

On NetWare 5 and later servers, a link to NetWare Remote Manager is available to provide you with Web-based monitoring, diagnosis, and troubleshooting information for NetWare servers.

iMonitor is integrated with NetWare Remote Manager in the following ways:

- ◆ NetWare Remote Manager's lightweight Web server (httpstk.nlm) provides the first layer of the iMonitor architecture on the NetWare platform.
- ◆ iMonitor registers with NetWare Remote Manager (portal.nlm) so that links to iMonitor and other eDirectory-specific information are available through the NetWare Remote Manager interface.

These links are found under the Manage eDirectory section in the Remote Manager interface. Links to eDirectory agent health information are also found in the Diagnose Server section under Health Monitor in the eDirectory-related categories.

NetWare Remote Manager also registers with eDirectory, which allows iMonitor and NetWare Remote Manager to cross-reference each other for a more seamless movement between each tool.

Configuration Files

Configuration files are included with iMonitor to allow you to change or set default behavior or values in the utility.

The configuration files are text files containing configuration parameter tags together with their desired values. These files are located in the same directory as the iMonitor executable (which is

usually in the same location as the Novell eDirectory executables) on NetWare and Windows, and in the /etc directory on Linux, Solaris, AIX, and HP-UX.

- ◆ “[ndsimon](#)” on page 169
- ◆ “[ndsimonhealth](#)” on page 169

ndsimon

The ndsimon configuration file lets you modify trace file settings, control access to the server, set the maximum number of object to be displayed when listing a container or displaying search results, and specify the number of minutes of inactivity allowed before a connection is logged out.

Server	Configuration File
NetWare	sys:\system\ndsimon.ini
Windows NT and Windows 2000	<i>install directory</i> \novell\NDS\ndsimon.ini
Linux, Solaris, AIX, and HP-UX	/etc/ndsimon.conf

There are two groups of parameters that you can set in the ndsimon configuration file.

- ◆ Parameters that apply to how the iMonitor executable itself runs

Except on NetWare, when the iMonitor executable loads, it will attempt to listen on the traditional HTTP port 80. If that port is in use, it will back off to port 8008. If that port is in use, iMonitor will then back off again, increasing the port by 2 (8010, 8012, etc.) up to 8078.

Where SSL is configured and available, a similar bind pattern is attempted. First, port 81 is tried, and then 8009, 8011, 8013, etc.

This allows iMonitor to coexist with a Web server running on the same server. However, on some platforms, iMonitor might load before the installed Web server does, or you might want iMonitor to bind to a port of your choice. Both regular and SSL ports can be configured using the HttpPort and the HttpsPort parameters respectively. Commented-out examples exist in the shipping configuration file. By default, iMonitor binds to all NIC addresses on the server where it loads. However, there is an Address parameter that you can use to specify a list of addresses, in comma delimited format, to bind to.

On NetWare, similar port selection rules are used, but they are controlled by the NetWare Remote Manager HTTP stack (httpstk.nlm) and work as specified in the NetWare Remote Manager documentation.

- ◆ Parameters that apply to specific features or pages

The configuration file that ships with iMonitor contains samples of the parameters that can be modified. These parameters are preceded by a pound sign (#). This indicates that they are commented out or not used when iMonitor parses the configuration file. For the shipping configuration file, iMonitor uses all internally bound default values for these parameters. To enable any of these parameters or to add any parameters, simply delete the # character from the beginning of the line.

ndsimonhealth

The ndsimonhealth configuration file lets you modify default settings for the Agent Health page. You can enable or disable Agent Health options, set reporting levels and ranges for options, and set server reporting levels.

Server	Configuration File
NetWare	sys:\system\ndsimonhealth.ini
Windows NT and Windows 2000	<i>install directory</i> \novell\NDS\ndsimonhealth.ini
Linux, Solaris, AIX, and HP-UX	/etc/ndsimonhealth.conf

There are three types of options you can set in the ndsimonhealth configuration file.

- ◆ Enable/disable only options

To disable an option, remove the pound sign (#) from in front of the option and replace any levels listed after the colon (:) with OFF. To set reporting levels of these options, remove the # character from in front of the option and add a reporting level after the colon. Valid levels are WARN, MARGINAL, and SUSPECT. For these options, you can input only one reporting level.

- ◆ General options that take a range of settings

These options can be enabled and disabled or have their reporting level set, as well as the ranges for those reporting levels.

To set the reporting level for any of these options, use the option name followed by -active: and the reporting levels you want. For example, to set time_delta active, add the following line to the configuration file:

```
time_delta-active: WARN
```

To set time_delta inactive, add the following line to the configuration file:

```
time_delta-active: OFF
```

When entering ranges, the specified range is the range that this reporting level should not be displayed for.

See the time_delta example below for an example of how to set an option to be active for all three reporting levels and how to set the ranges. In this example, anything not in the range -2 to 2 is at least marginal, anything not in the range -5 to 5 is at least suspect, and anything not in the range -10 to 10 is a warning.

```
time_delta-active: WARN | SUSPECT | MARGINAL
time_delta-Min_Warn:      -10
time_delta-Min_Suspect:   -5
time_delta-Min_Marginal:  -2
time_delta-Max_Marginal:   2
time_delta-Max_Suspect:   5
time_delta-Max_Warn:     10
```

For help on any of these options, enter the following URL in iMonitor:

```
http://XXX.XXX.XXX.XXX:PORT/nds/help?hbase=/nds/health/OPTION_NAME
```

XXX.XXX.XXX.XXX:PORT is the IP address and port where iMonitor can be reached, and *OPTION_NAME* is the name of the option you want help on (for example, time_delta).

To view the currently set levels and ranges, use your browser to go to the health page that contains the option you are interested in, then add the following to the end of the URL line in the browser:

```
&op=setup
```

- ◆ Options that need custom or complex settings

There are three different server reporting levels that can be set:

- ◆ WARN detects servers running a version of eDirectory that should be upgraded as soon as possible.
- ◆ SUSPECT detects servers running a version of eDirectory that should be noted for upgrade.
- ◆ MARGINAL detects servers running a version of eDirectory that is not current.

These options set the reporting level if the server version falls within the specified range.

iMonitor Features

This section provides brief descriptions of iMonitor features.

Online help is provided in each section of iMonitor for more detailed information about each feature and function.

- ◆ [“Viewing eDirectory Server Health” on page 172](#)
- ◆ [“Viewing Partition Synchronization Status” on page 172](#)
- ◆ [“Viewing Server Connection Information” on page 172](#)
- ◆ [“Viewing Known Servers” on page 173](#)
- ◆ [“Viewing Replica Information” on page 174](#)
- ◆ [“Controlling and Configuring the DS Agent” on page 174](#)
- ◆ [“Configuring Trace Settings” on page 175](#)
- ◆ [“Viewing Process Status Information” on page 176](#)
- ◆ [“Viewing Agent Activity” on page 176](#)
- ◆ [“Viewing Traffic Patterns” on page 177](#)
- ◆ [“Viewing Background Processes” on page 177](#)
- ◆ [“Viewing eDirectory Server Errors” on page 177](#)
- ◆ [“Viewing DSRepair Information” on page 177](#)
- ◆ [“Viewing Agent Health Information” on page 178](#)
- ◆ [“Browsing Objects in Your Tree” on page 178](#)
- ◆ [“Viewing Entries for Synchronization or Purging” on page 179](#)
- ◆ [“Viewing the Synchronization Status of a Replica” on page 179](#)
- ◆ [“Configuring and Viewing Reports” on page 179](#)
- ◆ [“Viewing Schema, Class, and Attribute Definitions” on page 181](#)
- ◆ [“Searching for Objects” on page 181](#)
- ◆ [“Using the Stream Viewer” on page 182](#)
- ◆ [“Clone DIB Set” on page 182](#)

Viewing eDirectory Server Health

From the Agent Summary page, you can view the health of your eDirectory servers, including synchronization information, agent process status, and the total servers known to your database.

1 In iMonitor, click Agent Summary .

2 Choose from the following options:

Agent Synchronization Summary lets you view the number and types of replicas you have and the length of time since they have been successfully synchronized. You can also view the number of errors for each replica type. If there is only one replica or partition to view, the heading is Partition Synchronization Status.

If the Agent Synchronization Summary doesn't appear, there are no replicas you can view based on your identity.

Servers Known to Database Totals lets you view the type and count of servers known to your database, and whether they are up or down.

Agent Process Status Totals let you view the status of processes without the administrator's intervention that run on an agent. When there is a problem or piece of information, a status is recorded. The table increases or decreases, depending on the number of recorded statuses.

Viewing Partition Synchronization Status

From the Agent Synchronization page you can view the synchronization status of your partitions. You can filter the information by selecting from the options listed in the Assistant frame on the left side of the page.

1 In iMonitor, click Agent Synchronization in the Assistant frame.

2 Choose from the following options:

Partition Synchronization Status lets you view the partition, number of errors, last successful synchronization, and maximum ring delta.

Partition lets you view the links to each partition's Replica Synchronization page.

Last Successful Sync lets you view the amount of time since all replicas of an individual partition were successfully able to synchronize from the server.

Maximum Ring Delta shows the amount of data that might not be successfully synchronized to all the replicas in the ring. For example, if a user has changed his login script within the past 30 minutes, and the maximum ring delta has a 45-minute allocation, the user's login might not be successfully synchronized, and he might get the previous login script when he attempts to log in. If, however, the user changed his login script more than 45 minutes ago, he should get the new login script consistently from all replicas.

If Unknown is listed under Maximum Ring Delta, it means the transitive synchronized vector is inconsistent and the maximum ring delta cannot be calculated due to replica/partition operations occurring, or some other problem.

Viewing Server Connection Information

From the Agent Information page you can view the connection information for your server.

1 In iMonitor, click Agent Information in the Assistant frame.

2 Choose from the following options:

Ping Info shows that iMonitor has attempted an IP ping to the set of addresses being advertised for the server. Success is as indicated.

DNS Name shows that iMonitor has attempted to do an address reversal on IP addresses supported by the server and is indicating the associated DNS name.

Depending on the transport, configuration, and platform you are running on, you might not see this information.

Connection Information lets you view connection information for the server, including the server referral, time delta, Root Most Master, and replica depth.

Depending on the transport, configuration, and platform you are running on, you might not see this information.

Server Referral lets you view the set of addresses by which your server can be reached.

Time Synchronized indicates that synthetic or future time is not being used unless a replica's last-issued time stamp is greater than the current time.

eDirectory believes time is synchronized well enough to issue time stamps based on the server's current time. The time synchronization protocol might or might not currently be in a synchronized state.

Time Delta lets you view the difference in time between iMonitor and the remote server in seconds. A negative integer indicates that iMonitor's time is ahead of the server's time; a positive integer indicates that iMonitor's time is slower than the server.

Root Most Master specifies that the replica that is highest or closest to the root of the naming tree is a master replica.

Replica Depth lets you view the depth of the rootmost replica (the number of levels between the rootmost replica and the root of the tree).

Viewing Known Servers

From the Known Servers List, you can view the list of servers known to the database of the source server. You can filter the list to show all servers known to the database or to show all servers in the replica ring. If a server has an icon next to it, the server participates in a replica ring.

1 In iMonitor, click Known Servers in the Assistant frame.

2 Choose from the following options:

Entry ID lists the identifier on the local server for an object. Entry IDs cannot be used across servers.

NDS Revision lists the eDirectory build number or version being cached or stored on the server that you are communicating with.

Status shows whether the server is up, down, or unknown. If the status shows as unknown, this means that this server has never needed to communicate with the server being shown as unknown.

Last Updated shows the last time this server attempted to communicate with the server and found out it was down. If this column is not showing, all servers are currently up.

Viewing Replica Information

From the Partitions page, you can view information about the replicas on the server you are communicating with. You can filter the page by selecting from the options in the Assistant frame on the left side of the page.

Server Partition Information let you view information about the server's partition, including the entry ID, replica state, purge time, and last modification time.

Partition let you view information about the partition Tree object on the server.

Purge Time indicates the time when you can remove previously deleted data from the database because all replicas have seen the deletion.

Last Modification Time lets you view the last-issued time stamp of data written to the database for the replica. This lets you see if time is in the future and if synthetic time is being used.

Replica Synchronization lets you view the Replica Synchronization Summary page that refers to the partition. The Replica Synchronization page shows information about the partition synchronization status and replica status. You can also view lists of partitions and replicas.

Controlling and Configuring the DS Agent

From the Agent Configuration page, you can control and configure the DS Agent. The functionality you have on this page will depend on the rights of the current identity and the version of eDirectory you are looking at.

1 In iMonitor, click Agent Configuration .

2 Choose from the following options:

Agent Information let you view the connection information for your server.

Partitions lets you view the replicas on the server you are communicating with.

Replication Filters lets you view the replication filters configured for the specified eDirectory agent. NDS eDirectory 8.5 (build version 85.xx) is the first eDirectory version to implement a feature known as Filtered Replicas. See "[Filtered Replicas](#)" on page 53 for more information on what Filtered Replicas are, why they are used, and how to configure them.

Agent Triggers initiate certain background processes. These triggers are equivalent to using the SET DSTRACE=*option* command.

Background Process Settings modify the interval at which certain background processes run. These settings are equivalent to the SET DSTRACE=*option* command.

Agent Synchronization lets you disable or enable inbound or outbound synchronization. You can specify in hours the amount of time you want synchronization disabled.

Database Cache lets you configure the amount of database cache used by the DS database engine. Various cache statistics are also provided to assist you in determining whether you have an appropriate amount of cache available. Having an inadequate amount of cache might severely impact your system's performance.

Login Settings lets you disable the queuing of login updates. You can also increase or decrease the amount of time between updates if updates are enabled.

The latest versions of eDirectory implement a performance enhancement for login speed. This enhancement queues up changes that, in previous versions of NDS, were required to be done at login time while the user waited. Any change to the eDirectory database requires a lock, so

during peak usage periods, login times could be lengthier and more unpredictable depending on how many requests needed the database lock at any given time. By removing this lock requirement and queuing login updates, login speed is much faster and more predictable.

This option lets different eDirectory environments control this queuing behavior. In some environments, the data queued is extremely important and must be written to the database immediately. The user must then wait while the updates take place. In other environments, this data is not used at all and can be omitted. The default behavior should be adequate for most environments.

Configuring Trace Settings

From the Trace Configuration page, you can set trace settings. Novell iMonitor's DSTrace is a server-centric feature. That is, it can be initiated only on a server where iMonitor is running. If you need to access this feature on another server, you must switch to the iMonitor running on that server.

To access information on the Trace Configuration page, you must be the equivalent of Administrator of the server or a console operator. You are prompted to enter your username and password so your credentials can be verified before you can access information on this page.

1 In iMonitor, click Trace Configuration .

2 Choose from the following options:

Update lets you submit changes to Trace Options and Trace Line Prefixes. If DSTrace is off, click Trace On to turn it on. If DSTrace is already on, click Update to submit changes to the current trace.

Trace On/Off turns DSTrace on or off. The button text changes based on the current DSTrace state. If DSTrace is on, the button text will read Trace Off. Clicking it toggles DSTrace between off and on. When DSTrace is off, clicking Trace On is equivalent to clicking Update.

Trace Line Prefixes lets you choose which pieces of data are added to the beginning of any trace line.


DS Trace Options apply to the events on the local DS Agent where the trace is initiated. The options show errors, potential problems, and other information about eDirectory on your local server. Turning on DS Trace options can increase CPU utilization and might reduce your system's performance; therefore, DS Trace should generally be used for diagnostic purposes, not as a standard practice. These options are a more convenient equivalent of the SET DSTRACE=+*option* command.

Event Configuration lists the eDirectory event options you can enable or disable for monitoring in DSTrace. The event system generates events for local activities such as adding objects, deleting objects, and modifying attribute values. For each type of event, a structure is returned that contains information specific to that type of event.

Trace History lets you view a list of previous trace runs. Each previous trace log is identified by the period of time during which the trace data was being gathered.

Trace Triggers let you view the trace flags that must be set in order to display the specified DS Agent information in DSTrace. These triggers might write large quantities of information to trace. Generally, we recommend that these triggers be enabled only when instructed by Novell Support.

3 Click Trace On to turn DS Trace on and submit any changes.

4 Click  or Trace Live to view DS Trace in iMonitor.

Viewing Process Status Information

From the Agent Process Status page, you can view background process status errors and more information about each error that occurred. You can filter the information on this page by selecting from the options listed in the Assistant frame on the left side of the page.

- 1 In iMonitor, click Agent Process Status in the Assistant frame.

Background process statuses that are currently reported include

- ◆ Schema synchronization
- ◆ Obituary processing
- ◆ External reference/DRL
- ◆ Limber
- ◆ Repair

Viewing Agent Activity

From the Agent Activity page, you can determine traffic patterns and potential system bottlenecks. You can use this page to view the verbs and requests that are currently being handled by eDirectory. You can also see which of those requests are attempting to obtain DIB locks in order to write to the database and how many of those requests are waiting to obtain a DIB lock.

If you are viewing a server running Novell eDirectory 8.6 or later, you will also see a list of partitions and the servers that participate in the replica ring with the server specified in the Navigator frame. With the introduction of Novell eDirectory 8.6, synchronization is no longer single threaded. Any 8.6 server might outbound multiple partitions simultaneously to one or more replication partners. For this reason, the synchronization activity page was created so you can more easily monitor this parallel synchronization strategy.

- 1 In iMonitor, click Agent Activity in the Assistant frame.

- 2 Choose from the following options:

Verb Activity and Statistics lets you view a running count of all verbs called and requests made since eDirectory was last initialized. These pages also shows how many of those requests are currently active and the minimum, maximum, and average times (shown in milliseconds) that it takes to process those requests.

Synchronization Current and Schedule lists different times that inbound and outbound synchronization occurred. If inbound or outbound synchronization is currently taking place, you see an icon indicating that the process is active, when that cycle was started, and which server it is occurring with.

If inbound and outbound synchronization is disabled, you see an icon indicating that fact and when it is scheduled to be re-enabled. For outbound synchronization, the next scheduled time is also shown.

Events lets you view a list of the currently active events, statistics for event handlers and a summary of event statistics, and the current event rights functions that have been called.

Background Process Schedule lets you view the background processes that are scheduled, what their current state is, and when they are scheduled to run again.

Viewing Traffic Patterns

From the Verb Statistics page, you can determine traffic patterns and potential system bottlenecks. You can use this page to view a running count of all verbs called and requests made since eDirectory was last initialized. This page also shows how many of those requests are currently active and the minimum, maximum, and average times (in milliseconds) it takes to process those requests. Background process, bindery, and standard eDirectory requests are tracked.

If you view this page on an older version of eDirectory, you might not see as much information as if you are running eDirectory 8.5 or later.

Viewing Background Processes

From the Background Process Schedule page, you can view the background processes that are scheduled, what their current state is, and when they are scheduled to run again. Novell iMonitor's Background Process Schedule is a server-centric feature. That is, it can only be viewed on a server where iMonitor is running. If you need to access the background process schedule on another server, you must switch to the iMonitor running on that server. As you upgrade more servers to eDirectory 8.5, iMonitor's server-centric features will be more available to you. Other server-centric features include the DSTrace and DSRepair pages.

To access information on the Background Process Schedule page, you must be the equivalent of Administrator of the server or a console operator. You are prompted to log in so your credentials can be verified before you can access information on this page.

Viewing eDirectory Server Errors

From the Error Index page, you can view information about the errors found on your eDirectory servers. The errors are separated into two fields: eDirectory-specific errors and other errors that might be of interest. Each error listed is hyperlinked to a description that contains an explanation, possible cause, and troubleshooting actions.


- 1 In iMonitor, click Error Index in the Assistant frame.

From the Error Index page you can link to the latest Novell documentation on errors, technical information, and white papers.

Viewing DSRepair Information

From the DSRepair page, you can view problems and back up or clean up your DIB sets. Novell iMonitor's DSRepair is a server-centric feature. That is, it can be initiated only on a server where iMonitor is running. If you need to access the DSRepair information on another server, you must switch to the iMonitor running on that server. As you upgrade more servers to later versions of eDirectory, iMonitor's server-centric features will be more available to you. Other server-centric features include the DSTrace and Background Process Schedule pages.

To access information on this page, you must be the equivalent of Administrator of the server or a console operator. You are prompted to log in so your credentials can be verified before you can access information on this page.

- 1 In iMonitor, click DSRepair .

- 2 Choose from the following options:

Downloads lets you retrieve repair-related files from the file server. You will not be able to access dsrepair.log if the DSRepair utility is running or you have initiated a repair from the DSRepair page in iMonitor until the operation is finished.

Delete Old DIB Sets lets you delete an old DIB set by clicking the red X.

WARNING: This action is irreversible. When you select this option, the old DIB set will be purged from the file system.

DS Repair Advanced Switches lets you fix problems, check for problems, or create a backup of your database. You will not need to enter information in the Support Options field unless you are directed to do so by Novell Support.

- 3 Click Start Repair to run DS Repair on this server.

Viewing Agent Health Information

From the Agent Health page, you can view health information about the specified eDirectory agent and the partitions and replica rings it participates in.

- 1 In iMonitor, click Agent Health in the Assistant frame.
- 2 Click the links to view detailed information.

Browsing Objects in Your Tree

From the Browse page, you can browse any object in your tree. The Navigation bar at the top of the page lets you know what server the object you are viewing is on, and the path to the object. The Replica frame on the left of the page lets you view or access the same object on any real partition. Click any underlined object on the page to view more information about an object. You can also click any portion of the name in the Navigator frame to browse up the tree.

The information displayed on this page depends on the eDirectory rights you are logged in with, the type of object you are browsing, and the version of NDS or eDirectory you are running. This page displays XRef objects if you are logged in with Supervisor rights. You can use the replica list to jump to a real copy of the replica. If you are browsing for objects in dynamic groups, the time stamp will not be displayed for the dynamic members.

Replica Synchronization displays the synchronization status of the replica that contains this object.

Entry Synchronization shows which attributes need to be synchronized from this server's point of view.

Connection Information indicates where iMonitor got the information for this object.

Entry Information displays the names, flags, base class, modification time stamp, and summary of connection information for the object.

Send Entry to All Replicas resends this entry's attributes to all other replicas. This process could take some time if the object has many attribute values. This does not make all other copies of the object identical. It simply allows the other replicas to reconsider each attribute.

Send All (visible only if the object being browsed is a partition root and the Advanced Mode Option is enabled) resends all entries in this partition to all the servers holding replicas of the partition. This does not make all copies of the objects being sent identical. It simply allows the other replicas to reconsider each object and its attributes.

Viewing Entries for Synchronization or Purging

From the Change Cache page, you can view a list of entries that this server needs to consider for synchronization or purging. This option is available only if the server you are accessing is running eDirectory 8.6 or later and the object you are viewing is a partition root. You must have Supervisor rights to the NCP™ server to view this page.

Entry Synchronization lets you determine why an entry needs to be synchronized.

Viewing DirXML Details

From the DirXML Summary page, you can view a list of any DirXML drivers running on your server, the status of each driver, any pending associations, and driver details.

1 In iMonitor, click DirXML Summary .

2 Choose from the following options:

Status displays the current state of the specified driver. Possible states include stopped, starting, running, shut down, pending, and getting schema.

Start Option displays the current startup option specified for the selected driver.

Pending displays the number of associations that have not yet been made.

Driver Details Icon displays subscriber and publisher details, XML rules, filters, and pending association lists for DirXML drivers running on your server. Details on the first 50 pending objects are also displayed on this page. The XML rule details provided on this page can be used to determine what to look for in the pending objects to allow their creation to proceed for the specified DirXML driver.

Viewing the Synchronization Status of a Replica

From the Replica Synchronization page, you can view the synchronization status of a replica.

1 In iMonitor, click Agent Synchronization in the Assistant frame.

2 Click Replica Synchronization for the partition you want to view.

3 Use the links on this page and in the navigation bar on the left to access other partitions and jump through your replica ring.

Configuring and Viewing Reports

From the Reports page, you can view and delete reports run directly on this server. Some reports might take a long time to run and can be resource intensive.

Scheduled reports run without authenticating as a user (that is, as [Public]). Any reports you run directly are run as your identity. All report data is stored on the server from that report was run from.

The Report Config page lets you view a list of preconfigured, custom, and scheduled reports. Use this page to modify and run reports and to create custom reports for iMonitor pages. The following table lists preconfigured reports included with iMonitor 2.1.

Report	Description
Server Information	Walks the entire tree, communicates with every NCP server it can find, and reports any errors it finds. Use this report to diagnose time synchronization and limber problems, or to find out if the current server is able to communicate with all other servers from this server's perspective. If selected in the Configuration page, this server can also generate NDS Agent Health information for every server in the tree.
Obituary Listing	Lists all obituaries on this server.
Object Statistics	Evaluates the objects in a given scope, then generates lists of objects matching the requested criteria. These criteria include such things as future time, unknown objects, renamed objects, counts of base classes, containers, alias, and external references.
Service Advertising	Lists all directories and servers known to the current server through SLP or SAP.
Agent Health	Gathers health information for the current server.



Viewing and Deleting Reports

- 1 In iMonitor, click Reports .
- 2 Click  to delete a report or  to view a report.

Running a Report



- 1 In iMonitor, click Reports  > Report Config.
- 2 Click  to run a report.

Configuring or Scheduling a Report

- 1 In iMonitor, click Reports  > Report Config.
- 2 Click  to configure and schedule a report.
- 3 Select any options you want, then click Save Defaults to save the options you selected.
- 4 (Optional) Configure the report to run on either a periodic basis or at a later time.
 - 4a Specify a frequency, start time, and start day.
 - 4b Click Schedule.
- 5 Click Run Report to start the report.

Creating a Custom Report

Custom reports let you launch any iMonitor page as a report.

- 1 In iMonitor, click Reports  > Report Config.
- 2 Click  on the Custom Reports line in the Runnable Report list.
- 3 Enter a name for the report, then enter the URL to the iMonitor page you want to launch as a report.

When running a custom report, enter the URL as follows:

/nds/required page

- 4** Specify the number of versions of the report you want to keep.
- 5** (Optional) Click Save to save the report.
- 6** (Optional) Configure the report to run on either a periodic basis or at a later time.
 - 6a** Specify a frequency, start time, and start day.
 - 6b** Click Schedule.
- 7** Click Run Report to start the report.

Viewing Schema, Class, and Attribute Definitions

From the Schema page, you can view your schema, class, and attribute definitions. You can view the schema that is loaded on your tree, with any extensions that have been made, and information specific to your particular schema, such as any changes or extensions you've made to the schema.

- 1** In iMonitor, click Schema in the Assistant frame.
- 2** Choose from the following options:

Synchronization List lists the servers that this server will synchronize with. This option is available only for servers running NDS eDirectory 8.5 or later. You must have Supervisor rights on the server to view this information.

Schema Root displays information about the schema replica closest to the root of the tree in this context.

Each eDirectory server stores a replica of the schema in its entirety. The schema replica is stored separately from the partitions that contain directory objects. Changes to any one schema replica are propagated to the other replicas. You can perform modifications to the schema only through a server that stores a writable replica of the root partition. Servers storing read-only replicas of the root partition can read but not modify schema information.


Attribute Definitions lists the name of each attribute, the syntax that the attribute value will be in, and the constraints that the attribute operates under. Use the navigation frame on the left to browse for and access individual attributes.

Class Definitions lists the name of each class, its rules, and its attributes. Use the navigation frame on the left to browse for and access individual attributes.

Searching for Objects

From the Search page, you can search objects based on a variety of query options and filters. The search query options and filters are grouped in two levels of search request forms: basic and advanced. The basic search request form is designed for average users of eDirectory and simple searches. The advanced search request form is designed for advanced users and complicated searches. Currently, only server-level search is supported.

All the search options and filters in the four sections are conjunctive. Blank fields (except the Relative Distinguished Name) will be ignored. Use the Ctrl key to deselect an item or select more than one item on the multilists. Deselected multilists will also be ignored.

- 1** In Novell iMonitor, click Search .
- 2** Choose from the following options:
 - Scope Options** lets you specify the scope of the search.

Entry Filters lets you specify search query filters related to the entry information.

Attribute and Value Filters lets you specify search query filters related to the attributes and values.

Display Options lets you specify options which control the display format of the search results.

- 3 Click the Help button at the bottom of the search request form to see brief help information added to the form itself.

Click Reload or Refresh to clear the help information.

Using the Stream Viewer

From the Stream Viewer page, you can view the current stream in any of the following formats:

- ◆ Plain text
- ◆ HTML
- ◆ GIF
- ◆ JPEG
- ◆ BMP
- ◆ WAV
- ◆ Hex Dump
- ◆ Other

If you have stream attributes that you consistently want to view in a particular format, you can use the Stream Viewer to select default display settings.

NDS Stream Attribute Setup changes the default display format for streams in your browser. It is up to your browser to display the stream correctly, so it might not always apply the settings you have selected.

You must be authenticated to the server to apply any changes you have made to the default settings. Your changes are stored in streams.ini (for NetWare and Windows servers) or streams.conf (for Solaris and Linux servers), so you can also manually edit the default settings.

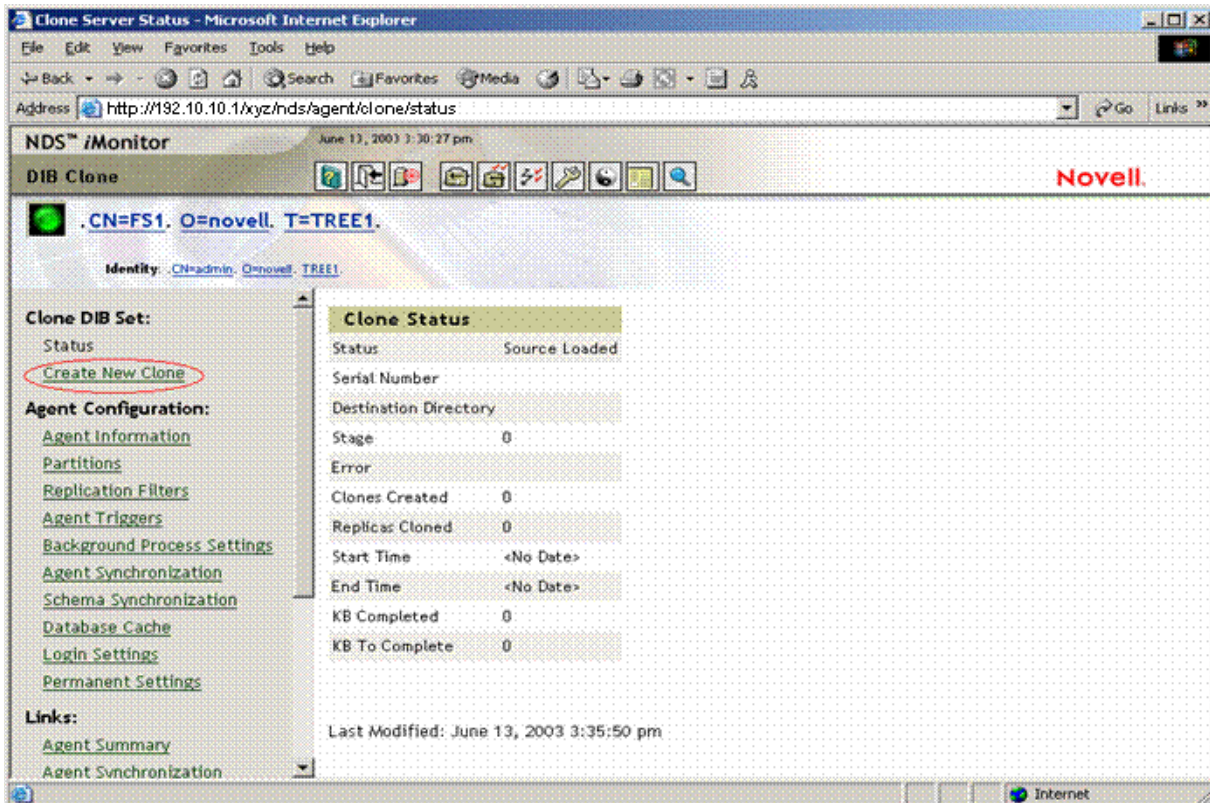
Clone DIB Set

This option creates a complete DIB fileset duplicate of an eDirectory database stored on a single server (the source server). The clone can then be placed on another server (the target server). When the target server initiates eDirectory, it loads the DIB fileset, contacts the master replica of the server object, resolves its name, then synchronizes any changes to the DIB fileset made after the clone was created.

The clone of an eDirectory DIB set should only be placed on a server running the same operating system as the server the clone was created on. For example, if you want to restore a cloned DIB fileset to a Solaris server, create the clone on a Solaris server and not on a NetWare or Windows server.

Although the back end for this feature shipped with eDirectory 8.7, it was not supported until eDirectory 8.7.1 running iMonitor 2.1 or later. This option does not apply to any version of Novell eDirectory or NDS prior to 8.7.

Figure 28 Clone DIB Set Page in iMonitor



This section includes the following information:

- ♦ “Clone DIB Set Use Cases” on page 183
- ♦ “Creating a Clone” on page 184

Clone DIB Set Use Cases

Clone DIB Set provides the following use cases:

- ♦ Create a new server with partitions already in an “on” state.

Advantages include the following:

- ♦ All servers in the ring do not need to be up and running to add a new server to the replica ring.
- ♦ A new server will automatically have all partitions with no synchronization necessary.
- ♦ Quicker up time.
- ♦ Disaster recovery

Advantages	Disadvantages
<ul style="list-style-type: none"> ♦ Only need one copy of the partition to succeed. ♦ Less down time on large servers with multiple partitions. 	<ul style="list-style-type: none"> ♦ Must have at least one good copy of the partitions in question. ♦ Won't handle any SSL or security backups. ♦ Does not handle the file system.

- ◆ Backup and restore

Advantages	Disadvantages
<ul style="list-style-type: none"> ◆ Quicker up time, especially on large scale databases. 	<ul style="list-style-type: none"> ◆ Only adds core eDirectory. LDAP, SNMP, SSL, etc. are not installed or configured. ◆ Will not get the latest changes. Only a snapshot is taken. Roll forward logs are not executed.

Because of the listed disadvantages, we do not recommend using Clone DIB Set for backup and restore purposes.

Creating a Clone

A clone DIB fileset can be created with the originating server either online or offline. The offline method requires eDirectory to be brought down. In the online mode, eDirectory is not locked.

- ◆ [“Online Method” on page 184](#)
- ◆ [“Offline Method” on page 185](#)

Online Method

- 1 Extend the schema of the tree.

Make sure to extend the schema, or an error will occur. Use `dibclone.sch`, which is present in the eDirectory installation. This will add the needed attributes for the iMonitor clone utility to operate.

Platform	To Extend the Schema
NetWare	Use NWConfig (NWConfig.nlm > Configuration Options > Directory Options > Extend Schema). <code>dibclone.sch</code> is located in <code>sys:\system\schema</code> .
Windows	Use NDSCons.exe (in NDSCons.exe, load <code>install.dlm</code> , then click Install Additional Schema Files). <code>dibclone.sch</code> is located in <code>C:\Novell\NDS</code> .
Linux, Solaris, AIX, and HP-UX	Use <code>ndssch</code> . <code>dibclone.sch</code> is located in <code>/usr/lib/nds-schema</code> . For more information, see “Using the ndssch Utility to Extend the Schema on Linux, Solaris, AIX, or HP-UX” on page 108 .

- 2 Create the clone DIB fileset.

- 2a Load the `dsclone` module on the source server.

Platform	Module
NetWare	At the server console, enter <code>dsclone.nlm</code> .

Platform	Module
Windows	In NDSCons.exe, select dsclone.dll, then click Start.
Linux, Solaris, AIX, and HP-UX	Add an "ndsclone" entry to the ndsmodules.conf file, then use the <code>http://IP address:port/dhost</code> page to load the Directory Clone Agent.

2b Run Clone DIB Configuration in iMonitor.

Click Agent Configuration > Clone DIB Set > Create New Clone.

2c Specify the fully qualified name of the target server and the file path where the cloned DIB files will be placed, then check the Create Clone Object and the Clone DIB Online boxes.

The NCP Server name (Clone Object) of the target server must match the target server name.

2d Click Submit.

The NDS Clone object is created and the DIB fileset is copied to the specified destination.

3 Move the cloned DIB fileset onto the target server in the proper directory location.

Additionally, on Linux, Solaris, AIX, and HP-UX systems, transfer the `/etc/nds.conf` file to the target server and update all the references to the source server in the file with the target server name.

4 Run eDirectory on the source server.

Make sure the master replica of the target Server object is running eDirectory and is available. When eDirectory initializes on the target server, it communicates with the master replica where the final naming of the target server is resolved.

Offline Method

1 Extend the schema of the tree.

Make sure to extend the schema, or an error will occur. Use `dibclone.sch`, which is present in the eDirectory installation. This will add the needed attributes for the iMonitor clone utility to operate.

Platform	To Extend the Schema
NetWare	Use NWConfig (NWConfig.nlm > Configuration Options > Directory Options > Extend Schema). <code>dibclone.sch</code> is located in <code>sys:\system\schema</code> ,
Windows	Use NDSCons.exe (in NDSCons.exe, load <code>install.dlm</code> , then click Install Additional Schema Files). <code>dibclone.sch</code> is located in <code>C:\Novell\NDS</code> ,

Platform	To Extend the Schema
Linux, Solaris, AIX, and HP-UX	<p>Use ndssch.</p> <p>dibclone.sch is located in /usr/lib/nds-schema.</p> <p>For more information, see “Using the ndssch Utility to Extend the Schema on Linux, Solaris, AIX, or HP-UX” on page 108.</p>

2 Create the clone DIB fileset.

2a Run Clone DIB Configuration in iMonitor.

Click Agent Configuration > Clone DIB Set > Create New Clone.

2b Specify the fully qualified name of the target server, check the Create Clone Object box, then uncheck the Clone DIB Online box.

The NCP Server name of the target server must match the target server name.

2c Click Submit.

The NDS clone object is created, eDirectory is brought down on the source server, and an error reports that eDirectory is locked.

2d Manually copy the *.nds, nds*, and nds.rfl/*.* files to a destination or media on the target server convenient for moving the set to the target server. Additionally, on Linux, Solaris, AIX, and HP-UX systems, transfer the /etc/nds.conf file to the target server and update all the references to the source server in the file with the target server name.

2e Bring up eDirectory on the source server.

If eDirectory is restarted on the source server before the files are copied, this cone is invalid. The new NCP Server object must then be deleted and the clone must be recreated.

3 Move the cloned DIB fileset onto the target server into the proper directory location.

4 Run eDirectory on the target server.

Make sure the master replica of the new target Server object is running eDirectory and is available. When eDirectory initializes on the target server, it communicates with the master replica where the final naming of the target server is resolved.

Completing the eDirectory Configuration

SDIKEY

1 Bring down eDirectory on the target server.

2 Copy the NICISDI.KEY file from the appropriate directory on the source server to the target server.

Platform	Directory
NetWare	sys:\system\nici\NICISDI.KEY
Windows	C:\WINNT\System32\Novell\NIC\NICISDI.KEY
Linux, Solaris, AIX, and HP-UX	/var/novell/nici/0/nicisdi.key

- 3 Start eDirectory on the target server.

Configuring SAS, LDAP, and SNMP Services

On Linux, Solaris, AIX, and HP-UX, the services listed below can be configured in one operation by entering the following command at the command line:

```
ndsconfig upgrade [-a admin FDN]
```

SAS

Platform	Command or Tool
NetWare	Create SAS Service object and Certificates using iManager.
Windows	Create SAS Service object and Certificates using iManager.
Linux, Solaris, AIX, and HP-UX	<code>ndsconfig -t <i>tree_name</i> -o <i>server_context</i> -m sas</code>

LDAP

Platform	Command or Tool
NetWare	Create LDAP Server and Group Objects using iManager.
Windows	Create LDAP Server and Group Objects using iManager.
Linux, Solaris, AIX, and HP-UX	<code>ndsconfig -t <i>tree_name</i> -o <i>server_context</i> -m ldap</code> or Create LDAP Server and Group Objects using iManager.

SNMP

Platform	Command or Tool
NetWare	<code>SNMPINST -c <i>adminContext password ServerDN</i></code>
Windows	<code>rundll32 snmpinst, snmpinst -c <i>createobj -a userFDN -p password -h hostname_or_IP_address</i></code>
Linux, Solaris, AIX, and HP-UX	<code>ndsconfig -t <i>tree_name</i> -o <i>server_context</i> -m snmp</code>

Ensuring Secure iMonitor Operations

Securing access to your iMonitor environment involves the following protective steps:

1. Use a firewall and provide VPN access (this also applies to Novell iManager and any other Web-based service that should have restricted access).
2. Whether a firewall is in place or not, limit the type of access allowed through iMonitor to further protect against Denial of Service (DoS) attacks.

Although substantial efforts have been made to ensure that iMonitor validates the data it receives via URL requests, it is nearly impossible to guarantee that every conceivable invalid

input is rejected. To reduce the risk of DoS attacks via invalid URLs, there are three levels of access that can be controlled through **iMonitor's configuration file** using the LockMask: option.

Access Level	Description
0	Require no authentication before iMonitor processes URLs. In this case, the eDirectory rights of the .[Public]. identity are applied to any request, and information displayed by iMonitor is restricted to the rights of the .[Public]. user. However, because no authentication is required to send URLs to iMonitor, iMonitor might be vulnerable to DoS attacks that are based on sending garbage in the URL.
1 (Default)	Before iMonitor processes URLs, require successful authentication as some eDirectory identity. In this case, the eDirectory rights of that identity are applied to any request and are, therefore, restricted by those rights. The same DoS vulnerability as level 0 exists, except the attack must be launched by someone who has actually authenticated to the server. Until a successful authentication occurs, the response to any iMonitor URL request is a login dialog box, so iMonitor should be impervious to attacks by unauthenticated users when it is configured in this state.
2	Before iMonitor processes URLs, require successful authentication as an eDirectory identity that has supervisor equivalency on the server that iMonitor is authenticating to. The same DoS vulnerability as level 1 exists, except the attack must now be launched by someone who has actually authenticated as a supervisor of the server. Until a successful authentication occurs, the response to any iMonitor URL request is a login dialog box, so iMonitor should be impervious to attacks by unauthenticated users and non-supervisor authenticated users when it is configured in this state.

Level 1 is the default because many administrators do not have supervisory access to every server in the tree but might need to use the iMonitor service on a server that their servers interact with.

NOTE: There are several features of iMonitor, such as Repair and Trace, that require supervisor equivalency to access regardless of the LockMask setting.

8

Merging Novell eDirectory Trees

The Novell® eDirectory™ Merge utility allows you to merge two separate Novell eDirectory trees into a single eDirectory tree. Only the Tree objects are merged; container objects and their leaf objects maintain separate identities within the newly merged tree.

TIP: To move leaf objects or merge partitions, use ConsoleOne® or Novell iManager.

The two trees you merge are called the local source tree and the target tree. Before merging one tree into another tree, the target tree should have all but one replica of the root partition removed. When there is only one replica of the root partition in the target tree, you can proceed with the merge. After the merge, there will be two replicas of the root partition—the replica that was on the target tree and the replica that was on the source tree server that ran the merge operation. If you need additional replicas of the root partition in your tree, you can place them after the merge has completed.

If the target tree server contains more than one replica of the root partition when the merge takes place, servers not holding the master replica might have a problem with the placement of external reference objects. These objects are contained in subordinate reference partition roots that must be placed on the other servers that have a replica of the root partition to represent partition boundaries. For each partition subordinate to the root partition in the source tree, there must be a subordinate reference partition root placed in the target tree. If there is a failure, it will report an eDirectory error code of -605 for synchronization status. In this case, use DSRepair to run a local database repair on the server producing the error. See [“Performing a Local Database Repair” on page 206](#) for more information.

DSMerge does not change eDirectory names or contexts within the containers. Object and property rights for the merged objects are retained.

This chapter contains the following topics:

- ♦ [“Merging eDirectory Trees” on page 189](#)
- ♦ [“Grafting a Single Server Tree” on page 195](#)
- ♦ [“Renaming a Tree” on page 199](#)

Merging eDirectory Trees

To merge eDirectory trees, use the Merge Tree Wizard in Novell iManager. This wizard lets you merge the root of two separate eDirectory trees. Only the Tree objects are merged; container objects and their leaf objects maintain separate identities within the newly merged tree.

The two trees you merge are called the source tree and the target tree. The target tree is the tree that the source tree will be merged into.

DSMerge does not change object names within the containers. Object and property rights for the merged tree are retained.

- ◆ “Prerequisites” on page 190
- ◆ “Target Tree Requirements” on page 190
- ◆ “Merging the Source into the Target Tree” on page 191
- ◆ “Partition Changes” on page 191
- ◆ “Preparing the Source and Target Trees” on page 192
- ◆ “Synchronizing Time before the Merge” on page 192
- ◆ “Merging Two Trees” on page 193
- ◆ “Post-Merge Tasks” on page 194

Prerequisites


- Novell eDirectory 8.7.3 must be installed on the server containing the master replica of the source tree’s [Root] partition.
- Other servers in the source tree should be upgraded to eDirectory 8.6 or later to ensure proper functionality.

Target Tree Requirements

- Novell eDirectory 8.7.3 must be installed on the server containing the master replica of the target tree’s [Root] partition. If this server is running any other version of NDS[®] or eDirectory, the merge operation will not complete successfully.
- Other servers in the target tree should be upgraded to eDirectory 8.6 or later to ensure proper functionality.
- You cannot maintain containers with the same name subordinate to Tree in both the source and target trees. Before merging two trees, one of the containers must be renamed.
- If both the source and target trees have a Security object, one of them must be removed before merging the trees.

Schema Requirements

Before attempting to perform a merge operation, the schema of both trees must match exactly. You should run DSRRepair on the server containing the master replica of the [Root] partition for each tree. Use the Import Remote Schema option to ensure that each tree is aware of all schema in the other tree.

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance > Schema Maintenance.
- 3** Specify which server will perform the schema maintenance operation, then click Next.
- 4** Authenticate to the specified server, then click Next.
- 5** Click Import Remote Schema > Next.
- 6** Specify the name of the tree the schema is to be imported from.
- 7** Click Start.

You might have to perform this option on both the source and target tree until no schema differences are reported; otherwise, the merge operation will not succeed.

- 8** When a “Completed” message appears with information returned from the schema maintenance operation, click Close to exit.

Merging the Source into the Target Tree

When you merge the trees, the servers in the source tree become part of the target tree.

The target Tree object becomes the new Tree object for objects in the source tree, and the tree name of all servers in the source tree is changed to the target tree's name.

After the merge, the tree name for the target tree servers is retained.

The objects that were subordinate to the source Tree object become subordinate to the target Tree object.

Partition Changes

During the merge, DSMerge splits the objects below the source Tree object into separate partitions.

All replicas of the Tree partition are then removed from servers in the source tree, except for the master replica. The server that contained the master replica of the source tree receives a replica of the target tree's Tree partition.

Figure 29 and Figure 30 illustrate the effect on partitions when you merge two trees.

Figure 29 eDirectory Trees before a Merge

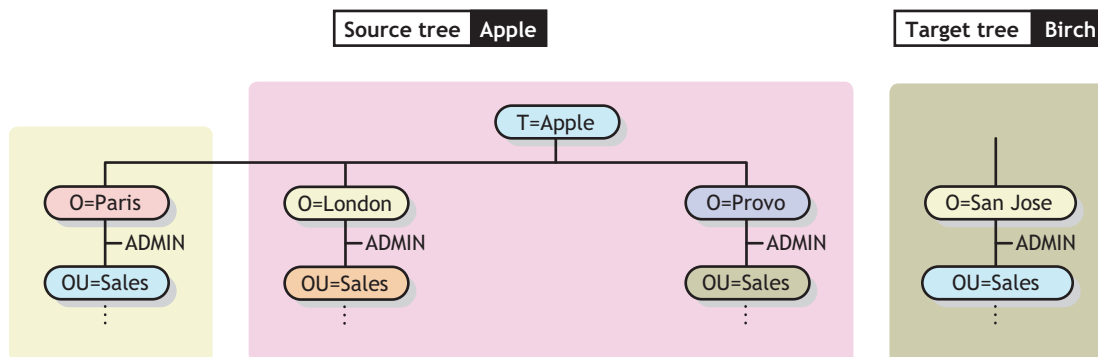
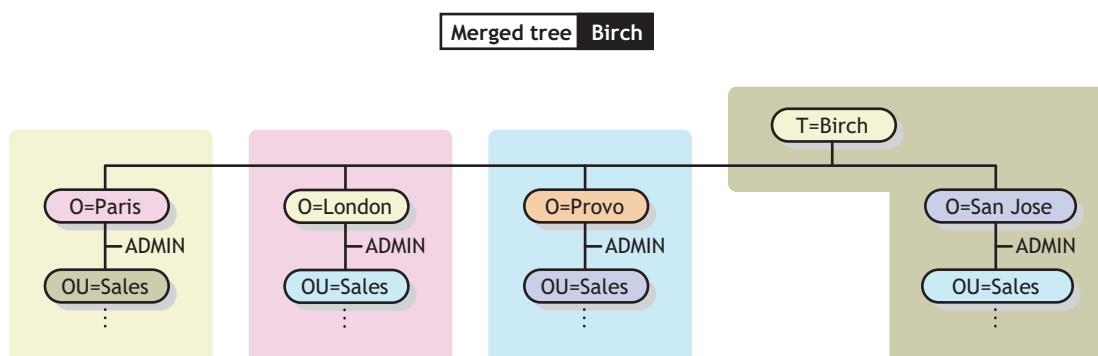


Figure 30 Merged eDirectory Tree



Preparing the Source and Target Trees

Before performing a merge operation, ensure that the state of synchronization for all servers affected by the operation is stable. The following table provides prerequisites for preparing source and target trees for merging.

Prerequisite	Required Action
WANMAN should be turned off on all servers that hold a replica of the source tree's Tree partition or the target tree's Tree partition.	Review your WANMAN policy so that WAN communication restrictions do not interfere with the merge operation. If required, turn WANMAN off before initiating the merge operation.
No aliases or leaf objects can exist at the source tree's Tree object.	Delete any aliases or leaf objects at the source tree's Tree object.
No identical names can exist between the source and target trees.	Rename objects on the source and target trees if identical names exist. Move objects from one of the containers to a different container in its tree if you don't want to rename the container objects, then delete the empty container before running DSMerge. For more information, see Chapter 3, "Managing Objects," on page 87 . You can have identical container objects in both trees if they are not immediately subordinate to the Tree object.
No login connections should exist on the source tree.	Close all connections on the source tree.
The eDirectory version must be the same on both the source and target trees.	Upgrade all non-eDirectory 8.7.3 servers that have a replica of the root partition.
The target tree must have only one copy of the root replica.	Remove all replicas on the target tree except the master replica.
The schema on both the source and target trees must be the same.	Run DSMerge. If reports indicate schema problems, use DSRepair to match the schemas. (See "Importing Remote Schema" on page 216 for more information.) Run DSMerge again.
Only one tree can have a security container subordinate to the tree root.	If both the source and target trees have a security container, remove one container as explained in Appendix A, "NMAS Considerations," on page 535 .

Because the merge operation is one single transaction, it is not subject to catastrophic failure caused by power outages or hardware failure. However, you should perform a regular backup of the eDirectory database before using DSMerge. For more information, see [Chapter 14, "Backing Up and Restoring Novell eDirectory," on page 365](#).

Synchronizing Time before the Merge

IMPORTANT: Proper configuration of time synchronization is a very involved process. Make sure you allow enough time to synchronize both trees before you merge the trees.

Novell eDirectory will not work properly if different time sources are used that have different times or if all servers in a tree are not time synchronized.

Before you do the merge, make sure that all servers in both trees are time synchronized and that they use only one time server as a time source. However, the target tree time can be ahead of the source tree time by as much as five minutes.

Generally, there should be only one Reference or one Single time server in a tree. Likewise, after the merge, the tree should contain only one Reference or one Single time server.

If each of the trees you are merging has either a Reference or a Single time server, reassign one of them to refer to the Reference or Single time server in the other tree so that the final merged tree contains only one Reference or Single time server.

For more information on time server types, see the *Network Time Management Administration Guide* (http://www.novell.com/documentation/lg/nw65/time_enu/data/h15k6r0y.html).

Merging Two Trees

For complete functionality of all menu options, run DSMerge on a server that contains the master replica of the Tree partition.

If you don't know where the master replica is stored, you will be prompted with the correct server name when you attempt an operation that requires the master replica.

To perform a merge operation, use either of the following methods:

- ◆ Novell iManager
- ◆ The eMBox command line client

For more information, see “Using the eMBox Client to Merge Trees” on page 200.

When merging large trees, it is significantly faster to designate the tree with the fewest objects immediately subordinate to the Tree object as the source tree. By doing this, you create fewer partition splits during the merge, because all objects subordinate to the Tree object result in new partitions.

Because the source tree name no longer exists after the merge, you might need to change your client workstation configurations. For the Novell Client™ for DOS/Windows, check the Preferred Tree and Preferred Server statements in the net.cfg files. For the Novell Client for Windows NT/2000 and Windows 95/98, check the Preferred Tree and Preferred Server statements on the client Property Page.

If Preferred Server is used, the client is unaffected by a tree merge or rename operation because the client still logs in to the server by name. If Preferred Tree is used and the tree is renamed or merged, then that tree name no longer exists. Only the target tree name is retained after the merge. Change the preferred tree name to the new tree name.

TIP: To minimize the number of client workstations you need to update, designate the tree with the most client workstations as the target tree, because the final tree retains the name of the target tree. Or rename the tree after the merge operation so that the final tree name corresponds to the tree with the greater number of client workstations attaching to it. For more information, see “Renaming a Tree” on page 199.

Use the following list of prerequisites to determine readiness for the merge operation:


- You have access to the source tree server through iManager
- You have the name and password of the Administrator objects that have Supervisor object rights to the Tree object of both trees you want to merge
- The eDirectory database for the two trees has been backed up
- All servers in both trees are synchronized and using the same time source

- ❑ (Optional) All servers in the tree are operational (Servers that are down will update automatically when they are operational.)
- ❑ Review the merge prerequisites listed in [“Preparing the Source and Target Trees” on page 192](#)

The merge process itself only takes a few minutes, but there are other variables that increase the length of time for the merge operation to complete:

- ◆ Many objects subordinate to the Tree object that must be split into partitions
- ◆ Many servers in the source tree that require a tree name change

To merge two trees:

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance > Merge Tree.
- 3** Specify which server will run Merge (this will be the source tree), then click Next.
- 4** Authenticate to the server, then click Next.
- 5** Specify an Administrator username and password for the source tree.
- 6** Specify the target tree name and the Administrator username and password, then click Start.
A Merge Tree Wizard Status window appears and shows the progress of the merge.
- 7** When a “Completed” message appears with information returned from the merge process, click Close to exit.

Post-Merge Tasks

Following the merging of two trees, it might be necessary to complete the following steps:

- 1** Verify that all tree names were changed correctly.
- 2** Check the new partitions that the merge operation created.
If you have many small partitions in the new tree, or if you have partitions that contain related information, you might want to merge them. For more information, see [“Merging a Partition” on page 115](#).
- 3** Copy a new replica to any non-NetWare 5 servers, if you did not upgrade before running DSMerge.
- 4** Re-create any leaf objects or aliases in the tree that were deleted before you ran DSMerge.
- 5** Evaluate partitioning of the eDirectory tree.
Merging trees might change replica placement requirements on the new tree. You should carefully evaluate and change the partitioning as needed.
- 6** Update your client workstation configuration.
For the Novell Client for DOS/Windows, check the Preferred Tree and Preferred Server statements in the net.cfg files. For the Novell Client for Windows NT/2000 and Windows 95/98, check the Preferred Tree and Preferred Server statements on the client Property Page, or rename the target tree.

If Preferred Server is used, the client is unaffected by a tree merge or rename operation because the client still logs in to the server by name. If Preferred Tree is used and the tree is renamed or merged, then that tree name no longer exists. Only the target tree name is retained after the merge. Change the preferred tree name to the new tree name.

The Access Control List (ACL) for the Tree object of the source tree is preserved. Therefore, the rights of the source tree's user Admin to the Tree object are still valid.

After the merge is complete, both admin users still exist and are uniquely identified by different container objects.

For security reasons, you might want to delete one of the two Admin User objects or restrict the rights of the two objects.

Grafting a Single Server Tree

The Graft Tree option lets you graft a single server source tree's Tree object under a container specified in the target tree. After the graft is completed, the source tree receives the target tree's name.

During the graft, DSmerge changes the object class of the source tree's Tree object to Domain and makes a new partition. The new Domain object is the partition root for the new partition. All the objects under the source tree's Tree object are located under the Domain object.

The target tree's administrator has rights to the resulting tree's root container and, therefore, has rights to the source tree's grafted root.

NOTE: It might take up to several hours for the inherited rights to be recalculated and become effective. This time will vary based on the tree's complexity, size, and number of partitions.

The source tree's administrator has rights only in the newly created Domain object.

Figure 31 and Figure 32 on page 196 illustrate the effects of grafting a tree into a specific container.

Figure 31 eDirectory Trees before a Graft

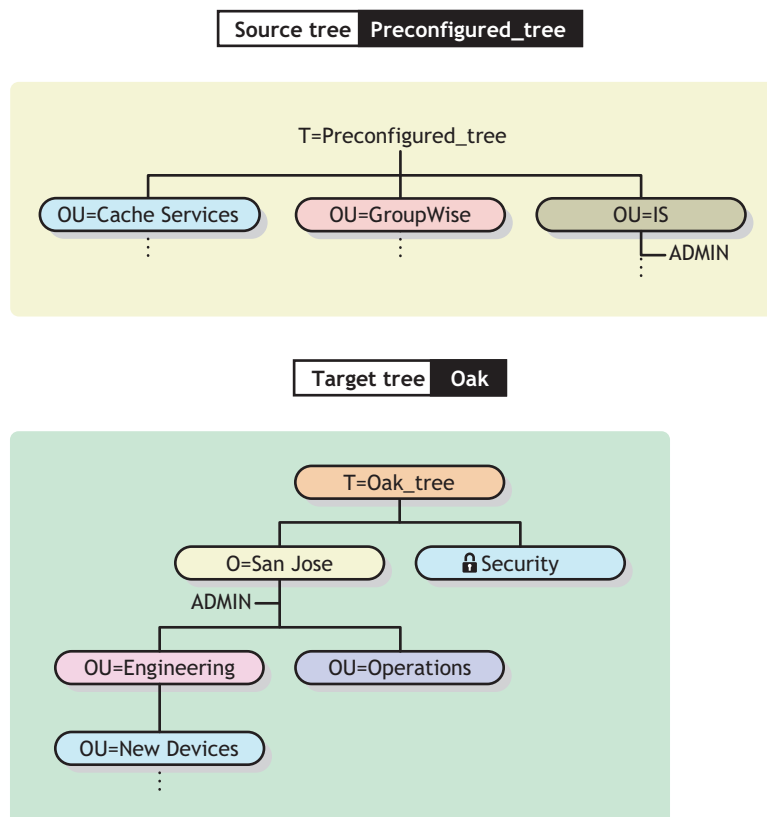
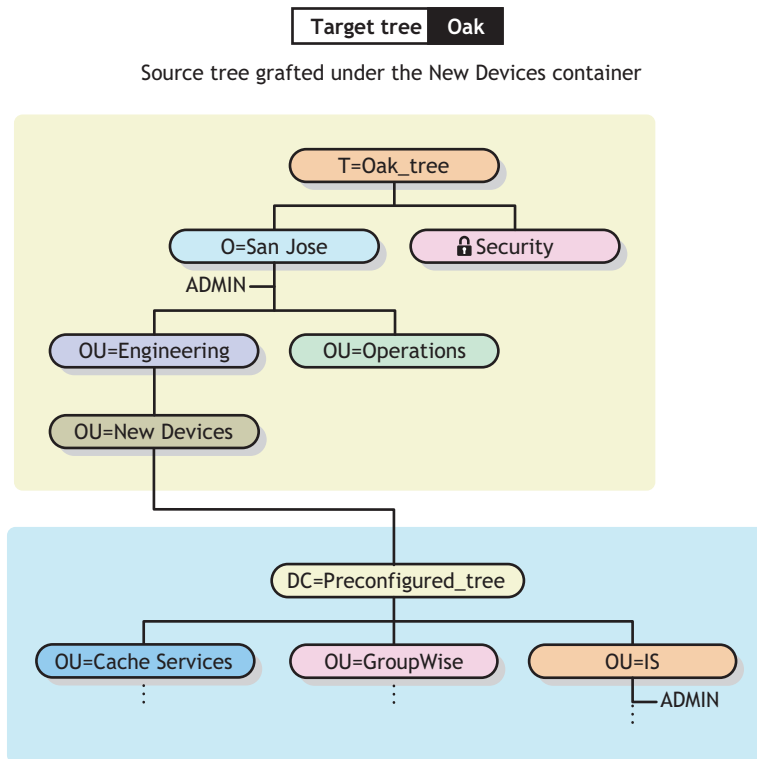


Figure 32 Grafted eDirectory Tree



This sections contains the following information:

- ◆ “Understanding Context Name Changes” on page 196
- ◆ “Preparing the Source and Target Trees” on page 197
- ◆ “Containment Requirements for Grafting” on page 198
- ◆ “Grafting the Source and Target Tree” on page 199

Understanding Context Name Changes

After the source tree has been grafted into the target tree container, the distinguished names for objects in the source tree will be appended with the source tree’s name followed by the distinguished name of the target tree’s container name where the source tree was merged. The relative distinguished name will remain the same.

For example, if you are using dot delimiters, the typeful name for Admin in the Preconfigured_tree (source tree) is

```
CN=Admin.OU=IS.T=Preconfigured_tree
```

After the Preconfigured_tree is merged into the New Devices container in the Oak_tree, the typeful name for Admin is

```
CN=Admin.OU=IS.DC=Preconfigured_tree.OU=Newdevices.
OU=Engineering.O=Sanjose.T=Oak_tree.
```

NOTE: The maximum length for a distinguished name should not exceed 256 characters. This limitation is particularly important when you are grafting the root of one tree into a container near the bottom of the target tree.

The last dot following Oak_tree (Oak_tree.) indicates that the last element in the distinguished name is the tree name. If you leave off the trailing dot, then also leave off the tree name.

Preparing the Source and Target Trees

Before initiating the graft operation, ensure that the state of all of the servers affected by the operation is stable. The following table provides prerequisites for preparing the source and target trees before grafting.

Prerequisite	Required Action
WANMAN should be turned off on all servers that hold a replica of the source tree's Tree partition or the target tree's Tree partition.	Review your WANMAN policy so that WAN communication restrictions do not interfere with the merge operation. If required, turn WANMAN off before initiating the merge operation.
The source tree must have only one server.	Remove all but one server from the source tree.
No aliases or leaf objects can exist at the source tree's Tree object.	Delete any aliases or leaf objects at the source tree's Tree object.
No similar names can exist in the graft container.	Rename objects under the target tree graft container or rename the source tree. Move objects from one of the containers to a different container in its tree if you don't want to rename objects, then delete the empty container before running DSMerge. For more information, see Chapter 3, "Managing Objects," on page 87 . You can have identical container objects in both trees if they are not immediately subordinate to the same parent object. Objects are uniquely identified by their immediate container object.
The eDirectory version for both the source tree and target tree container must be 8.51 SP2a or later.	DSMerge will search for the appropriate version of eDirectory. If an acceptable version isn't found, DSMerge will return an error. You can get the latest version of eDirectory from the Novell Download page (http://download.novell.com) .
The container where you will join the target tree is in a partition that has no replicas (a single-server partition).	If the target container has multiple replicas, do <i>one</i> of the following: <ul style="list-style-type: none"> ◆ Make the partition associated with this container the master replica and delete other replicas. ◆ Split the target tree graft container into a separate partition and remove replicas. After the graft is complete, the partition association can be re-established.
The server holding the target container must also hold a replica of the ROOT partition.	If the server doesn't hold a replica of ROOT, the graft will fail and you will see error <code>-672 No Access</code> because the directory is unable to verify administrator rights for the target tree. Use iManager to add a replica for ROOT. For more information, see "Adding a Replica" on page 117 .

Prerequisite	Required Action
The schema on both the source and target trees must be the same.	<p>Run the Graft option in DSMerge. If reports indicate schema problems, run DSRepair on the target tree to import the schema from the source tree.</p> <p>The graft operation automatically imports the schema from the target tree to the source tree.</p> <p>Run DSMerge again.</p>
Only one tree can have a security container subordinate to the tree root.	If both the source and target tree have the security container, remove one container as explained in Appendix A, "NMAS Considerations," on page 535.
The source tree's time reference must be reconfigured.	<p>The source tree should usually be set as a secondary server configured to get its time source from a server in the target tree.</p> <p>To reconfigure Timesync, see Configuring Timesync on Servers (http://www.novell.com/documentation/lg/nw65/time_enu/data/abzqzx2.html) in the <i>Network Time Management Administration Guide</i>.</p>


Containment Requirements for Grafting

To graft a source tree into a target tree container requires that the target tree container be prepared to accept the source tree. The target tree container must be able to contain an object of the class domain. If there is a problem with containment, error -611 `Illegal Containment` will occur during the graft operation.

Use the information in the following table to determine if you need to run DSRepair to modify containment lists.

Target Tree Container Requirements	<p>The target tree container object must include the domain object in its containment list.</p> <p>You can check this using iMonitor > Schema. If the containment list does not include Domain, run DSRepair to make schema enhancements.</p>
Source Tree Requirements	<p>The graft operation changes the source tree root from the class Tree Root to the class Domain. All of the object classes that are subordinate to the Tree must be able to be contained by the class Domain according to the schema rules.</p> <p>You can check this using iMonitor > Schema. If the containment list does not include Domain, run DSRepair to make schema enhancements.</p>


If containment requirements aren't met, run DSRepair to correct the schema.

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance > Schema Maintenance.
- 3** Specify the server that will perform the operation, then click Next.
- 4** Specify a user name, password, and context for the server where you will be performing the operation, then click Next.

- 5 Click Optional Schema Enhancements, then click Start.
- 6 Follow the online instructions to complete the operation.

Grafting the Source and Target Tree

After you ensure that prerequisites are met, use DSMerge to perform the graft.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance > Graft Tree.
- 3 Specify which server will run Graft (this will be the source tree), then click Next.
- 4 Authenticate to the server, then click Next.
- 5 Specify the source tree Administrator name and password and the target tree name, Administrator name, and Password.
- 6 Click Start.

A Graft Tree Wizard Status window appears, showing the progress of the graft. A “Completed” message finally appears with information returned from the graft process.

- 7 Click Close to exit.

Renaming a Tree

You must rename a tree if the two trees you want to merge have the same name.

You can rename only the source tree. To rename the target tree, run the Rename Tree Wizard in Novell iManager against a server on the target tree.

If you change a tree name, the bindery context does not automatically change. Because the bindery context set in the autoexec.ncf file also contains the tree name (for example, SET Bindery Context = O=n.test_tree_name), a server with a recently changed tree name does not use the context that it used before the tree name change.

Therefore, after you change a tree’s name, you might need to change your client workstation configurations. For the Novell Client for DOS/Windows, check the Preferred Tree and Preferred Server statements in the net.cfg files. For Novell Client for Windows NT/2000 and Windows 95/98, check the Preferred Tree and Preferred Server statements on the client Property Page.

If Preferred Server is used, the client is unaffected by a tree merge or rename operation because the client still logs in to the server by name. If Preferred Tree is used and the tree is renamed or merged, then that tree name no longer exists. Only the target tree name is retained after the merge. Change the preferred tree name to the new tree name.

When you merge two trees, to minimize the number of client workstations that need to be updated, designate the tree with the most client workstations as the target tree because the final tree retains the name of the target tree.


You can also rename the tree after the merge so that the final tree name corresponds to the tree name with the majority of client workstations.

Another option is to rename the merged tree to the name of the original source tree. If you choose this option, then you must update the net.cfg files on the target tree client workstations.

Use the following list of prerequisites to determine readiness for the renaming operation:

- Access to a server console on the source tree or an established RCONSOLE session with the server
- The Supervisor object right to the Tree object of the source tree
- (Optional) All servers in the tree are operational (Servers that are down will update automatically when they are operational.)

To rename the tree:

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance > Rename Tree.
- 3** Specify which server will run the Rename Tree Wizard (this should be a server in the target tree), then click Next.
- 4** Authenticate to the server, then click Next.
- 5** Specify a new tree name and an Administrator username and password.
- 6** Click Start.
A Rename Tree Wizard Status window appears, showing the progress of the Rename process.
- 7** When a “Completed” message appears with information returned from the Rename process, click Close to exit.

Using the eMBox Client to Merge Trees

The eDirectory Management Toolbox (eMBox) Client is a command line Java client that gives you remote access to DSmerge. The `emboxclient.jar` file is installed on your server as part of eDirectory. You can run it on any machine with a JVM. For more information on the eMBox Client, see [“Using the eMBox Command Line Client” on page 465](#).

Using the DSmerge eMTool

- 1** Run the eMBox Client in interactive mode by entering the following at the command line:

```
java -cp path_to_the_file/emboxclient.jar embox -i
```

(If you have already put the `emboxclient.jar` file in your class path, you need to enter only `java embox -i`.)

The eMBox Client prompt appears:

```
eMBox Client>
```

- 2** Log in to the server that will run DSmerge (this will be the source tree) by entering the following:

```
login -sserver_name_or_IP_address -pport_number  
-uusername.context -wpassword -n
```

The port number is usually 80 or 8008, unless you have a Web server that is already using the port. The `-n` option opens a nonsecure connection.

The eMBox Client will indicate whether the login is successful.

- 3** Enter a merge command, using the following syntax:

```
dsmerge.task options
```

For example:

`dsmerge.m -uadmin -ptest -TApple -Uadmin -Ptest` merges the target tree Apple (with target tree username Admin and user password test) with the source tree you are currently logged in to (with source tree username Admin and user password test).

`dsmerge.g -uadmin -ptest -TOrange -Uadmin -Ptest -CFruit` grafts the source tree you are currently logged in to (with source tree username Admin and user password test) into the Fruit container in the target tree Orange (with target tree username Admin and user password test).

A space must be between each switch. The order of the switches is not important.

The eMBox Client will indicate whether the DSMerge operation was successful.

See “[DSMerge eMTool Options](#)” on page 201 for more information on the DSMerge eMTool options.

- 4** Log out from the eMBox Client by entering the following command:

```
logout
```

- 5** Exit the eMBox Client by entering the following command:

```
exit
```

DSMerge eMTool Options

The following tables lists the DSMerge eMTool options. You can also use the list `-tdsmerge` command in the eMBox Client to list the DSMerge options with details. See “[Listing eMTools and Their Services](#)” on page 469 for more information.

Merge Operation	eMBox Client Command
Check whether the tree can be renamed	<code>dsmerge.pr -uUser -pUser_password -nNew_tree_name</code>
Rename the tree	<code>dsmerge.r -uUser -pUser_password -nNew_tree_name</code>
Check whether two trees can be merged	<code>dsmerge.pm -uSource_tree_user -pSource_tree_user_password -TTarget_tree_name -UTarget_tree_user -PTarget_tree_password</code>
Merge two trees	<code>dsmerge.m -uSource_tree_user -pSource_tree_user_password -TTarget_tree_name -UTarget_tree_user -PTarget_tree_password</code>
Check whether the source tree can be grafted into the target tree container	<code>dsmerge.pg -uSource_tree_user -pSource_tree_user_password -TTarget_tree_name -UTarget_tree_user -PTarget_tree_password -CTarget_tree_container</code>
Graft the source tree into the container in the target tree	<code>dsmerge.g -uSource_tree_user -pSource_tree_user_password -TTarget_tree_name -UTarget_tree_user -PTarget_tree_password -CTarget_tree_container</code>
Cancel the running dsmerge operation	cancel

9

Repairing the Novell eDirectory Database

The Repair utility lets you maintain and repair the database of a Novell® eDirectory™ tree. This utility performs the following operations:

- ◆ Corrects eDirectory problems such as bad records, schema mismatches, bad server addresses, and external references.
- ◆ Makes advanced changes to the eDirectory schema.
- ◆ Checks the structure of the database automatically without closing the database and without user intervention.
- ◆ Checks the database operational indexes.
- ◆ Reclaims free space by discarding empty records.
- ◆ Repairs the local database.
- ◆ Repairs replicas, replica rings, and Server objects.
- ◆ Analyzes each server in each local partition for synchronization errors.
- ◆ Locates and synchronizes objects in the local database.

Some eDirectory database problems are not fatal, and eDirectory will continue to operate. But if the database becomes corrupted, you will get a message on the console that the server could not open the local database. In this case, run Repair or contact Novell Support.

Novell does not recommend running repair operations unless you run into problems with eDirectory, or are told to do so by Novell Support. However, you are encouraged to use the diagnostic features available in Repair and in other Novell utilities such as Novell iMonitor. For more information, see [Chapter 7, “Using Novell iMonitor 2.1,” on page 163](#).

Novell iManager contains the following Repair Wizards:

Wizard	Description
Basic Repair Wizard	Lets you perform an unattended full repair, local database repair, or single object repair. You can also check external references and delete unknown leaf objects.
Log File Wizard	Lets you open the repair log file and set log file options.
Repair via iMonitor	Lets you open iMonitor and use the repair options available in that program.
Replica Repair Wizard	Lets you repair all or selected replicas, repair time stamps and declare a new epoch, designate the current server as the new master replica, and destroy the selected replica, if necessary.

Wizard	Description
Replica Ring Repair Wizard	Lets you repair all or selected replica rings, send all objects to every server in the ring, receive all objects from the master to the selected replica, and remove the current server from the replica ring, if necessary.
Schema Maintenance Wizard	Lets you request schema from the tree, reset the local schema, declare a new schema epoch, perform optional schema enhancements, import remote schema, declare a new schema epoch, and perform a post NetWare® 5 schema update.
Server Repair Wizard	Lets you repair all network addresses, or repair only a server's network addresses.
Sync Repair Wizard	Lets you synchronize the selected replica on the current server, report the synchronization status on the current server, report the synchronization status on all servers, perform a time synchronization, and schedule an immediate synchronization.

The wizards help you with the following operations:

- ◆ [“Performing Basic Repair Operations” on page 204](#)
- ◆ [“Viewing and Configuring the Repair Log File” on page 208](#)
- ◆ [“Performing a Repair in Novell iMonitor” on page 209](#)
- ◆ [“Repairing Replicas” on page 210](#)
- ◆ [“Repairing Replica Rings” on page 212](#)
- ◆ [“Maintaining the Schema” on page 214](#)
- ◆ [“Repairing Server Network Addresses” on page 217](#)
- ◆ [“Performing Synchronization Operations” on page 219](#)
- ◆ [“Advanced DSRepair Options” on page 221](#)
- ◆ [“Using the eMBox Client to Repair a Database” on page 225](#)

Performing Basic Repair Operations

The Basic Repair Wizard lets you perform an unattended full repair, local database repair, or single object repair. You can also check external references and delete unknown leaf objects.

- ◆ [“Performing an Unattended Full Repair” on page 205](#)
- ◆ [“Performing a Local Database Repair” on page 206](#)
- ◆ [“Checking External References” on page 207](#)
- ◆ [“Repairing a Single Object” on page 207](#)
- ◆ [“Deleting Unknown Leaf Objects” on page 208](#)

Performing an Unattended Full Repair

An unattended full repair checks for and repairs most critical eDirectory errors in the eDirectory database files of a given server. It performs eight primary operations each time it is run, none of which require any intervention by the administrator. During some of these operations, the local database is locked. An unattended full repair builds a temporary set of local database files and runs the repair operation against those files. That way, if a serious problem develops, the original files are still intact.

This is the suggested means of repair if you are not familiar with the interactive options of the Local Database Repair. Running the Unattended Full Repair might require twice the amount of disk space currently used by the database files. See [“Performing a Local Database Repair” on page 206](#) for more information.


Rebuilding the operational indexes used by eDirectory is possible only when the local database is locked.

The following table lists the operations performed during an unattended full repair:

Operation	Database Locked?	Description
Database Structure and Index Checked	Yes	Reviews the structure and format of database records and indexes. This ensures that no structural corruption has been introduced into the eDirectory environment at the database level.
Rebuild the Entire Database	Yes	Resolves errors found during structure and index checks. It restores proper data structures and re-creates the eDirectory database and index files.
Perform Tree Structure Check	Yes	Examines the links between database records to make sure that each child record has a valid parent. This helps ensure database consistency. Invalid records are marked so that they can be restored from another partition replica during the eDirectory replica synchronization process.
Repair All Local Replicas	Yes	Resolves eDirectory database inconsistencies by checking each object and attribute against schema definitions. It also checks the format of all internal data structures. This operation can also resolve inconsistencies found during the tree structure check by removing invalid records from the database. As a result, all child records linked through the invalid record are marked as orphans. These orphan records are not lost, but this process could potentially generate a large number of errors while the database is being rebuilt. This is normal, and the orphan objects will be automatically reorganized over the course of replica synchronization.
Check Local References	Yes	Local References are pointers to other objects maintained in the eDirectory database on this file server. This operation evaluates the internal database pointers to make sure that they are pointing to the correct eDirectory objects. If invalid references are found, they are corrected. This operation might take a long time to complete, depending on how many inter-object relationships exist.

Operation	Database Locked?	Description
Repair Network Addresses	No	Checks server network addresses stored in eDirectory against the values maintained in local SAP, SLP, or DNS tables to make sure that eDirectory still has accurate information. If a discrepancy is found, eDirectory is updated with the correct information.
Validate Stream Syntax Files	Yes	Stream Syntax Files, such as login scripts, are stored in a special area of the eDirectory database. This operation checks to make sure that each stream syntax file is associated with a valid eDirectory object. If not, the stream syntax file is deleted and the attribute referencing it is purged.
Validate Mail Directories (NetWare Only)	Yes	By default, eDirectory creates mail directories in the sys:mail directory of NetWare® servers in order to support legacy bindery users. Login scripts for bindery users are stored in the user's mail directory. This operation checks to make sure that each mail directory is associated with a valid eDirectory User object. If not, the mail directory is deleted.
Check Volume Objects And Trustees (NetWare Only)	No	<p>Ensures that each volume on the NetWare server is associated with a Volume object in eDirectory. If not, it searches the context that the server resides in to see whether a Volume object exists. If no Volume object exists, one is created.</p> <p>After validating the volume information, a list of trustee IDs is validated. Each object in eDirectory has a unique trustee ID. This ID is used to grant rights to other objects, including NetWare volumes, in the eDirectory tree. This task ensures that each trustee ID in the volume list is a valid eDirectory object. If not, the trustee ID is removed from the volume list.</p>

To perform an unattended full repair:

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance Utilities > Basic Repair.
- 3** Specify the server that will perform the operation, then click Next.
- 4** Specify a user name, password, and context for the server where you will perform the operation, then click Next.
- 5** Click Unattended Full Repair, then click Start.
- 6** Follow the online instructions to complete the operation.

Performing a Local Database Repair


Use this repair operation to resolve inconsistencies in the local database so that it can be opened and accessed by eDirectory.

A local database repair can be performed on a temporary set of files if you specifically request it. Otherwise, the repair operation will take place on the live database.

Performing the repair operation on a temporary set of database files requires closing the database during this part of the operation. If you choose to work on a temporary set of files, you will be

prompted to commit the repair modifications before they are made permanent. Otherwise, changes take place immediately.


Following a repair operation, you can view a log of the repair operations to determine if further operations are required to complete the repair. For more information, see [“Viewing and Configuring the Repair Log File” on page 208](#).

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance Utilities > Basic Repair.
- 3 Specify the server that will perform the operation, then click Next.
- 4 Specify a user name, password, and context for the server where you will perform the operation, then click Next.
- 5 Click Local Database Repair, then click Next.
- 6 Specify the options you want for running the local repair, then click Start.
- 7 Follow the online instructions to complete the operation.

Checking External References

This repair operation checks each external reference object to determine if a replica containing the object can be located. If all the servers containing a replica of the partition that the object is in are inaccessible, the object will not be found. If the object cannot be found, a warning is posted.

This operation also provides obituary information.


- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance Utilities > Basic Repair.
- 3 Specify the server that will perform the operation, then click Next.
- 4 Specify a user name, password, and context for the server where you will perform the operation, then click Next.
- 5 Click Check External References, then click Start.
- 6 Follow the online instructions to complete the operation.

Repairing a Single Object

This repair operation will try to resolve any inconsistencies in an eDirectory object which might be preventing eDirectory from accessing such data. This operation works only on user-created partitions and on the external reference partition.

This operation is performed on the live database files. If the corruption is at the physical level, you might need to perform a Physical and Structure check before the Single Object Repair is run.

Make sure you always have a current backup copy of the eDirectory database.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance Utilities > Basic Repair.
- 3 Specify the server that will perform the operation, then click Next.
- 4 Specify a user name, password, and context for the server where you will perform the operation, then click Next.


- 5 Click Single Object Repair, then click Start.
- 6 Specify the object you want to repair, then click Next.
- 7 Follow the online instructions to complete the operation.

Deleting Unknown Leaf Objects

Repair changes inconsistent objects to Unknown objects when they do not have mandatory properties or are invalid in other respects (their properties don't meet minimum requirements for an object type). Unknown objects are real objects and eDirectory knows about them. They are unknown because their object class cannot be fully validated. Unknown objects, represented by question mark icons, can be deleted but cannot easily be changed back to their original object type.

This repair operation deletes all objects in the local eDirectory database that have the Unknown object class and maintain no subordinate objects. The deletion is later synchronized to other replicas in the eDirectory tree.

IMPORTANT: This operation should not be run unless you understand the consequences or have been advised by Novell Support to run it.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance Utilities > Basic Repair.
- 3 Specify the server that will perform the operation, then click Next.
- 4 Specify a user name, password, and context for the server where you will perform the operation, then click Next.
- 5 Click Delete Unknown Leaf Objects, then click Start.
- 6 Follow the online instructions to complete the operation.

Viewing and Configuring the Repair Log File

The Repair log file contains detailed information about local partitions and servers. This information helps you diagnose damage to the database. The Log File Wizard lets you open the repair log file and set log file options.


This sections contains information on the following operations:

- ♦ [“Opening the Log File” on page 208](#)
- ♦ [“Setting Log File Options” on page 209](#)

Opening the Log File

Use this operation to view your repair log file. The default name of the file is dsrepair.log. The results of the operations performed by your repairs are written to it.


You can turn the log file operation off or on, change the name, and delete or reset the log file. See [“Setting Log File Options” on page 209](#) for more information.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance Utilities > Log File.
- 3 Specify the server that will perform the operation, then click Next.

- 4** Specify a user name, password, and context for the server where you will perform the operation, then click Next.
- 5** Click Open Log File, then click Start.
- 6** Follow the online instructions to complete the operation.

Setting Log File Options

Use this operation to manage the repair log file. You can turn the log file on or off, delete the log file, append the log file, or change the filename.


- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance Utilities > Log File.
- 3** Specify the server that will perform the operation, then click Next.
- 4** Specify a user name, password, and context for the server where you will perform the operation, then click Next.
- 5** Click Log File Options, then click Next.
- 6** Follow the online instructions to complete the operation.

Performing a Repair in Novell iMonitor

You can access Repair features by using the Repair Via iMonitor option in Novell iManager. The Repair page in iMonitor lets you view problems and back up or clean up your eDirectory database.

In iMonitor, DSRepair is a server-centric feature. In other words, this feature is available only on the local server where iMonitor is running. If you need to access this feature on another server, you must switch to the iMonitor running on that server.

You must be the equivalent of the Administrator of the server or a console operator on the server where you are trying to access the DS Repair page. For this reason, you must first log in so your credentials can be verified before you can access information on this page.

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance Utilities > Repair via iMonitor.
- 3** Specify the server that will perform the operation, then click OK.
To open iMonitor and run the repair options manually, click Run iMonitor and Let Me Access Repair from There before you click OK.
- 4** Specify a user name, context, and password for the server you are trying to access, then click OK to open the iMonitor Repair page.
- 5** Select the repair options you want, then click Start Repair.

For more information on using the repair features available in iMonitor, see [“Viewing DSRepair Information” on page 177](#).

Repairing Replicas

Repairing a replica consists of checking each object in the replica for consistency with the schema, and checking each attribute of the object for consistency with the schema and the data according to the syntax of the attribute. Other internal data structures associated with the replica are also checked.


Use the Replica Repair Wizard to perform the following operations:

- ◆ “Repairing All Replicas” on page 210
- ◆ “Repairing Selected Replicas” on page 210
- ◆ “Repairing Time Stamps” on page 211
- ◆ “Designating This Server As the New Master Replica” on page 212
- ◆ “Destroying the Selected Replica” on page 212

Repairing All Replicas

This operation repairs all of the replicas displayed in the replica table.


If you have not performed a Local Database Repair operation on the local eDirectory database within the last 30 minutes, you should do so before performing this operation. See “Performing a Local Database Repair” on page 206 for more information.

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance Utilities > Replica Repair.
- 3** Specify the server that will perform the operation, then click Next.
- 4** Specify a user name, password, and context for the server where you will perform the operation, then click Next.
- 5** Click Repair All Replicas, then click Start.
- 6** Follow the online instructions to complete the operation.

Repairing Selected Replicas

This operation repairs only the selected replica listed in the replica view.

If you have not performed a Local Database Repair operation on the local eDirectory database within the last 30 minutes, you should do so before performing this operation. See “Performing a Local Database Repair” on page 206 for more information.

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance Utilities > Replica Repair.
- 3** Specify the server that will perform the operation, then click Next.
- 4** Specify a user name, password, and context for the server where you will perform the operation, then click Next.
- 5** Click Repair the Selected Replica, then click Next.
- 6** Specify the replica you want to repair, then click Start.
- 7** Follow the online instructions to complete the operation.

Repairing Time Stamps

NOTE: Before using this operation, use the Sync Repair Wizard to make sure that all servers in the replica ring are communicating properly. See [“Performing Synchronization Operations” on page 219](#) for more information.

This operation provides a new point of reference to the master replica so that all updates to replicas of the selected partition are current.

This operation is always performed on the master replica of a partition. The master replica does not need to be the local replica on this server.


Time stamps are placed on objects when they are created or modified, and they must be unique. All time stamps in a master replica are examined. If any time stamps are postdated to the current network time, they are replaced with a new time stamp. If the time stamp is current, a new time stamp is not issued. After all time stamps are consistent in time, a new epoch is declared.

Use this operation if you notice a discrepancy between objects in a replica or in an object's properties. For example, if you update your login script but your old login script still appears when logging in, you should check to ensure that replicas are synchronizing properly. If the differences between the time stamps in the future and the current time is not more than minutes, eDirectory will eventually correct the condition by itself. Declaring a new epoch is a very expensive operation, and should not be used regularly.

Novell eDirectory is a loosely consistent database, so you should allow for five to ten minutes before checking replica synchronization. This operation results in the following conditions:

- ◆ A new epoch is declared on the master replica, possibly affecting all objects in the replica.
- ◆ All time stamps are examined and repaired as required.
- ◆ Updates are not accepted from replicas with postdated time stamps (epochs) until the replicas are synchronized.
- ◆ A replica receives a copy of all objects in a master replica or any other replica that has received a new epoch.
- ◆ The replica becomes the same epoch as the master replica.
- ◆ Any modifications from a previous epoch are lost.
- ◆ The master replica does not need to reside on the current server, but you must have the Supervisor right to the master replica to perform the repair operation.
- ◆ The other replicas are put in a new state.


To repair time stamps and declare a new epoch:

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance Utilities > Replica Repair.
- 3** Specify the server that will perform the operation, then click Next.
- 4** Specify a username, password, and context for the server where you will perform the operation, then click Next.
- 5** Click Repair Timestamps and Declare a New Epoch, then click Next.
- 6** Follow the online instructions to complete the operation.

Designating This Server As the New Master Replica

This operation designates the local replica of the selected partition as the master replica. You can use this operation to designate a new master replica if the original one is lost. A master can be lost if the server that contains the master replica has a hard disk failure and must be replaced.


Do not use this option to perform the normal partition operations available in Novell iManager. For more information, see [Chapter 5, “Managing Partitions and Replicas,” on page 113](#).

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance Utilities > Replica Repair.
- 3 Specify the server you want to designate as the new master replica, then click Next.
- 4 Specify a user name, password, and context to authenticate to the server, then click Next.
- 5 Click Designate This Server As the New Master Replica, then click Next.
- 6 Follow the online instructions to complete the operation.

Destroying the Selected Replica

Use this operation to remove the selected replica from this server. The replica will be deleted or changed to a subordinate reference.

Do not use this option to perform the normal partition operations available in Novell iManager. For more information, see [Chapter 5, “Managing Partitions and Replicas,” on page 113](#).

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance Utilities > Replica Repair.
- 3 Specify the server containing the replica you want to destroy, then click Next.
- 4 Specify a user name, password, and context to authenticate to the server, then click Next.
- 5 Click Destroy the Selected Replica, then click Next.
- 6 Specify the replica you want to destroy, then click Next.
- 7 Follow the online instructions to complete the operation.

Repairing Replica Rings

Repairing a replica ring consists of checking the replica ring information on each server that contains a replica and validating remote ID information.


Use the Replica Ring Repair Wizard to perform the following operations:

- ♦ [“Repairing All Replica Rings” on page 213](#)
- ♦ [“Repairing the Selected Replica Ring” on page 213](#)
- ♦ [“Sending All Objects to Every Server in the Ring” on page 213](#)
- ♦ [“Receiving All Objects from the Master to the Selected Replica” on page 214](#)
- ♦ [“Removing This Server from the Replica Ring” on page 214](#)

Repairing All Replica Rings

This operation repairs the replica ring of all the replicas displayed in the replica view.


If you have not performed a Local Database Repair operation on the local eDirectory database within the last 30 minutes, you should do so before performing this operation. See [“Performing a Local Database Repair” on page 206](#) for more information.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance Utilities > Replica Ring Repair.
- 3 Specify the server that will perform the operation, then click Next.
- 4 Specify a user name, password, and context for the server where you will perform the operation, then click Next.
- 5 Click Repair All Replica Rings, then click Next.
- 6 Follow the online instructions to complete the operation.

Repairing the Selected Replica Ring

This operation repairs the replica ring of the selected replica listed in the replica table.

If you have not performed a Local Database Repair operation on the local eDirectory database within the last 30 minutes, you should do so before performing this operation. See [“Performing a Local Database Repair” on page 206](#) for more information.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance Utilities > Replica Ring Repair.
- 3 Specify the server that will perform the operation, then click Next.
- 4 Specify a user name, password, and context for the server where you will perform the operation, then click Next.
- 5 Click Repair the Selected Replica Ring, then click Next.
- 6 Specify the replica you want to repair, then click Next.
- 7 Follow the online instructions to complete the operation.


Sending All Objects to Every Server in the Ring

This operation sends all objects from the selected server in the replica ring to all other servers that contain a replica of the partition.

Use this operation to ensure that the selected partition's replica on the selected server in the replica ring is synchronized with all other servers in the replica ring. This operation cannot be performed on a server that contains only a subordinate reference replica of the partition.

Modifications that have been made to other replicas that have not yet synchronized with the replica on the selected server will be lost. You should verify the synchronization status before performing this operation.

IMPORTANT: This operation can cause heavy network traffic because of the re-creation of the objects in the replica. It is not a diagnostic operation.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance Utilities > Replica Ring Repair.


- 3 Specify the server that will perform the operation, then click Next.
- 4 Specify a user name, password, and context for the server, then click Next.
- 5 Click Send All Objects to Every Server in the Ring, then click Next.
- 6 Follow the online instructions to complete the operation.

Receiving All Objects from the Master to the Selected Replica

This operation receives all objects from the master replica to the replica on the selected servers.

Use this operation to ensure that the selected partition's replica on the selected server in the replica ring is synchronized with the master replica. This operation cannot be performed on a server that contains the master replica.


IMPORTANT: This operation can produce a lot of network traffic. By requesting this operation, the current replica will behave as if a new replica is being placed on the server. It will also put the replica in a new state.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance Utilities > Replica Ring Repair.
- 3 Specify the server that will perform the operation, then click Next.
- 4 Specify a user name, password, and context for the server, then click Next.
- 5 Click Receive All Objects from the Master to the Selected Replica, then click Next.
- 6 Follow the online instructions to complete the operation.

Removing This Server from the Replica Ring

This operation removes the specified server from the selected replica stored on the current server.

WARNING: Misuse of this operation can cause irrevocable damage to the eDirectory database. You should not use this operation unless directed to by Novell Support personnel.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance Utilities > Replica Ring Repair.
- 3 Specify the server that will perform the operation, then click Next.
- 4 Specify a user name, password, and context for the server, then click Next.
- 5 Click Remove This Server from the Replica Ring, then click Next.
- 6 Follow the online instructions to complete the operation.

Maintaining the Schema

The schema is a system of rules and definitions for object attributes that establishes the content and format of each object and the object's relationships in the database.

The Schema Maintenance Wizard contains several schema operations that might be necessary to bring an eDirectory server's schema into compliance with the master of [Root]. However, these operations should be used only when necessary. The local and unattended repair operations already verify the schema.

For more information on the eDirectory schema, see [Chapter 4, "Managing the Schema," on page 103](#).


Use the Schema Maintenance Wizard to perform the following operations:

- ♦ “Requesting Schema from the Tree” on page 215
- ♦ “Resetting the Local Schema” on page 215
- ♦ “Performing a Post-NetWare 5 Schema Update” on page 215
- ♦ “Performing Optional Schema Enhancements” on page 216
- ♦ “Importing Remote Schema” on page 216
- ♦ “Declaring a New Schema Epoch” on page 217

Requesting Schema from the Tree

Use this operation to request the master replica of the root of the tree to synchronize its schema to this server. Any changes to the schema will be propagated to this server from the master replica of the [Root] for the next 24 hours.


IMPORTANT: If all servers request the schema from the master replica, network traffic can increase. Therefore, use this option with caution.

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance Utilities > Schema Maintenance.
- 3** Specify the server that will perform the operation, then click Next.
- 4** Specify a user name, password, and context for the server where you will perform the operation, then click Next.
- 5** Click Request Schema from Tree, then click Next.
- 6** Follow the online instructions to complete the operation.

Resetting the Local Schema

This operation invokes a schema reset which clears the time stamps on the local schema and requests an inbound schema synchronization.

This operation is unavailable if executed from the master replica of the [Root] partition. This is to ensure that not all servers in the tree reset at once.


- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance Utilities > Schema Maintenance.
- 3** Specify the server that will perform the operation, then click Next.
- 4** Specify a user name, password, and context for the server where you will perform the operation, then click Next.
- 5** Click Reset Local Schema, then click Next.
- 6** Follow the online instructions to complete the operation.

Performing a Post-NetWare 5 Schema Update

This operation extends and modifies the schema for compatibility with post-NetWare 5 DS changes.

Depending on your current eDirectory version, this option might be needed in order to update it to a newer version. Read the release notes for the new eDirectory version you'll be upgrading to in order to see if its use is necessary.

This operation requires that this server contain a replica of the [Root] partition (preferably the Master of [Root]) and that the state of the replica is On.

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance Utilities > Schema Maintenance.
- 3** Specify the server that will perform the operation, then click Next.
- 4** Specify a user name, password, and context for the server where you will perform the operation, then click Next.
- 5** Click Post NetWare 5 Schema Update, then click Next.
- 6** Follow the online instructions to complete the operation.


Performing Optional Schema Enhancements

This operation extends and modifies the schema for containment and other schema enhancements.

This operation requires that this server contain a replica of the [Root] partition and that the state of the replica must be On. In addition, all NetWare 4 servers in the tree must have the following DS.NLM versions:

Server	Version
4.10	ds.nlm v5.17 or later
4.11 / 4.2	ds.nlm v6.01 or later

Previous versions of eDirectory cannot synchronize these changes.


- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance Utilities > Schema Maintenance.
- 3** Specify the server that will perform the operation, then click Next.
- 4** Specify a username, password, and context for the server where you will perform the operation, then click Next.
- 5** Click Optional Schema Enhancements, then click Next.
- 6** Follow the online instructions to complete the operation.

Importing Remote Schema

This operation lets you select an eDirectory tree that contains the schema you want to add to the current tree's schema.

After you select a tree, the server that holds the master replica of the [Root] partition is contacted. The schema from that server is used to extend the schema on the current tree.

In order to merge two trees, you might need to import the schema from one tree to the other more than once. See [Chapter 8, "Merging Novell eDirectory Trees," on page 189](#) for more information.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance Utilities > Schema Maintenance.
- 3 Specify the server that will perform the operation, then click Next.
- 4 Specify a username, password, and context for the server where you will perform the operation, then click Next.
- 5 Click Import Remote Schema, then click Next.
- 6 Follow the online instructions to complete the operation.

Declaring a New Schema Epoch

An epoch is an instant in time that is arbitrarily selected as a point of reference. It is synonymous with era or new version. Epochs control the synchronization of replicas. When a new epoch is declared, it begins on the master replica. Other replicas cannot send updates to a replica with a newer epoch, but they receive updates from it until they become fully synchronized with it.


When other replicas of a given partition are synchronized with the updated replica, meaning that each replica's epoch is the same, bidirectional synchronization is allowed again.

When you declare a new schema epoch, the master replica of the [Root] partition is contacted and illegal time stamps are repaired on the schema records. A new epoch for the schema is then declared on that server, but it affects the entire tree.

All other servers receive a new copy of the schema including the repaired time stamps.

If the receiving server contains a schema that was not in the new epoch, objects and attributes that use the old schema are changed to the Unknown object class or attribute.

IMPORTANT: Do not perform this operation unless instructed to do so by Novell Support.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance Utilities > Schema Maintenance.
- 3 Specify the server that will perform the operation, then click Next.
- 4 Specify a user name, password, and context for the server where you will perform the operation, then click Next.
- 5 Click Declare a New Epoch, then click Next.
- 6 Follow the online instructions to complete the operation.

Repairing Server Network Addresses

The Server Repair Wizard lets you repair all server network addresses in replica rings and Server objects in the local database. You can also repair a selected server's network address in replica rings and Server objects in the local database.

Use the Server Repair Wizard to perform the following operations:


- ♦ [“Repairing All Network Addresses” on page 218](#)
- ♦ [“Repairing a Server's Network Addresses” on page 218](#)

Repairing All Network Addresses

This operation checks the network address for every server in the local eDirectory database. It searches the SAP tables, the SLP directory agent, and DNS local or remote information, depending on the transport protocol available, for each server's name.

Each address is then compared to the eDirectory Server object's Network Address attribute and the address record in each Replica attribute of every partition [Root] object. If the addresses are different, they are updated to be the same.


If the server address cannot be found in the SAP tables, local/remote DNS information, or SLP directory agents, no repair is performed.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance Utilities > Server Repair.
- 3 Specify the server that will perform the operation, then click Next.
- 4 Specify a username, password, and context for the server where you will perform the operation, then click Next.
- 5 Click Repair All Network Addresses, then click Next.
- 6 Follow the online instructions to complete the operation.

Repairing a Server's Network Addresses

This operation checks the network address for the selected server in the local eDirectory database files. It searches the local SAP tables, the SLP directory agent, or local or remote DNS information, depending on the transport protocols currently bound, for the server's name. The server's address is then compared to the eDirectory Server object's Network Address attribute and the address record in each Replica attribute of every partition [Root] object. If the addresses are different, they are updated to be the same.

If the server address cannot be found in the SAP tables, SLP, or local/remote DNS information, no repair is performed.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance Utilities > Server Repair.
- 3 Specify the server that will perform the operation, then click Next.
- 4 Specify a username, password, and context for the server where you will perform the operation, then click Next.
- 5 Click Repair This Server's Network Addresses, then click Next.
- 6 Follow the online instructions to complete the operation.

Issues

Novell SLP is an optional package. The authentication feature is not implemented as a part of the Novell SLP package.

eDirectory is now interoperable with OpenSLP, and the authentication features of OpenSLP are used.

Performing Synchronization Operations

The Sync Repair Wizard lets you synchronize a selected replica on the current server, report the synchronization status on the current server, report the synchronization status on all servers, perform a time synchronization, and schedule an immediate synchronization.

Use the Sync Repair Wizard to perform the following operations:


- ◆ “Synchronizing the Selected Replica on This Server” on page 219
- ◆ “Reporting the Synchronization Status on This Server” on page 219
- ◆ “Reporting the Synchronization Status on All Servers” on page 220
- ◆ “Performing a Time Synchronization” on page 220
- ◆ “Scheduling an Immediate Synchronization” on page 221

Synchronizing the Selected Replica on This Server

Use this operation to determine the complete synchronization status of every server that has a replica of the selected partition.

This helps you determine the health of a partition. If all of the servers with a replica of the partition are synchronizing properly, the partition is considered healthy. Each server in the replica ring is contacted, then each server performs an immediate synchronization to every other server in the replica ring.

Servers do not synchronize to themselves. Therefore, the status for the current server’s own replica is displayed as Host.


- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance Utilities > Sync Repair.
- 3** Specify the server that will perform the operation, then click Next.
- 4** Specify a username, password, and context for the server where you will perform the operation, then click Next.
- 5** Click Sync the Selected Replica on This Server, then click Next.
- 6** Follow the online instructions to complete the operation.

Reporting the Synchronization Status on This Server

This operation reports the replica synchronization status for every partition that has a replica on the current server.

This operation reads the Synchronization Status attribute from the replica [Root] object on each server that holds replicas of the partitions. It displays the time of the last successful synchronization to all servers and any errors that have occurred since the last synchronization.

It also displays a warning message if synchronization has not completed within 12 hours.

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance Utilities > Sync Repair.
- 3** Specify the server that will perform the operation, then click Next.


- 4 Specify a username, password, and context for the server where you will perform the operation, then click Next.
- 5 Click Report the Sync Status on This Server, then click Next.
- 6 Follow the online instructions to complete the operation.

Reporting the Synchronization Status on All Servers

Use this operation to determine the replica synchronization status for every partition that has a replica on the current server.

This operation reads the Synchronization Status attribute from the replica [Root] object on each server that holds replicas of the partitions. It displays the time of the last successful synchronization to all servers and any errors that have occurred since the last synchronization.

It also displays a warning message if synchronization has not completed within twelve hours.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance Utilities > Sync Repair.
- 3 Specify the server that will perform the operation, then click Next.
- 4 Specify a username, password, and context for the server where you will perform the operation, then click Next.
- 5 Click Report the Sync Status on All Servers, then click Next.
- 6 Follow the online instructions to complete the operation.

Performing a Time Synchronization

This operation contacts every server known to the local eDirectory database and requests information about each server's eDirectory and time synchronization status.

The version of eDirectory running on each server is reported in the DS version field.


The Replica Depth field reports a -1 if no replicas are stored on a given server. 0 is reported if the server contains a replica of the [Root] partition. A positive integer is reported if a replica exists on a given server and indicates how many objects away from [Root] the closest replica to [Root] is.

All servers in an eDirectory tree must be synchronized to the same time source. If all servers are not synchronized to the same time, object synchronization across replicas will not be managed correctly when collisions occur.

The Sync Repair Wizard cannot report the time source for each server, but it does reveal the time server type. This information can then be used to determine if time synchronization is configured properly.

IMPORTANT: You should use Novell iMonitor to monitor for the "Nearly-In-Sync" time synchronization status instead of using DSRepair. See [Chapter 7, "Using Novell iMonitor 2.1," on page 163](#) for more information.


For more information, see ["Synchronizing Network Time" on page 84](#).

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance Utilities > Sync Repair.
- 3 Specify the server that will perform the operation, then click Next.

- 4 Specify a username, password, and context for the server where you will perform the operation, then click Next.
- 5 Click Time Sync, then click Next.
- 6 Follow the online instructions to complete the operation.

Scheduling an Immediate Synchronization

This operation schedules a synchronization of all replicas to occur immediately. Use this operation if you want to review synchronization information without having to wait for the synchronization process to run as normally scheduled.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance Utilities > Sync Repair.
- 3 Specify the server that will perform the operation, then click Next.
- 4 Specify a username, password, and context for the server where you will perform the operation, then click Next.
- 5 Click Schedule Immediate Sync, then click Next.
- 6 Follow the online instructions to complete the operation.

Advanced DSRepair Options

In addition to the Repair features available in Novell iManager, the DSRepair utilities for each eDirectory platform contain some advanced features that are hidden from normal use. These advanced features are enabled through switches when loading the DSRepair utility on the various platforms.

- ♦ [“Running DSRepair on the eDirectory Server” on page 221](#)
- ♦ [“DSRepair Command Line Options” on page 222](#)
- ♦ [“Using Advanced DSRepair Switches” on page 224](#)

Running DSRepair on the eDirectory Server

NetWare

To run DSRepair, enter **dsrepair.nlm** at the server console.

To open DSRepair with advanced options, enter **dsrepair -a** at the server console.

Windows

- 1 Click Start > Settings > Control Panel > Novell eDirectory Services.
- 2 Click dsrepair.dlm, then click Start.

To open DSRepair with advanced options, enter **-a** in the Startup Parameters field in the Novell eDirectory Services Console before you start dsrepair.dlm.

Linux, Solaris, AIX, and HP-UX

To run DSRepair, enter **ndsrepair** at the server console, using the following syntax:

```
ndsrepair { -U | -E | -C | -P [-Ad] | -S [-Ad] | -N | -T | -J entry_id | --
version} [-F filename] [-A yes|no] [-O yes|no]
```

or

```
ndsrepair -R [-l yes|no] [-u yes|no] [-m yes|no] [-i yes|no] [-f yes|no] [-d
yes|no] [-t yes|no] [-o yes|no] [-r yes|no] [-v yes|no] [-c yes|no] [-F
filename] [-A yes|no] [-O yes|no]
```

IMPORTANT: The **-Ad** option should not be used without prior direction from Novell Support personnel.

Examples

To perform an unattended repair and log events in the `/root/ndsrepair.log` file, or to append events to the log file if it already exists, enter the following command:

```
ndsrepair -U -A no -F /root/ndsrepair.log
```

To open DSRepair with advanced options, enter the following command:

```
ndsrepair -Ad
```

To display a list of all global schema operations along with the advanced options, enter the following command:

```
ndsrepair -S -Ad
```

To repair the local database by forcing a database lock, enter the following command:

```
ndsrepair -R -l yes
```

NOTE: The input for the `ndsrepair` command can be redirected from an option file. The option file is a text file that can contain replica and partition operation-related options and suboptions that do not require authentication to the server. Each option or suboption is separated by a new line. Make sure that the contents of the file are in the proper sequence. If the contents are not in the proper sequence, the results will be unpredictable.

DSRepair Command Line Options

Option	Description
-U	Unattended Full Repair option. Instructs <code>ndsrepair</code> to run and exit without further user intervention. This is the suggested means of repair unless you are told by Novell Support to perform certain operations manually. You can view the log file after the repair has completed to determine what changes <code>ndsrepair</code> has made.
-P	Replica and Partition Operations option. Lists the partitions that have replicas stored in the current server's eDirectory database files. The Replica options menu provides options to repair replicas, cancel a partition operation, schedule synchronization, and designate the local replica as the master replica.
-S	Global Schema Operations option. Contains several schema operations that might be necessary to bring the server's schema into compliance with the master of the Tree object. However, these operations should be used only when necessary. The local and unattended repair operations already verify the schema.

Option	Description
-C	Check External Reference Object option. Checks each external reference object to determine if a replica containing the object can be located. If all servers that contain a replica of the partition with the object are inaccessible, the object is not found. If the object cannot be found, a warning is posted.
-E	Report Replica Synchronization option. Reports replica synchronization status for every partition that has a replica on the current server. This operation reads the synchronization status attribute from the replica's Tree object on each server that holds replicas of the partitions. It displays the time of the last successful synchronization to all servers and any errors that have occurred since the last synchronization. A warning message is displayed if synchronization has not completed within twelve hours.
-N	Servers Known to This Database option. Lists all servers known to the local eDirectory database. If your current server contains a replica of the Tree partition, this server displays a list of all servers in the eDirectory tree. Select one server to cause the server options to be executed.
-J	Repairs a single object on the local server. You need to provide the Entry ID (in hexadecimal format) of the object you want to repair. You can use this option instead of using the Unattended Repair (-U) option to repair one particular object that is corrupted. The Unattended Repair option can take many hours depending on the size of database. This option helps you save time.
-T	Time Synchronization option. Contacts every server known to the local eDirectory database and requests information about each server's time synchronization status. If this server contains a replica of the Tree partition, then every server in the eDirectory tree will be polled. The version of eDirectory that is running on each server is also reported.
-A	Append to the existing log file. The information is added to the existing log file. By default, this option is enabled.
-O	Logs the output in a file. By default, this option is enabled.
-F <i>filename</i>	Logs the output in the specified file.
-R	Repair the Local Database option. Repairs the local eDirectory database. Use the repair operation to resolve inconsistencies in the local database so that it can be opened and accessed by eDirectory. This option has suboptions that facilitate repair operations on the database. This option has function modifiers which are explained in the table below.

The function modifiers used with the -R option are described below:

Option	Description
-l	Locks the eDirectory database during the repair operation.
-u	Uses a temporary eDirectory database during the repair operation.
-m	Maintains the original unrepaired database.
-i	Checks the eDirectory database structure and the index.
-f	Reclaims the free space in the database.

Option	Description
-d	Rebuilds the entire database.
-t	Performs a tree structure check. Set it to Yes to check all the tree structure links for correct connectivity in the database. Set it to No to skip the check. Default =Yes.
-o	Rebuilds the operational schema.
-r	Repairs all the local replicas.
-v	Validates the stream files.
-c	Checks local references.

Using Advanced DSRepair Switches

WARNING: The features described in this section can cause irreversible damage to your eDirectory tree if they are used improperly. Use these features only if instructed to do so by Novell Support personnel.

You should make a full backup of eDirectory on the server before using any of these features in a production environment. See [Chapter 14, “Backing Up and Restoring Novell eDirectory,” on page 365](#) for more information.

On NetWare, use these options at the server console when loading DSRepair (for example, dsrepair -XK2).

On Linux, Solaris, AIX, and HP-UX, enter `ndsrepair -R -Ad -XK2`.

On Windows, enter these options in the Startup Parameters field in NDSConsole before you start dsrepair.dlm. See [“Running DSRepair on the eDirectory Server” on page 221](#) for more information.

Switch	Description
-NLC	If a NetWare server has the STORE NETWARE 5 CONN SCL MLA USAGE IN NDS set parameter turned on, the NLS:CERT PEAK USED POOL attribute could get a very high value. Running DSRepair with -NLC will clear these high values.
-P	Marks all eDirectory objects of type Unknown as referenced. Referenced objects do not participate in the eDirectory replica synchronization process.
-WM	In many cases, the WM: Registered Workstations attribute will become very high when using ZENworks® 2.0. Running DSRepair with -WM will clear these high values.
-XK2	Kills all eDirectory objects in this server's eDirectory database. This operation is used to destroy a corrupt replica that cannot be removed in any other way.
-XK3	Kills all external references in this server's eDirectory database. This operation is used to destroy all external references in a nonfunctioning replica. If the references are the source of the problem, eDirectory can then re-create the references in order to get the replica functioning again.

Using the eMBox Client to Repair a Database

The eDirectory Management Toolbox (eMBox) Client is a command line Java client that gives you remote access to DSRepair. Because the eMBox Client can be run in batch mode, you can use it to do unattended repairs using the eDirectory DSRepair eMTool.

The `emboxclient.jar` file is installed on your server as part of eDirectory. You can run it on any machine with a JVM. For more information on the eMBox Client, see [“Using the eMBox Command Line Client” on page 465](#).

Using the DSRepair eMTool

- 1 Run the eMBox Client in interactive mode by entering the following at the command line:

```
java -cp path_to_the_file/emboxclient.jar embox -i
```

(If you have already put the `emboxclient.jar` file in your class path, you only need to enter `java embox -i`.)

The eMBox Client prompt appears:

```
eMBox Client>
```

- 2 Log in to the server you want to repair by entering the following:

```
login -sserver_name_or_IP_address -pport_number
      -uusername.context -wpassword -n
```

The port number is usually 80 or 8008, unless you have a Web server that is already using the port. The `-n` option opens a nonsecure connection.

The eMBox Client will indicate whether the login is successful.

- 3 Enter a repair command, using the following syntax:

```
dsrepair.task options
```

For example:

`dsrepair.ufr` performs an unattended full repair.

`dsrepair.rld -a -v` repairs the local database using the Repair All Local Replicas and Check Local References options.

A space must be between each switch. The order of the switches is not important.

The eMBox Client will indicate whether the repair is successful.

See [“DSRepair eMTool Options” on page 226](#) for more information on the DSRepair eMTool options.

- 4 Log out from the eMBox Client by entering the following command:

```
logout
```

- 5 Exit the eMBox Client by entering the following command:

```
exit
```

DSRepair eMTool Options

The following tables lists the DSRepair eMTool options. You can also use the **list -tdsrepair** command in the eMBox Client to list the DSRepair options with details. See [“Listing eMTools and Their Services” on page 469](#) for more information.

Option	Description
rso	Single object repair
-o	Object ID in hex
-d	Object DN
rts	Time synchronization
rss	Report synchronization status of all partitions
rld	Repair local database
-l	Lock eDirectory database during entire repair
-t	Use temporary eDirectory database during repair
-d	Maintain original unrepaired database
-p	Perform database structure check
-i	Perform database structure and index check
-f	Reclaim database free space
-e	Rebuild the entire database
-c	Perform tree structure check
-o	Rebuild operational schema
-a	Repair all local replicas
-m	Validate mail directories and stream files
-v	Check local references
ufr	Unattended full repair
rsn	Repair selected server's network address
-o	Object ID in hex
-d	Object DN
ran	Repair all network addresses
rsr	Repair selected replica
-p	Partition ID
-d	Partition DN
rer	Repair every replica
ror	Repair selected replica ring
-p	Partition ID
-d	Partition DN
rar	Repair replica ring, all replicas
ssa	Report the replica synchronization status of all servers
-p	Partition ID
-d	Partition DN
cer	Check external references

Option	Description
rao	Receive all objects for this replica
-p	Partition ID
-d	Partition DN
-s	Server ID
-d	Server DN
sao	Send all objects to every replica in the ring
-p	Partition ID
-d	Partition DN
-s	Server ID
-d	Server DN
dne	Repair time stamps and declare a new epoch
-p	Partition ID
-d	Partition DN
sri	Schedule immediate synchronization
-p	Partition ID
-d	Partition DN
	Server ID
	Server DN
sks	Synchronize the replica on the selected server
-p	Partition ID
-d	Partition DN
-s	Server ID
-d	Server DN
ske	Synchronize the replica on all servers
-p	Partition ID
-d	Partition DN
dsr	Destroy the selected replica on this server
-p	Partition ID
-d	Partition DN
xsr	Remove this server from the replica ring
-p	Partition ID
-d	Partition DN
-s	Server ID
-d	Server DN
dnm	Designate this server as the new master replica
-p	Partition ID
-d	Partition DN
dul	Delete unknown leaf objects

10 WAN Traffic Manager

WAN Traffic Manager (WTM) lets you manage replication traffic across WAN links, reducing network costs. WAN Traffic Manager is installed during the Novell® eDirectory™ installation and consists of the following elements:

- ♦ WTM

This resides on each server in the replica ring. Before eDirectory sends server-to-server traffic, WTM reads a WAN traffic policy and determines whether the traffic will be sent.

- ♦ WAN traffic policies

These rules control the generation of eDirectory traffic. WAN traffic policies are text stored as an eDirectory property value on a Server object, a LAN Area object, or both.

- ♦ WANMAN Novell iManager plug-in

This interface to WTM lets you create or modify policies, create LAN Area objects, and apply policies to LAN areas or servers. When WTM is installed (as part of the eDirectory installation), the schema includes a LAN Area object and a WAN Traffic Manager page on the Server object.

WAN Traffic Manager (wtm.nlm on NetWare® or wtm.dlm on Windows) must reside on each server whose traffic you want to control. If a partition's replica ring includes servers on both sides of a wide area link, you should install WAN Traffic Manager on all servers in that replica ring.

IMPORTANT: WAN Traffic Manager is not supported on Linux, Solaris, AIX, or HP-UX systems.

Understanding WAN Traffic Manager

Network directories, such as eDirectory, create server-to-server traffic. If this traffic crosses wide area network (WAN) links unmanaged, it can needlessly increase costs and overload slow WAN links during high-usage periods.

WAN Traffic Manager lets you control server-to-server traffic (over WAN links) generated by eDirectory and control eDirectory traffic between any servers in an eDirectory tree. WTM can restrict traffic based on cost of traffic, time of day, type of eDirectory operations, or any combination of these.

For example, you might restrict eDirectory traffic over a WAN link during high-usage times. This shifts high-bandwidth activities to off-hours. You might also limit replica synchronization traffic to times when rates are low to reduce costs.

WAN Traffic Manager controls only periodic events initiated by eDirectory, such as replica synchronization. It does not control events initiated by administrators or users, nor does it control non-eDirectory server-to-server traffic such as time synchronization.

The eDirectory processes listed in the following table generate server-to-server traffic.

Process	Description
Replica synchronization	<p>Ensures that changes to eDirectory objects are synchronized among all replicas of the partition. This means that any server that holds a copy of a given partition must communicate with the other servers to synchronize a change.</p> <p>Two types of replica synchronization can occur:</p> <ul style="list-style-type: none"> ◆ Immediate sync occurs after any change to an eDirectory object or any addition or deletion of an object in the directory tree. ◆ Slow sync occurs for specific changes to an eDirectory object that are repetitive and common to multiple objects, such as changes to login properties. Some examples of this are updates to Login Time, Last Login Time, Network Address, and Revision properties when a user logs in or out. <p>The slow sync process runs only in the absence of an immediate sync process. By default, immediate sync runs ten seconds after any change is saved and slow sync runs 22 minutes after other changes are made.</p>
Schema synchronization	<p>Ensures that the schema is consistent across the partitions in the directory tree and that all schema changes are updated across the network.</p> <p>This process runs once every four hours by default.</p>
Heartbeat	<p>Ensures that directory objects are consistent among all replicas of a partition. This means that any server with a copy of a partition must communicate with the other servers holding the partition to check the consistency.</p> <p>This process runs by default once every 30 minutes on every server that contains a replica of a partition.</p>
Limber	<p>Ensures that a server's replica pointer table is updated when that server's name or address is changed. Such changes occur when</p> <ul style="list-style-type: none"> ◆ The server is rebooted with a new server name or IPX™ internal address in the autoexec.ncf file. ◆ An address is added for an additional protocol. <p>When a server is booted, the limber process compares the server's name and IPX address with those stored in the replica pointer table. If either is different, eDirectory automatically updates all replica pointer tables that contain a listing of that server.</p> <p>The limber process also checks that the tree name is correct for each server in a replica ring.</p> <p>Limber runs five minutes after the server boots up and then every three hours.</p>
Backlink	<p>Verifies external references, which are pointers to eDirectory objects that are not stored in the replicas on a server. The backlink process normally runs two hours after the local database is opened and then every 13 hours thereafter.</p>


Process	Description
Connection management	<p>Servers in a replica ring require a highly secure connection for transferring NCP™ packets. These secure connections, called virtual client connections, are established by the connection management process.</p> <p>The connection management process might also need to establish a virtual client connection for schema synchronization or backlink processes. Time synchronization might also require such a connection, depending on the configuration of time services.</p>
Server status check	<p>Each server without a replica initiates a server status check. It establishes a connection to the nearest server that holds a writable replica of the partition containing the Server object.</p> <p>The server status check runs every six minutes.</p>

LAN Area Objects

A LAN Area object lets you easily administer WAN traffic policies for a group of servers. After you create a LAN Area object, you can add servers to or remove servers from the LAN Area object. When you apply a policy to the LAN Area, that policy applies to all the servers in the LAN Area.

You should create a LAN Area object if you have multiple servers in a LAN that is connected to other LANs by wide area links. If you do not create a LAN Area object, you must manage each server's WAN traffic individually.

Creating a LAN Area Object



- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click WAN Traffic > Create LAN Area.
- 3 Select WANMAN-LAN Area from the Object Class drop-down list.
- 4 Specify a name and context for the object, then click Create.

Continue with one of the following sections:

- ♦ [“Adding Servers to a LAN Area Object” on page 231](#)
- ♦ [“Applying WAN Policies” on page 233](#)

Adding Servers to a LAN Area Object

A server can belong to only one LAN Area object. If the server you are adding already belongs to a LAN Area object, the server is removed from that object and added to the new object.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click WAN Traffic > WAN Traffic Manager Overview.
- 3 Click View LAN Areas, then click the LAN Area object you want.
- 4 Click Server List, then click the Object Selector button .
- 5 Select the server you want.
- 6 Repeat [Step 4](#) through [Step 5](#) for each server you want to add.

To apply a WAN policy to the LAN Area object, thereby applying the policy to all the servers in the group, see [“Applying WAN Policies” on page 233](#).

- 7 Click Apply, then click OK.

Adding Additional Information to a LAN Area Object

You can use ConsoleOne® to add descriptive information to a LAN Area object. This feature is not available in Novell iMonitor.

- 1 In ConsoleOne, right-click a LAN Area object.
- 2 Click Properties > General.
- 3 Add the Owner, Description, Location, Department, and Organization information you want.
- 4 Click Apply, then click OK.

WAN Traffic Policies

A WAN traffic policy is a set of rules that control the generation of eDirectory traffic. These rules are created as text and are stored as an eDirectory property value on the Server object, the LAN Area object, or both. The policy is interpreted according to a simple processing language.

You can apply policies to individual servers or you can create LAN Area objects and assign several servers to one of these objects. Any policy that is applied to the LAN Area object is automatically applied to all servers that are assigned to the object.

WAN Traffic Manager comes with several predefined policy groups. You can use these policies as they are, modify them to meet your needs, or write new policies.

- ◆ [“Applying WAN Policies” on page 233](#)
- ◆ [“Modifying WAN Policies” on page 233](#)
- ◆ [“Renaming an Existing Policy” on page 234](#)
- ◆ [“Creating New WAN Policies” on page 235](#)

Predefined Policy Groups

The following table lists groups of predefined policies with similar functions:

Policy Group	Description
1-3am.wmg	Limits the time traffic is sent to between 1 a.m. and 3 a.m.
7am-6pm.wmg	Limits the time traffic is sent to between 7 a.m. and 6 p.m.
costlt20.wmg	Allows only traffic that has a cost factor below 20 to be sent.
ipx.wmg	Allows only IPX traffic.
ndsttys.wmg	Provides sample policies for various eDirectory traffic types.
onospoof.wmg	Allows only existing WAN connections to be used.
opns spoof.wmg	Allows only existing WAN connections to be used but assumes that a connection that hasn't been used for 15 minutes is being spoofed and should not be used.
samearea.wmg	Allows traffic only in the same network area.
tcpip.wmg	Allows only TCP/IP traffic.


Policy Group	Description
timecost.wmg	Restricts all traffic to between 1 a.m. and 1:30 a.m. but allows servers in the same location to talk continuously.

For detailed information on the predefined policy groups and their individual policies, see [“WAN Traffic Manager Policy Groups” on page 238](#).

Applying WAN Policies

You can apply WAN policies to an individual server or to a LAN Area object. Policies applied to an individual server manage eDirectory traffic for that server only. Policies applied to a LAN Area object manage traffic for all servers that belong to the object.


WAN Traffic Manager looks in wanman.ini for a WAN policy groups section, which contains a *key = values* statement. *Key* is the policy name displayed in the snap-in and *value* is the path to the text files containing delimited policies.

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click WAN Traffic > WAN Traffic Manager Overview.
- 3** Click View LAN Areas, then click a LAN Area object.
or
Click View NCP Server, then click an NCP Server object.
- 4** Click Add Policy, then select the policy group you want.
See [“Predefined Policy Groups” on page 232](#) for more information.
- 5** Click OK.
A list of the policies loaded from the policy group is displayed.
- 6** Click OK.
You can read what the policy does, make changes to the policy, or click Check Policy to check for errors in the policy.
- 7** To remove a policy that you don't want, select the policy from the Policy Name drop-down list, then click Delete Policy.
- 8** Click Apply, then click OK.

Modifying WAN Policies

You can modify any of the predefined policy groups included with WAN Traffic Manager to meet your own needs. You can also modify a policy you wrote yourself.

Modifying WAN Policies Applied to a Server


- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click WAN Traffic > WAN Traffic Manager Overview > View NCP Servers.
- 3** Click the Server object that contains the policy you want to edit.
- 4** Select the policy you want to edit from the Policy Name drop-down list.
- 5** In the Policy field, edit the policy to meet your needs.

To understand the structure of a WAN policy, see [“WAN Policy Structure” on page 251](#).


To understand the syntax of a WAN policy, see [“Construction Used within Policy Sections” on page 254](#).

- 6** Click Check Policy to identify errors in syntax or structure.
WAN Traffic Manager will not run policies with errors.
- 7** Click Apply if you made any changes.
- 8** To remove a policy that you don't want, select the policy from the Policy Name drop-down list, then click Delete Policy.
- 9** Click Apply, then click OK.

Modifying WAN Policies Applied to a LAN Area Object

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click WAN Traffic > WAN Traffic Manager Overview > View LAN Areas.
- 3** Click the LAN Area object that contains the policy you want to edit.
- 4** Select the policy you want to edit from the Policy Name drop-down list.
- 5** In the Policy field, edit the policy to meet your needs.
To understand the structure of a WAN policy, see [“WAN Policy Structure” on page 251](#).
To understand the syntax of a WAN policy, see [“Construction Used within Policy Sections” on page 254](#).
- 6** Click Check Policy to identify errors in syntax or structure.
WAN Traffic Manager will not run policies with errors.
- 7** Click Apply if you made any changes.
- 8** To remove a policy that you don't want, select the policy from the Policy Name drop-down list, then click Delete Policy.
- 9** Click Apply, then click OK.


Renaming an Existing Policy

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click WAN Traffic > WAN Traffic Manager Overview.
- 3** Click View LAN Areas, then click a LAN Area object.
or
Click View NCP Server, then click an NCP Server object.
- 4** Select the policy you want to rename from the Policy Name drop-down list.
- 5** Click Rename Policy, then specify the new name.
The name must be a fully distinguished name.
- 6** Click OK, click Apply, then click OK.


Creating New WAN Policies

You can write a WAN policy for a Server object or a LAN Area object. Policies written for an individual server manage eDirectory traffic for that server only, while policies written for a LAN Area object manage traffic for all servers that belong to the object.

Creating a WAN Policy for a Server Object

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click WAN Traffic > WAN Traffic Manager Overview > View NCP Servers.
- 3** Click the Server object you want to create a new policy for, then click Create Policy.
- 4** Specify a name for the new policy, then click OK.
The name you provide should be a fully distinguished name.
- 5** Specify the necessary information in the Policy text box.
To understand the structure of a WAN policy, see [“WAN Policy Structure” on page 251](#).
To understand the syntax of a WAN policy, see [“Construction Used within Policy Sections” on page 254](#).
- 6** Click Apply, then click OK.


Creating a WAN Policy for a LAN Area Object

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click WAN Traffic > WAN Traffic Manager Overview > View LAN Areas.
- 3** Click the LAN Area object you want to create a WAN policy for, then click Create Policy.
- 4** Specify a name for the new policy, then click OK.
The name you provide should be a fully distinguished name.
- 5** Specify the necessary information in the Policy text box.
To understand the structure of a WAN policy, see [“WAN Policy Structure” on page 251](#).
To understand the syntax of a WAN policy, see [“Construction Used within Policy Sections” on page 254](#).
- 6** Click Apply, then click OK.

Limiting WAN Traffic

WAN Traffic Manager comes with two predefined WAN Policy groups that limit traffic to specific hours. (For more information, see [“1-3am.wmg” on page 238](#) and [“7am-6pm.wmg” on page 238](#).) You can modify these policies to limit traffic to any span of hours you select.

The instructions below are for modifying the 1:00 a.m. to 3:00 a.m. group, but you can use the same steps to accomplish the same thing with the 7:00 a.m. to 6:00 p.m. group.

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click WAN Traffic > WAN Traffic Manager Overview.
- 3** Click View LAN Areas, then click a LAN Area object.
or
Click View NCP Server, then click an NCP Server object.

- 4** Click Add Policy.
- 5** Select 1-3am.wmg from the list of predefined policies, then click OK twice.

The policy is displayed in the Policy text box, which lets you make changes. For example, if you want to limit traffic to 2:00 a.m. to 5:00 p.m. rather than from 1:00 a.m. to 3:00 a.m., make the following changes:

```
/* This policy limits all traffic to between 2 and 5 pm */
LOCAL BOOLEAN Selected;
SELECTOR
  Selected := Now.hour >= 2 AND Now.hour < 17;
  IF Selected THEN
    RETURN 50; /* between 2am and 5pm this policy has a
high priority */
  ELSE
    RETURN 1; /* return 1 instead of 0 in case there are
no other policies */
    /* if no policies return > 0, WanMan assumes
SEND */
  END
END
PROVIDER
  IF Selected THEN
    RETURN SEND; /* between 2am and 5pm, SEND */
  ELSE
    RETURN DONT_SEND; /* other times, don't */
  END
END
```

In the comment lines (set off with `/*` and `*/`), the hour can be designated using a.m. and p.m. In the active code, however, it must be designated using 24-hour format. In that case, 5:00 p.m. becomes 17.

To better understand the structure of a WAN policy, see [“WAN Policy Structure” on page 251](#).

To better understand the syntax of a WAN policy, see [“Construction Used within Policy Sections” on page 254](#).

- 6** After modifying the syntax of the policy, click Check Policy to identify errors in syntax or structure.

The results of the policy check are displayed.

WAN Traffic Manager will not run policies with errors.
- 7** If you want to keep the original 1-3 am policy, add the new policy under a different name.
 - 7a** Click Rename Policy.
 - 7b** Enter a name for the edited policy, then click OK.
- 8** Click Apply, then click OK.

Assigning Cost Factors

Cost factors let WAN Traffic Manager compare the cost of traffic with certain destinations, then manage the traffic using WAN policies. WAN policies use cost factors to determine the relative expense of WAN traffic. You can then use this information in determining whether to send traffic.

A cost factor is expressed as expense per unit of time. It can be in any units as long as the same units are used consistently in each WAN traffic policy. You can use dollars per hour, cents per

minute, yen per second, or any other ratio of expense to time, as long as you use that ratio exclusively.


You can assign destination cost factors representing the relative expense of traffic to particular address ranges. Therefore, you can assign cost for an entire group of servers in one declaration. You can also assign a default cost factor to be used when no cost is specified for a destination.

If no cost is assigned for the destination, the default cost is used. If you have specified no default cost for the server or LAN Area object, a value of -1 is assigned.

For information about a sample policy that restricts traffic based on cost factor, see [“Costlt20.wmg” on page 238](#).

For information about how to modify a policy, see [“Modifying WAN Policies” on page 233](#).



Assigning Default Cost Factors

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click WAN Traffic Management > WAN Traffic Manager Overview.
- 3** Click View LAN Areas, then click a LAN Area object.
or
Click View NCP Server, then click an NCP Server object.
- 4** Click Costs, then specify a cost in the Default Cost field.

The cost must be a nonnegative integer. If supplied, the default cost will be assigned to all destinations in the Server or LAN Area object that do not fall within a destination address range with an assigned cost. For example, you might specify the cost in monetary units, such as dollars, or in packets per second.

- 5** Click Apply, then click OK.

Assigning a Cost to a Destination Address Range

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click WAN Traffic Management > WAN Traffic Manager Overview.
- 3** Click View LAN Areas, then click a LAN Area object.
or
Click View NCP Server, then click an NCP Server object.
- 4** Click Costs.
- 5** Click the Add button .
- 6** In the Create Wanman Cost window, select TCP/IP Address Type or IPX Address Type.
- 7** Specify the start address and stop address of the range, in the appropriate format for TCP/IP or IPX.
- 8** In the Cost text field, specify the cost as a nonnegative integer.
- 9** Click OK, click Apply, then click OK.

WAN Traffic Manager Policy Groups

WAN Traffic Manager comes with the following predefined policy groups.

For more information on applying policy groups, see [“Applying WAN Policies” on page 233](#).

1-3am.wmg

The policies in this group limit the time traffic can be sent to between 1 a.m. and 3 a.m. There are two policies:

- ◆ 1 - 3 am, NA

Limits the checking of backlinks, external references, and login restrictions; the running of Janitor or Limber; and schema synchronization to these hours.

- ◆ 1 - 3 am

Limits all other traffic to these hours.

To restrict all traffic to these hours, both policies must be applied.

7am-6pm.wmg

The policies in this group limit the time traffic can be sent to between 7 a.m. and 6 p.m. There are two policies:

- ◆ 7 am - 6 pm, NA

Limits the checking of backlinks, external references, and login restrictions; the running of Janitor or Limber; and schema synchronization to these hours.

- ◆ 7 am - 6 pm

Limits all other traffic to these hours.

To restrict all traffic to these hours, both policies must be applied.

Costlt20.wmg

The policies in this group allow only traffic that has a cost factor below 20 to be sent. There are two policies:

- ◆ Cost < 20, NA

Prevents the checking of backlinks, external references, and login restrictions; the running of Janitor or Limber; and schema synchronization unless the cost factor is less than 20.

- ◆ Cost < 20

Prevents all other traffic unless the cost factor is less than 20.

To prevent all traffic with a cost factor of 20 or greater, both policies must be applied.

lpx.wmg

The policies in this group allow only IPX traffic. There are two policies:

- ◆ IPX, NA

Prevents the checking of backlinks, external references, and login restrictions; the running of Janitor or Limber; and schema synchronization unless the traffic that is generated is IPX.

- ◆ IPX

Prevents all other traffic unless the traffic is IPX.

To prevent all non-IPX traffic, both policies must be applied.

Ndsttyps.wmg

The policies in this group are sample policies for various eDirectory traffic types. They contain the variables eDirectory passes in a request of this type.

- ◆ “Sample Catch All with Addresses” on page 239
- ◆ “Sample Catch All without Addresses” on page 239
- ◆ “Sample NDS_BACKLINK_OPEN” on page 239
- ◆ “Sample NDS_BACKLINKS” on page 240
- ◆ “Sample NDS_CHECK_LOGIN_RESTRICTION” on page 241
- ◆ “Sample NDS_CHECK_LOGIN_RESTRICTION_OPEN” on page 243
- ◆ “Sample NDS_JANITOR” on page 243
- ◆ “Sample NDS_JANITOR_OPEN” on page 245
- ◆ “Sample NDS_LIMBER” on page 246
- ◆ “Sample NDS_LIMBER_OPEN” on page 247
- ◆ “Sample NDS_SCHEMA_SYNC” on page 248
- ◆ “Sample NDS_SCHEMA_SYNC_OPEN” on page 248
- ◆ “Sample NDS_SYNC” on page 249

Sample Catch All with Addresses

A sample policy for traffic types with addresses.

Sample Catch All without Addresses

A sample policy for traffic types without addresses.

Sample NDS_BACKLINK_OPEN

NDS_BACKLINK_OPEN is a traffic type that is used if either CheckEachNewOpenConnection or CheckEachAlreadyOpenConnection was set to 1 during the corresponding NDS_BACKLINKS query.

This query is generated whenever CheckEachNewOpenConnection is 1 and eDirectory needs to open a new connection for backlinking or when CheckEachAlreadyOpenConnection is 1 and eDirectory needs to reuse an already existing connection.

- ◆ Version (Input Only, Type INTEGER)
The version of eDirectory.
- ◆ ExpirationInterval (Input and Output, Type INTEGER)

If `ConnectionIsAlreadyOpen` is TRUE, `ExpirationInterval` is set to the expiration interval already set on the existing connection. Otherwise, it is set to the `ExpirationInterval` assigned in the `NDS_BACKLINKS` query. A 0 value indicates that the default (2 hours) should be used. On exit, the value of this variable is assigned as the expiration interval for the connection.

Value	Description
<0, 0	Use the default expiration interval (default).
>0	Expiration interval to be assigned to this connection.

◆ `ConnectionIsAlreadyOpen` (Input Only, Type BOOLEAN)

This variable is TRUE if eDirectory can reuse an existing connection and FALSE if it needs to create a new connection.

Value	Description
TRUE	eDirectory determines that it already has a connection to this address and can reuse that connection.
FALSE	eDirectory does not have a connection to this address and must create one.

◆ `ConnectionLastUsed` (Input Only, Type TIME)

If `ConnectionIsAlreadyOpen` is TRUE, then `ConnectionLastUsed` is the last time that a packet was sent from eDirectory using this connection. Otherwise, it is 0.

Value	Description
TRUE	<i>ConnectionLastUsed</i> is the time that eDirectory last sent a packet on this connection.
FALSE	<i>ConnectionLastUsed</i> will be 0.

Sample NDS_BACKLINKS

Before eDirectory checks any backlinks or external references, it queries WAN Traffic Manager to see if this is an acceptable time for this activity. `NDS_BACKLINKS` does not have a destination address; it requires a `NO_ADDRESSES` policy. If WAN Traffic Manager returns `DONT_SEND`, backlink checking will be put off and rescheduled. The following variables are supplied:

◆ `Last` (Input Only, Type TIME)

The time of the last round of backlink checking since eDirectory started. When eDirectory starts, *Last* is initialized to 0. If `NDS_BACKLINKS` returns `SEND`, *Last* is set to the current time after eDirectory finishes backlinking.

◆ `Version` (Input Only, Type INTEGER)

The version of eDirectory.

◆ `ExpirationInterval` (Output Only, Type INTEGER)

The expiration interval for all connections created while backlinking.

Value	Description
<0, 0	Use the default expiration interval (default).
>0	Expiration interval to be assigned to this connection.

- ◆ Next (Output Only, Type TIME)

Tells eDirectory when to schedule the next round of backlink checking.

Value	Description
In past, 0	Use the default scheduling.
In future	Time when backlinking should be scheduled.

- ◆ CheckEachNewOpenConnection (Output Only, Type INTEGER)

Tells eDirectory what to do if it needs to create a new connection while doing backlinking.

CheckEachNewOpenConnection is initialized to 0.

Value	Description
0	Return Success without calling WAN Traffic Manager, allowing the connection to proceed normally (default).
1	Call WAN Traffic Manager and let the policies decide whether to allow the connection.
2	Return ERR_CONNECTION_DENIED without calling WAN Traffic Manager, causing the connection to fail.

- ◆ CheckEachAlreadyOpenConnection (Output Only, Type INTEGER)

This variable tells eDirectory what to do if it needs to reuse a connection it believes is already open while doing backlinking. CheckEachAlreadyOpenConnection is initialized to 0.

Value	Description
0	Return Success without calling WAN Traffic Manager, allowing the connection to proceed normally (default).
1	Call WAN Traffic Manager and let the policies decide whether to allow the connection.
2	Return ERR_CONNECTION_DENIED without calling WAN Traffic Manager, causing the connection to fail.

Sample NDS_CHECK_LOGIN_RESTRICTION

Before eDirectory checks a login restriction, it queries WAN Traffic Manager to see if this is an acceptable time for this activity. The traffic type NDS_CHECK_LOGIN_RESTRICTIONS does not have a destination address; it requires a NO_ADDRESSES policy. If WAN Traffic Manager returns DONT_SEND, the check errors out.

The following variables are provided:

- ◆ Version (Input Only, Type INTEGER)

The version of eDirectory.

- ◆ Result (Output Only, Type INTEGER)

If the result of NDS_CHECK_LOGIN_RESTRICTIONS is DONT_SEND, then the following values are returned to the operating system.

Value	Description
0	Login is allowed.
1	Login is not allowed during the current time block.
2	Account is disabled or expired.
3	Account has been deleted.

- ◆ ExpirationInterval (Output Only, Type INTEGER)

The expiration interval that should be assigned to this connection.

Value	Description
<0, 0	Use the default expiration interval (default).
>0	Expiration interval to be assigned to this connection.

- ◆ CheckEachNewOpenConnection (Output Only, Type INTEGER)

Value	Description
0	Return Success without calling WAN Traffic Manager, allowing the connection to proceed normally (default).
1	Call WAN Traffic Manager and let the policies decide whether to allow the connection.
2	Return ERR_CONNECTION_DENIED without calling WAN Traffic Manager, causing the connection to fail.

- ◆ CheckEachAlreadyOpenConnection (Output Only, Type INTEGER)

Value	Description
0	Return Success without calling WAN Traffic Manager, allowing the connection to proceed normally (default).
1	Call WAN Traffic Manager and let the policies decide whether to allow the connection.
2	Return ERR_CONNECTION_DENIED without calling WAN Traffic Manager, causing the connection to fail.

Sample NDS_CHECK_LOGIN_RESTRICTION_OPEN

NDS_CHECK_LOGIN_RESTRICTION_OPEN is only used if either CheckEachNewOpenConnection or CheckEachAlreadyOpenConnection was set to 1 during the corresponding NDS_CHECK_LOGIN_RESTRICTIONS query. This query is generated whenever CheckEachNewOpenConnection is 1 and eDirectory needs to

- ◆ Open a new connection before running Limber.
- ◆ Open a new connection before checking the login restriction.
- ◆ Reuse an already existing connection.

The following variables are provided:

- ◆ Version (Input Only, Type INTEGER)

The version of eDirectory.

- ◆ ExpirationInterval (Input and Output, Type INTEGER)

Value	Description
<0, 0	Use the default expiration interval (default).
>0	Expiration interval to be assigned to this connection.

- ◆ ConnectionIsAlreadyOpen (Input Only, Type BOOLEAN)

Value	Description
TRUE	eDirectory determines that it already has a connection to this address and can reuse that connection.
FALSE	eDirectory does not have a connection to this address and must create one.

- ◆ ConnectionLastUsed (Input Only, Type TIME)

If ConnectionIsAlreadyOpen is TRUE, then ConnectionLastUsed is the last time that a packet was sent from eDirectory using this connection. Otherwise, it will be 0.

Value	Description
TRUE	<i>ConnectionLastUsed</i> is the time that eDirectory last sent a packet on this connection.
FALSE	<i>ConnectionLastUsed</i> will be 0.

Sample NDS_JANITOR

Before eDirectory runs the janitor, it queries WAN Traffic Manager to see if this is an acceptable time for this activity. The NDS_JANITOR does not have a destination address; it requires a NO_ADDRESSES policy. If WAN Traffic Manager returns DONT_SEND, janitor work is put off and rescheduled.

The following variables are provided:

- ◆ Last (Input Only, Type TIME)

The time of the last round of janitor work since eDirectory started. When eDirectory starts, *Last* is initialized to 0. If NDS_JANITOR returns SEND, *Last* is set to the current time after eDirectory finishes the janitor.

- ◆ Version (Input Only, Type INTEGER)

The version of eDirectory.

- ◆ ExpirationInterval (Output Only, Type INTEGER)

The expiration interval for all connections created while running the Janitor.

Value	Description
<0, 0	Use the default expiration interval (default).
>0	Expiration interval to be assigned to this connection.

- ◆ Next (Output Only, Type TIME)

Tells eDirectory when to schedule the next round of Janitor work.

Value	Description
In the past, 0	Use the default scheduling.
In the future	Time when the janitor should be scheduled.

- ◆ CheckEachNewOpenConnection (Output Only, Type INTEGER)

Tells eDirectory what to do if it needs to create a new connection while running the janitor.

CheckEachNewOpenConnection is initialized to 0.

Value	Description
0	Return Success without calling WAN Traffic Manager, allowing the connection to proceed normally (default).
1	Call WAN Traffic Manager and let the policies decide whether to allow the connection.
2	Return ERR_CONNECTION_DENIED without calling WAN Traffic Manager, causing the connection to fail.

- ◆ CheckEachAlreadyOpenConnection (Output Only, Type INTEGER)

Tells eDirectory what to do if it needs to reuse a connection it determines is already open while running the Janitor.

CheckEachAlreadyOpenConnection is initialized to 0.

Value	Description
0	Return Success without calling WAN Traffic Manager, allowing the connection to proceed normally (default).

Value	Description
1	Call WAN Traffic Manager and let the policies decide whether to allow the connection.
2	Return ERR_CONNECTION_DENIED without calling WAN Traffic Manager, causing the connection to fail.

Sample NDS_JANITOR_OPEN

NDS_JANITOR_OPEN is used only if either CheckEachNewOpenConnection or CheckEachAlreadyOpenConnection was set to 1 during the corresponding NDS_JANITOR query. This query is generated whenever CheckEachNewOpenConnection is 1 and eDirectory needs to open a new connection before doing backlinking, or when CheckEachAlreadyOpenConnection is 1 and eDirectory needs to reuse an already existing connection.

The following variables are provided:

- ◆ Version (Input Only, Type INTEGER)

The version of eDirectory.

- ◆ ExpirationInterval (Input and Output, INTEGER)

If ConnectionIsAlreadyOpen is TRUE, ExpirationInterval is set to the expiration interval already set on the existing connection. Otherwise, it is set to the ExpirationInterval assigned in the NDS_JANITOR query. A 0 value indicates that the default (2 hours, 10 seconds) should be used. On exit, the value of this variable is assigned as the expiration interval for the connection.

Value	Description
<0, 0	Use the default expiration interval (default).
>0	Expiration interval to be assigned to this connection.

- ◆ ConnectionIsAlreadyOpen (Input Only, Type BOOLEAN)

This variable is TRUE if eDirectory needs to reuse an existing connection and FALSE if it needs to create a new connection.

Value	Description
TRUE	eDirectory determines that it already has a connection to this address and can reuse that connection.
FALSE	eDirectory does not have a connection to this address and must create one.

- ◆ ConnectionLastUsed (Input Only, Type TIME)

If ConnectionIsAlreadyOpen is TRUE, then ConnectionLastUsed is the last time that a packet was sent from eDirectory using this connection. Otherwise, it will be 0.

Value	Description
TRUE	ConnectionLastUsed is the time that eDirectory last sent a packet on this connection.
FALSE	ConnectionLastUsed will be 0.

Sample NDS_LIMBER

Before eDirectory runs limber, it queries WAN Traffic Manager to see if this is an acceptable time for this activity. The traffic type NDS_LIMBER does not have a destination address; it requires a NO_ADDRESSES policy. If WAN Traffic Manager returns DONT_SEND, limber is put off and rescheduled.

The following variables are provided:

- ◆ Last (Input Only, Type TIME)
The time of last limber since eDirectory started.
- ◆ Version (Input Only, Type INTEGER)
The version of eDirectory.
- ◆ ExpirationInterval (Output Only, Type INTEGER)
The expiration interval for all connections created while running limber checks.

Value	Description
<0, 0	Use the default expiration interval (default).
>0	Expiration interval to be assigned to this connection.

- ◆ CheckEachNewOpenConnection (Output Only, Type INTEGER)

Value	Description
0	Return Success without calling WAN Traffic Manager, allowing the connection to proceed normally (default).
1	Call WAN Traffic Manager and let the policies decide whether to allow the connection.
2	Return ERR_CONNECTION_DENIED without calling WAN Traffic Manager, causing the connection to fail.

- ◆ CheckEachAlreadyOpenConnection (Output Only, Type INTEGER)

Value	Description
0	Return Success without calling WAN Traffic Manager, allowing the connection to proceed normally (default).
1	Call WAN Traffic Manager and let the policies decide whether to allow the connection.

Value	Description
2	Return ERR_CONNECTION_DENIED without calling WAN Traffic Manager, causing the connection to fail.

- ◆ Next (Output Only, Type TIME)

Time for the next round of limber checking. If this is not set, NDS_LIMBER will use the default.

Sample NDS_LIMBER_OPEN

NDS_LIMBER_OPEN is used only if either CheckEachNewOpenConnection or CheckEachAlreadyOpenConnection was set to 1 during the corresponding NDS_LIMBER query. This query is generated whenever CheckEachNewOpenConnection is 1 and eDirectory needs to open a new connection before running limber. This query is generated whenever CheckEachNewOpenConnection is 1 and eDirectory needs to open a new connection before doing schema synchronization or when CheckEachAlreadyOpenConnection is 1 and eDirectory needs to reuse an already existing connection.

- ◆ Version (Input Only, Type INTEGER)

The version of eDirectory.

- ◆ ExpirationInterval (Input and Output, Type INTEGER)

The expiration interval that should be assigned to this connection.

Value	Description
<0, 0	Use the default expiration interval (default).
>0	Expiration interval to be assigned to this connection.

- ◆ ConnectionIsAlreadyOpen (Input Only, BOOLEAN)

Value	Description
TRUE	eDirectory determines that it already has a connection to this address and can reuse that connection.
FALSE	eDirectory does not have a connection to this address and must create one.

- ◆ ConnectionLastUsed (Input Only, Type TIME)

If ConnectionIsAlreadyOpen is TRUE, then ConnectionLastUsed is the last time that a packet was sent from DS using this connection. Otherwise, it is 0.

Value	Description
TRUE	ConnectionLastUsed is the time that eDirectory last sent a packet on this connection.
FALSE	ConnectionLastUsed will be 0.

Sample NDS_SCHEMA_SYNC

Before eDirectory synchronizes the schema, it queries WAN Traffic Manager to see if this is an acceptable time for this activity. The traffic type NDS_SCHEMA_SYNC does not have a destination address; it requires a NO_ADDRESSES policy. If WAN Traffic Manager returns DONT_SEND, schema synchronization is put off and rescheduled.

The following variables are provided:

- ◆ Last (Input Only, Type TIME)
The time of the last successful schema synchronization to all servers.
- ◆ Version (Input Only, Type INTEGER)
The version of eDirectory.
- ◆ ExpirationInterval (Output Only, Type INTEGER)
The expiration interval for all connections created while synchronizing the schema.

Value	Description
<0, 0	Use the default expiration interval (default).
>0	Expiration interval to be assigned to this connection.

- ◆ CheckEachNewOpenConnection (Output Only, Type INTEGER)

Value	Description
0	Return Success without calling WAN Traffic Manager, allowing the connection to proceed normally (default).
1	Call WAN Traffic Manager and let the policies decide whether to allow the connection.
2	Return ERR_CONNECTION_DENIED without calling WAN Traffic Manager, causing the connection to fail.

- ◆ CheckEachAlreadyOpenConnection (Output Only, Type INTEGER)

Value	Description
0	Return Success without calling WAN Traffic Manager, allowing the connection to proceed normally (default).
1	Call WAN Traffic Manager and let the policies decide whether to allow the connection.
2	Return ERR_CONNECTION_DENIED without calling WAN Traffic Manager, causing the connection to fail.

Sample NDS_SCHEMA_SYNC_OPEN

NDS_SCHEMA_SYNC_OPEN is used only if either CheckEachNewOpenConnection or CheckEachAlreadyOpenConnection was set to 1 during the corresponding NDS_SCHEMA_SYNC query. This query is generated whenever

CheckEachNewOpenConnection is 1 and eDirectory needs to open a new connection before doing schema synchronization or when CheckEachAlreadyOpenConnection is 1 and eDirectory needs to reuse an already existing connection.

- ◆ Version (Input Only, Type INTEGER)

The version of eDirectory.

- ◆ ExpirationInterval (Input and Output, INTEGER)

The expiration interval that should be assigned to this connection.

Value	Description
<0, 0	Use the default expiration interval (default).
>0	Expiration interval to be assigned to this connection.

- ◆ ConnectionIsAlreadyOpen (Input Only, BOOLEAN)

Value	Description
TRUE	eDirectory determines that it already has a connection to this address and can reuse that connection.
FALSE	eDirectory does not have a connection to this address and must create one.

- ◆ ConnectionLastUsed (Input Only, Type TIME)

If ConnectionIsAlreadyOpen is TRUE, then ConnectionLastUsed is the last time that a packet was sent from eDirectory using this connection. Otherwise, it is 0.

Value	Description
TRUE	<i>ConnectionLastUsed</i> is the time that eDirectory last sent a packet on this connection.
FALSE	<i>ConnectionLastUsed</i> will be 0.

Sample NDS_SYNC

Whenever eDirectory needs to synchronize a replica, it makes a query to WAN Traffic Manager using the traffic type NDS_SYNC. The following variables are provided by eDirectory for use in WAN policies:

- ◆ Last (Input Only, Type TIME)

Time of the last successful synchronization to this replica.

- ◆ Version (Input Only, Type INTEGER)

The version of eDirectory.

- ◆ ExpirationInterval (Output Only, Type INTEGER)

The expiration interval for the connection to the server holding the updated replica.

Value	Description
<0, 0	Use the default expiration interval (default).
>0	Expiration interval to be assigned to this connection.

Onospoof.wmg

The policies in this group allow only existing WAN connections to be used. There are two policies:

- ◆ Already Open, No Spoofing, NA
Prevents the checking of backlinks, external references, and login restrictions; the running of Janitor or Limber; and schema synchronization except on existing WAN connections.
- ◆ Already Open, No Spoofing
Prevents all other traffic to existing WAN connections.

To prevent all traffic to existing connections, both policies must be applied.

Opnspoof.wmg

The policies in this group allow only existing WAN connections to be used but assume that a connection that hasn't been used for 15 minutes is being spoofed and should not be used. There are two policies:

- ◆ Already Open, Spoofing, NA
This policy prevents the checking of backlinks, external references, and login restrictions; the running of Janitor or Limber; and schema synchronization except on existing WAN connections that have been open less than 15 minutes.
- ◆ Already Open, Spoofing
This policy prevents other traffic to existing WAN connections that have been open less than 15 minutes.

To prevent all traffic to existing connections open less than 15 minutes, both policies must be applied.

Samearea.wmg

The policies in this group allow traffic only in the same network area. A network area is determined by the network section of an address. In a TCP/IP address, Wan Traffic Manager assumes a class C address (addresses whose first three sections are in the same network area). In an IPX address, all addresses with the same network portion are considered to be in the same network area. There are three policies:

- ◆ Same Network Area, NA
Prevents the checking of backlinks, external references, and login restrictions; the running of Janitor or Limber; and schema synchronization unless the traffic that would be generated is in the same network area.
- ◆ Same Network Area, TCPIP
Restricts TCP/IP traffic unless the traffic that would be generated is in the same TCP/IP network area.

- ◆ Same Network Area, IXP

Restricts IPX traffic unless that traffic that would be generated is in the same IPX network area.

Tcpip.wmg

The policies in this group allow only TCP/IP traffic. There are two policies:

- ◆ TCPIP, NA

Prevents the checking of backlinks, external references, and login restrictions; the running of Janitor or Limber; and schema synchronization unless the traffic that would be generated is TCP/IP.

- ◆ TCPIP

Prevents all other traffic unless the traffic is TCP/IP.

To prevent all non-TCP/IP traffic, both policies must be applied.

Timecost.wmg

The policies in this group restrict all traffic to between 1 a.m. and 1:30 a.m. but allow servers in the same location to talk continuously. This group uses the following policies, all of which must be applied:

- ◆ COSTLT20

Has a priority of 40 for NA and address traffic.

- ◆ Disallow Everything

Allows no traffic to be sent. If WAN Traffic Manager finds no (0) policies where the selector returned greater than 0, it defaults to SEND. This policy prevents this case.

- ◆ NDS Synchronization

Restricts NDS_SYNC traffic to between 1 a.m. and 1:30 a.m.

- ◆ Start Rest. Procs, NA

Allows all processes to start at any time, but WAN Traffic Manager must be consulted for each *_OPEN call. It schedules the process to run four times a day at 1:00, 7:00, 13:00, and 19:00.

- ◆ Start Unrest. Procs 1-1:30, NA

Allows all processes to start between 1:00 a.m. and 1:30 a.m. and run to completion without further queries to WAN Traffic Manager. The processes run four times a day, every six hours. The 1:00 process is handled by this policy; the other processes are handled by the Start Rest. Procs, NA.

WAN Policy Structure

A WAN policy consists of three sections:

- ◆ [“Declaration Section” on page 252](#)
- ◆ [“Selector Section” on page 254](#)
- ◆ [“Provider Section” on page 254](#)

Declaration Section

The Declaration section of a policy contains definitions of local variables and variables coming in through a client request. These definitions are used within the Selector and Provider sections. These variables are stored along with system-defined variables.

Variable declarations are separated by a semicolon (;). Multiple declarations for the same type can be combined in one line or wrapped to the next line; they are not line sensitive. A sample Declaration section is shown below:

```
REQUIRED INT R1;
REQUIRED TIME R2;
REQUIRED BOOLEAN R3,R4;
REQUIRED NETADDRESS R5,R6;
OPTIONAL INT P1 := 10;
OPTIONAL BOOLEAN := FALSE;
LOCAL INT L1 :=10;
LOCAL INT L2;
LOCAL TIME L3;
LOCAL BOOLEAN L4 :=TRUE, L5 :=FALSE;
LOCAL NETADDRESS L6;
```

The required and optional declarations are specific to a particular traffic type. Policies that do not contain the required variables will not run. The optional declarations must have a value to provide a default if none is passed in. WAN Traffic Manager provides system symbols (predefined variables) for use with all traffic types.

Each declaration consists of three parts:

- ◆ Scope
- ◆ Type
- ◆ List of names/optional value pairs

Scope

Valid scopes are listed in the following table.

Scope	Description
REQUIRED	<p>Variables defined as REQUIRED in scope can be used in multiple sections, but only once within the Declaration section.</p> <p>No values can be defined for a REQUIRED scope variable. Its value must come from the GetWanPolicy request.</p>
OPTIONAL	<p>Variables defined as OPTIONAL in scope can be used in multiple sections of a policy, but only once within the Declaration section.</p> <p>OPTIONAL scope variables are assigned to a default value. These values are not initialized. They are set only if a value is not passed. If a WAN policy request does not pass a new value to the parameter that matches in both name and type, the value defined in the Declaration is used when processing the policy.</p> <p>You must assign a value to variables defined as OPTIONAL in scope. Therefore, because TIME and NETADDRESS types cannot be initialized in the Declaration section, do not use an OPTIONAL scope with these variable types.</p>

Scope	Description
LOCAL	<p>Variables defined as LOCAL in scope can be used in multiple sections, but only once within the Declaration section.</p> <p>LOCAL scope variables exist only for a particular policy; that is, their values are not returned to the calling client.</p> <p>All parameter types can be defined. However, because TIME and NETADDRESS types cannot be initialized in the Declaration section, do not assign values to these types.</p>
SYSTEM	<p>Variables defined as SYSTEM in scope can be used in multiple sections, but only once within the Declaration section.</p>

Type

Valid types are listed in the following table.

Type	Description
INT	<p>Reflects the traffic type of the GetWanPolicy request that the policy is being run for. For example, the following policy specifies a Traffic Type of NDS_SYNC:</p> <p>IF TrafficType=NDS_SYNC THEN <i>action</i> END.</p>
BOOLEAN	<p>Used for values of only TRUE or FALSE. The value will be indeterminate if it is not set in a Declaration or a WAN policy request.</p>
TIME	<p>TIME scope variables must receive their values in the Selector or Provider sections or from the WAN policy request. Do not assign values to TIME scope variables in the Declaration.</p>
NETADDRE SS	<p>NETADDRESS scope variables must receive their values in the Selector or Provider sections. Do not assign values to NETADDRESS scope variables in the Declaration.</p>

You cannot assign values to Time and Netaddress types in the Declaration section. If these types do not already have a value, they receive their values in the Selector or Provider sections. Only single types are initialized in the Declaration section.

Names/Optional Value Pairs

Variable names are combinations of alphanumeric characters in a string of any length. Because only the first 31 characters are used, a variable must begin with a unique 31-character string. A variable name must start with an alphabetic character, or the symbol is interpreted as a numeric constant.

Variable names are case sensitive. For example, the variable *Rl* is not the same as the variable *rl*. The underscore character (`_`) is allowed in variable names.

Values in a declaration must be constants rather than variables or expressions. Thus, the declaration `LOCAL INT L2 := L3;` is not allowed. A value initializing a variable in the Declaration section can be changed in the Selector and Provider sections of the policy.

Selector Section

The Selector section of a policy begins with the keyword `SELECTOR` and concludes with the keyword `END`. Selector sections are evaluated to determine which loaded policy will be used.

The Selector sections of all the currently loaded policies are run to determine which policy has the greatest weight. When evaluated, the section returns a weight between 0-100, where 0 means do not use this policy, 1-99 means use this policy if no other policy returns a higher value, and 100 means use this policy.

The result of a Selector section is given in a `RETURN` declaration. If no `RETURN` declaration is made, a default value of 0 is returned. The following is a sample Selector section:

```
SELECTOR
RETURN 49;
END
```

When the Selector sections of multiple policies are evaluated, more than one policy might return the same value. In this case, it is indeterminate which policy will be selected. All else being equal, a server policy overrides a WAN policy.

For more information on writing declarations, see [“Construction Used within Policy Sections” on page 254](#). See also [“Provider Section” on page 254](#).

Provider Section

The Provider section begins with the keyword `PROVIDER` and concludes with the keyword `END`. The body of the Provider section consists of a list of declarations.

The result of this Declarations list is a value representing the policy’s suggestion to `SEND` or `DONT_SEND`.

The result of a Provider section is given in a `RETURN` declaration. If no `RETURN` declaration is made, a default value of `SEND` is returned.

The following is a sample Provider section:

```
PROVIDER
RETURN SEND;
END
```

For more information on writing declarations, see [“Construction Used within Policy Sections” on page 254](#).

Construction Used within Policy Sections

The following statements and constructions can be used, except as noted, in the Selector and Provider sections of a WAN policy. For more information on how to construct the Declaration section of a policy, see [“Declaration Section” on page 252](#).

Comments

Comments can be indicated by using `/*` at the beginning of the line and `*/` at the end. For example:

```
/* This is a comment. */
```

Comments can also be distinguished by `//` at the end of the line before a comment. For example:

```
IF L2 > L3 THEN //This is a comment.
```

IF-THEN Statement

IF-THEN statements are used to run a block of declarations conditionally.

Examples:

```
IF Boolean_expression THEN declarations
END
```

```
IF Boolean_expression THEN declarations
ELSE declarations
END
```

```
IF Boolean_expression THEN declarations
ELSIF Boolean_expression THEN declarations
END
```

IF *Boolean_Expression* THEN

This is the first clause in an IF-THEN statement. The Boolean expression is evaluated for a TRUE or FALSE result. If it is TRUE, the declarations that immediately follow are run. If it is FALSE, execution jumps to the next corresponding ELSE, ELSIF, or END declaration.

ELSE

This declaration marks the beginning of declarations that run if all corresponding preceding IF-THEN and ELSIF statements result in FALSE. For example:

```
IF Boolean_expression THEN statements
ELSIF Boolean_expression THEN statements
ELSIF Boolean_expression THEN statements
ELSE statements
END
```

ELSIF *Boolean_Expression* THEN

The Boolean expression is evaluated if the preceding IF-THEN declaration returns a FALSE. The ELSIF declaration is evaluated for a TRUE or FALSE result. If it is TRUE, the declarations that follow are run. If it is FALSE, execution jumps to the next corresponding ELSE, ELSIF, or END declaration.

For example:

```
IF Boolean_expression THEN statements
ELSIF Boolean_expression THEN statements
ELSIF Boolean_expression THEN statements
END
```

END

The END declaration terminates an IF-THEN construction.

RETURN

The RETURN declaration gives the results of the Selector and Provider sections.

Selector

In a Selector section, the RETURN declaration provides the integer result used as a weight for the policy. RETURN assigns a policy weight between 0-100, where 0 means do not use this policy, 1-

99 means use this policy if no other policy returns a higher value, and 100 means use this policy. If no RETURN declaration is made in a Selector section, a default value of 0 is returned.

A semicolon (;) is required to terminate the declaration. For example:

```
RETURN 49;
RETURN L2;
RETURN 39+7;
```

Provider

In a Provider section, the RETURN declaration provides the SEND or DONT_SEND result. If no RETURN declaration is made, a default value of SEND is returned.

A semicolon (;) is required to terminate the declaration. For example:

```
RETURN SEND;
RETURN DONT_SEND;
RETURN L1;
```

Assignment

The assignment declaration changes the value of a symbol using the := characters. The defined variable or system variable is stated first, then the := with a value, variable, or operation following. The assignment declaration must be terminated with a semicolon (;). For example:

```
variable.field:=expression; variable:=expression;
```

t1 and t2 are of type TIME, i1 and i2 are type INTEGER, and b1 and b2 are Boolean valid assignments:

```
t1 := t2;
b1 := t1 < t2;
i1 := t1.mday - 15;
b2 := t2.year < 2000
```

Invalid assignments:

```
b1 := 10 < i2 < 12;
```

(10 < i2) is Boolean, and a BOOLEAN cannot be compared to an INTEGER.

You could use b1 := (10 < i2) AND (i2 < 12); instead. For example:

```
b2 := i1;
```

b2 is Boolean and i1 is INTEGER. Therefore, they are incompatible types.

You could use b2 := i1 > 0; instead.

Strict type checking is performed. You are not allowed to assign an INT to a TIME variable.

Arithmetic Operators

You can include arithmetic operators in assignment declarations, RETURN declarations, or IF constructions. The valid operators are

- ◆ Addition (+)
- ◆ Subtraction (-)
- ◆ Division (/)

- ◆ Multiplication (*)
- ◆ Module (MOD)

Use only INT variable types with arithmetic operators. Do not use TIME, NETADDRESS, or BOOLEAN variable types in arithmetic expressions.

Avoid operations that result in values outside of the range -2147483648 to +2147483648 or division by 0.

Relational Operators

You can use relational operators in IF constructions. The valid operators are

- ◆ Equal to (=)
- ◆ Not equal to (<>)
- ◆ Greater than (>)
- ◆ Greater than or equal to (>=)
- ◆ Less than (<)
- ◆ Less than or equal to (<=)

You can use any relational operators with TIME and INT variable types. You can also use <> and = with NET ADDRESS and BOOLEAN variable types.

Logical Operators

The valid operators are

- ◆ AND
- ◆ OR
- ◆ NOT
- ◆ Less than (<)
- ◆ Greater than (>)
- ◆ Equal to (=)

Bitwise Operators

You can use bitwise operators on INT variable types to return an integer value. The valid operators are

- ◆ BITAND
- ◆ BITOR
- ◆ BITNOT

Complex Operations

The following precedence rules are enforced when processing complex expressions. Operators with the same precedence order are processed left-to-right. The order is as follows:

- ◆ Parenthesis
- ◆ Unary (+/-)

- ◆ BITNOT
- ◆ BITAND
- ◆ BITOR
- ◆ Multiplication, division, MOD
- ◆ Addition, subtraction
- ◆ Relational (>, >=, <, <=, =)
- ◆ NOT
- ◆ AND
- ◆ OR

If you are not certain of precedence, use parentheses. For example, if A, B, and C are integers or variables, A<B<C is not allowed. A<B would return a Boolean value, not an integer value, which cannot be compared to an integer C. However, (A<B) AND (B<C) would be syntactically correct.

PRINT

You can use PRINT declarations to send text and symbol values to the server's WAN Traffic Manager display screen and to the log file.

PRINT statements can have any number of arguments that can be literal strings, symbol names or members, integer values, or Boolean values, separated by commas.

You must enclose literal strings in double quotes (“ ”). PRINT declarations must end in a semicolon (;). For example:

```
PRINT "INT=",10,"BOOL=",TRUE,"SYM=",R1;
```

TIME and NETADDRESS variables use formatted PRINT declarations. TIME symbols are printed as follows:

```
m:d:y h:m
```

NETADDRESS variables are printed as follows:

```
Type length data
```

Type is either IP or IPX, *length* is the number of bytes, and *data* is the hexadecimal address string.

11

Understanding LDAP Services for Novell eDirectory

The Lightweight Directory Access Protocol (LDAP) is an Internet communications protocol that lets client applications access directory information. It is based on the X.500 Directory Access Protocol (DAP) but is less complex than a traditional client and can be used with any other directory service that follows the X.500 standard.

LDAP is used most often as the simplest directory access protocol.

Lightweight Directory Access Protocol (LDAP) Services for Novell® eDirectory™ is a server application that lets LDAP clients access information stored in eDirectory.

LDAP Services includes eDirectory features that are available through LDAP:

- ◆ Provisioning
- ◆ Account Management
- ◆ Authentication
- ◆ Authorization
- ◆ Identity Management
- ◆ Notification
- ◆ Reporting
- ◆ Qualification
- ◆ Segmentation

You can give different clients different levels of directory access, and you can access the directory over a secure connection. These security mechanisms let you make some types of directory information available to the public, other types available to your organization, and certain types available only to specified groups or individuals.

The directory features available to LDAP clients depend on the functionality built into the LDAP client and the LDAP server. For example, LDAP Services for eDirectory lets LDAP clients read and write data in the eDirectory database if the client has the necessary permissions. Some clients have the capability to read and write directory data; others can only read it.

Some typical client features let clients do one or more of the following:

- ◆ Look up information about a specific person, such as an e-mail address or phone number.
- ◆ Look up information for all people with a given last name, or a last name that begins with a certain letter.
- ◆ Look up information about any eDirectory object or entry.
- ◆ Retrieve a name, e-mail address, business phone number, and home phone number.
- ◆ Retrieve a company name and city name.

The following sections provide information about LDAP Services for eDirectory:

- ◆ “Key Terms for LDAP Services” on page 260
- ◆ “Understanding How LDAP Works with eDirectory” on page 262
- ◆ “Using LDAP Tools on Linux, Solaris, AIX, or HP-UX” on page 271
- ◆ “Extensible Match Search Filter” on page 280

For in-depth information on LDAP Services, see the [Novell LDAP Developer Documentation \(http://developer.novell.com/ndk/doc_novell_edirectory.htm\)](http://developer.novell.com/ndk/doc_novell_edirectory.htm).

For more information on LDAP, see the following Web sites:

- ◆ The University of Michigan (<http://www.umich.edu/~dirsvcs/ldap/ldap.html>)
- ◆ An LDAP Roadmap & FAQ (<http://www.kingsmountain.com/ldapRoadmap.shtml>)
- ◆ LDAPzone.com (<http://www.ldapzone.com>)

Key Terms for LDAP Services

Clients and Servers

LDAP Client—An application (for example, Netscape* Communicator*, Internet Explorer, or Novell Import Conversion Export utility).

LDAP Server—A server where nldap.nlm (for NetWare®), nldap.dlm (for Windows 2000/NT), libnldap.so (for Linux, Solaris and AIX systems), or libnldap.sl (for HP-UX systems) is running.

Objects

LDAP Group object—Sets up and manages the Novell LDAP properties on an LDAP server.

This object is created when you install eDirectory. An LDAP Group object contains configuration information that can be conveniently shared among multiple LDAP servers.

LDAP Server Object—Sets up and manages the way LDAP clients access and use the information on a Novell LDAP server.

This object is created when you install eDirectory. An LDAP Server object represents server-specific configuration data.

The following figure illustrates an LDAP Server object in Novell iManager.



Referrals

Referral—A message that the LDAP server sends to the LDAP client telling the client that this server can't provide complete results and that more data might be on another LDAP server.

The referral contains all the information needed to progress the operation.

Scenario: An LDAP client issues a request to an LDAP server but the server can't find the target entry of the operation locally. Using the knowledge references that it has about partitions and other servers, the LDAP server identifies another server that knows more about the entry. The LDAP server sends that information to the client.

The client establishes a new LDAP connection with the identified server and retries the operation.

Referrals have the following advantages:

- ◆ The LDAP client keeps control of the operation.

Because the client always knows what is happening, it can make better decisions and provide feedback to the user. Also, the client can opt not to follow through on a referral, or prompt a user before following it.

- ◆ Referrals often use network resources more efficiently than chaining.

In chaining, a requested search operation with many entries could be transmitted across the network twice. The first transmission would come from the server holding the data to the server doing the chaining. The second transmission would come to the client from the server doing the chaining.

With a referral, the client gets the data directly from the server that held the data, in one transmission.

- ◆ When a client knows where an entry is stored, the client can go directly to the server that has the data.

Chaining hides details from the client. Not knowing where data came from previously, the client most likely won't go directly to the server holding the data.

Referrals have the following disadvantages:

- ◆ The client must be able to recognize referrals and know how to follow them.
- ◆ LDAPv2 clients don't recognize referrals, or they use an obsolete, non-standard method for recognizing them.
- ◆ Every eDirectory partition must be serviced by an LDAP server.

Otherwise, referrals won't be sent for data in that partition.

Superior Referral—A referral to a server that holds data higher in the tree than the server being communicated with. See [“Configuring for Superior Referrals” on page 307](#).

Superior referrals deal with requests concerning objects that are in a higher or contiguous non-eDirectory partition of a multi-vendor tree.

To enable an eDirectory server to participate in this type of tree, eDirectory holds the hierarchical data above it in a partition marked as “nonauthoritative.” The objects in the non-authoritative area consist only of those entries needed to build the correct DN hierarchy. These entries are analogous to X.500 “Glue” entries.

eDirectory allows the placement of knowledge information in the form of LDAP referral data within the nonauthoritative area. This information is used to return referrals to the LDAP client.

When an LDAP operation takes place in a nonauthoritative area of the eDirectory tree, the LDAP server locates the correct reference data and returns a referral to the client.

Chaining—A server-based name-resolution protocol.

An LDAP client issues a request to an LDAP server, but the server can't find the target entry of the operation locally. Using the knowledge references that it has about partitions and other servers in the eDirectory tree, the LDAP server identifies another LDAP server that knows more about the DN. The first LDAP server then contacts the identified (second) LDAP server.

If necessary, this process continues until the first server contacts a server that holds a replica of the entry. eDirectory then handles all the details to complete the operation. Unaware of the server-to-server operations, the client assumes that the first server completed the request.

Through chaining, an LDAP server provides the following advantages:

- ◆ Hides all name-resolution details from the client
- ◆ Automatically takes care of reauthentication
- ◆ Acts as a proxy for the client
- ◆ Works seamlessly, even when some servers in the eDirectory tree don't support LDAP Services.

Chaining has the following disadvantages:

- ◆ The client might have to wait for some time without any feedback from the server, while the server chains to resolve the name.
- ◆ If the operation requires the LDAP server to send many entries across a WAN link, the operation might be very time consuming.
- ◆ If several servers are equally capable of progressing the operation, different servers might process two requests to operate on the same entry.

eDirectory attempts to sort the servers by the cost associated with contacting them. For load balancing, eDirectory randomly selects among servers with the lowest cost.

Understanding How LDAP Works with eDirectory

This section explains the following:

- ◆ [“Connecting to eDirectory from LDAP” on page 262](#)
- ◆ [“Class and Attribute Mappings” on page 265](#)
- ◆ [“Enabling Nonstandard Schema Output” on page 268](#)
- ◆ [“Syntax Differences” on page 269](#)
- ◆ [“Supported Novell LDAP Controls and Extensions” on page 270](#)

Connecting to eDirectory from LDAP

All LDAP clients bind (connect) to Novell eDirectory as one of the following types of users:

- ◆ [Public] User (Anonymous Bind)
- ◆ Proxy User (Proxy User Anonymous Bind)
- ◆ NDS or eDirectory User (NDS User Bind)

The type of bind the user authenticates with determines the content that the LDAP client can access. LDAP clients access a directory by building a request and sending it to the directory. When an LDAP client sends a request through LDAP Services for eDirectory, eDirectory completes the request for only those attributes that the LDAP client has the appropriate access rights to.

For example, if the LDAP client requests an attribute value (which requires the Read right) and the user is granted only the Compare right to that attribute, the request is rejected.

Standard login restrictions and password restrictions still apply. However, any restrictions are relative to where LDAP is running. Time and address restrictions are honored, but address restrictions are relative to where the eDirectory login occurred—in this case, the LDAP server.

Connecting As a [Public] User

An anonymous bind is a connection that does not contain a username or password. If an LDAP client without a name and password binds to LDAP Services for eDirectory and the service is not configured to use a Proxy User, the user is authenticated to eDirectory as user [Public].

User [Public] is a non-authenticated eDirectory user. By default, user [Public] is assigned the Browse right to the objects in the eDirectory tree. The default Browse right for user [Public] allows users to browse eDirectory objects but blocks user access to the majority of object attributes.

The default [Public] rights are typically too limited for most LDAP clients. Although you can change the [Public] rights, changing them will give these rights to all users. Because of this, we recommend that you use the Proxy User Anonymous Bind. For more information, see [“Connecting As a Proxy User” on page 263](#).

To give user [Public] access to object attributes, you must make user [Public] a trustee of the appropriate container or containers and assign the appropriate object and attribute rights.

Connecting As a Proxy User

A proxy user anonymous bind is an anonymous connection linked to an eDirectory username. If an LDAP client binds to LDAP for eDirectory anonymously, and the protocol is configured to use a Proxy User, the user is authenticated to eDirectory as the Proxy User. The name is then configured in both LDAP Services for eDirectory and in eDirectory.



The anonymous bind traditionally occurs over port 389 in LDAP. However, during the installation you can manually configure different ports.

The key concepts of proxy user anonymous binds are as follows:

- ◆ All LDAP client access through anonymous binds is assigned through the Proxy User object.
- ◆ Because LDAP clients do not supply passwords during anonymous binds, the Proxy User must have a null password and must not have any password restrictions (such as password change intervals). Do not force the password to expire or allow the Proxy User to change passwords.
- ◆ You can limit the locations that the user can log in from by setting address restrictions for the Proxy User object.
- ◆ The Proxy User object must be created in eDirectory and assigned rights to the eDirectory objects you want to publish. The default user rights provide Read access to a limited set of objects and attributes. Assign the Proxy User Read and Search rights to all objects and attributes in each subtree where access is needed.

- ♦ The Proxy User object must be enabled on the General page of the LDAP Group object that configures LDAP Services for eDirectory. Because of this, there is only one Proxy User object for all servers in an LDAP group. For more information, see [“Configuring LDAP Objects” on page 288](#).
- ♦ You can grant a Proxy User object rights to All Properties (default) or Selected Properties.


To give the Proxy User rights to only selected properties:

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click Rights > Modify Trustees.
- 3** Specify the name and context of the top container the Proxy User has rights over, or click  to browse to the container in question, then click OK.
- 4** On the Modify Trustees screen, click Add Trustee.
- 5** Browse to and click the Proxy User’s object, then click OK.
- 6** Click Assigned Rights to the left of the Proxy User you just added.
- 7** Check the All Attributes Rights and Entry Rights check boxes, then click Delete Property.
- 8** Click Add Property, then check the Show All Properties in Schema check box.
- 9** Select an inheritable right for the Proxy User, such as mailstop (in the lowercase section of the list) or Title, then click OK.

To add additional inheritable rights, repeat Steps 9 and 10.

- 10** Click Done, then click OK.

To implement proxy user anonymous binds, you must create the Proxy User object in eDirectory and assign the appropriate rights to that user. Assign the Proxy User Read and Search rights to all objects and attributes in each subtree where access is needed. You also need to enable the Proxy User in LDAP Services for eDirectory by specifying the same proxy username.

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click LDAP > LDAP Overview.
- 3** Click the name of an LDAP Group object to configure.
- 4** Specify the name and context of an eDirectory User object in the Proxy User field.
- 5** Click Apply, then click OK.

Using the ldapconfig Utility on UNIX

For example, LDAP Search Referral Usage specifies how the LDAP server processes LDAP referrals.

- 1** At a system prompt, enter the following command:


```
ldapconfig -s "LDAP:otherReferralUsage=1"
```
- 2** Enter the User FDN (Fully Distinguished eDirectory Username) and password.

Connecting As an NDS or eDirectory User

An eDirectory user bind is a connection that an LDAP client makes using a complete eDirectory username and password. The eDirectory user bind is authenticated in eDirectory, and the LDAP client is allowed access to any information the eDirectory user is allowed to access.

The key concepts of eDirectory user binds are as follows:

- ◆ eDirectory user binds are authenticated to eDirectory using the username and password entered at the LDAP client.
- ◆ The eDirectory username and password used for LDAP client access can also be used for NetWare client access to eDirectory.
- ◆ With non-TLS connections, the eDirectory password is transmitted in clear text on the path between the LDAP client and LDAP Services for eDirectory.
- ◆ If clear text passwords are not enabled, all eDirectory bind requests that include a username or password on non-TLS connections are rejected.
- ◆ If an eDirectory user password has expired, eDirectory bind requests for that user are rejected.

Assigning eDirectory Rights for LDAP Clients

1 Determine the type of username the LDAP clients will use to access eDirectory:

- ◆ [Public] User (Anonymous Bind)
- ◆ Proxy User (Proxy User Anonymous Bind)
- ◆ NDS User (NDS User Bind)

See [“Connecting to eDirectory from LDAP” on page 262](#) for more information.

2 If users will use one proxy user or multiple eDirectory usernames to access LDAP, use iManager to create these usernames in eDirectory or through LDAP.

3 Assign the appropriate eDirectory rights to the usernames that LDAP clients will use.

The default rights that most users receive provide limited rights to the user's own object. To provide access to other objects and their attributes, you must change the rights assigned in eDirectory.

When an LDAP client requests access to an eDirectory object and attribute, eDirectory accepts or rejects the request based on the LDAP client's eDirectory identity. The identity is set at bind time.

Class and Attribute Mappings

A *class* is a type of object in a directory, such as a user, server, or group. An attribute is a directory element that defines additional information about a specific object. For example, a User object attribute might be a user's last name or phone number.


A *schema* is a set of rules that defines the classes and attributes allowed in a directory and the structure of a directory (where the classes can be in relation to one another). Because the schemas of the LDAP directory and the eDirectory directory are sometimes different, mapping LDAP classes and attributes to the appropriate eDirectory objects and attributes might be necessary. These mappings define the name conversion from the LDAP schema to the eDirectory schema.

LDAP Services for eDirectory provides default mappings. In many cases, the correspondence between the LDAP classes and attributes and the eDirectory object types and properties is logical and intuitive. However, depending on your implementation needs, you might want to reconfigure the class and attribute mapping.

In most instances, the LDAP class to eDirectory object type mapping is a one-to-one relationship. However, the LDAP schema supports alias names such as CN and commonName that refer to the same attribute.

Mapping LDAP Group Attributes

The default LDAP Services for eDirectory configuration contains a predefined set of class and attribute mappings. These mappings map a subset of LDAP attributes to a subset of eDirectory attributes. If an attribute is not already mapped in the default configuration, an auto-generated map is assigned to the attribute. Also, if the schema name is a valid LDAP name with no spaces or colons, no mappings are required. You should examine the class and attribute mapping and reconfigure as needed.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click LDAP > LDAP Overview > View LDAP Groups.
- 3 Click an LDAP Group object, then click Attribute Map.
- 4 Add, delete, or modify the attributes you want.


Because there might be alternate names for certain LDAP attributes (such as CN and common name), you might need to map more than one LDAP attribute to a corresponding eDirectory attribute name. When LDAP Services for eDirectory returns LDAP attribute information, it returns the value of the first matched attribute it locates in the list.

If you map multiple LDAP attributes to a single eDirectory attribute, you should reorder the list to prioritize which attribute should take precedence because the order is significant.

- 5 Click Apply, then click OK.

Mapping LDAP Group Classes

When an LDAP client requests LDAP class information from the LDAP server, the server returns the corresponding eDirectory class information. The default LDAP Services for eDirectory configuration contains a predefined set of class and attribute mappings.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click LDAP > LDAP Overview.
- 3 Click an LDAP Group object, then click Class Map.
- 4 Add, delete, or modify the classes you want.

The default LDAP Services for eDirectory configuration contains a predefined set of class and attribute mappings. These mappings map a subset of LDAP classes and attributes to a subset of eDirectory classes and attributes. If an attribute or class is not mapped in the default configuration, an auto-generated map is assigned to the attribute or class.

Also, if the schema name is a valid LDAP name with no spaces or colons, no mappings are required. You should examine the class and attribute mapping and reconfigure as needed.

- 5 Click Apply, then click OK.

Mapping LDAP Classes and Attributes

Because the schemas of the LDAP directory and the eDirectory directory are different, mapping LDAP classes and attributes to the appropriate eDirectory objects and attributes is necessary. These mappings define the name conversion from the LDAP schema to the eDirectory schema.

No LDAP schema mappings are required for a schema entry if the name is a valid LDAP schema name. In LDAP, the only characters allowed in a schema name are alphanumeric characters and hyphens (-). No spaces are allowed in an LDAP schema name.

To ensure that searching by object IDs works after a schema extension other than LDAP, such as for .sch files, you must refresh the LDAP server configuration if the schema is extended outside of LDAP.

Many-to-One Mappings

To support LDAP from eDirectory, LDAP Services uses mappings in the protocol level (instead of the directory service level) to translate between LDAP and eDirectory attributes and classes. Because of this, two LDAP classes or attributes can be mapped to the same eDirectory class or attribute.

For example, if you create a Cn through LDAP and then search for CommonName=Value, you will get back a commonName, which might be the same attribute value for Cn.

If you request all attributes, you get the attribute that is first in the mappings list for that class. If you ask for an attribute by name, you will get the correct name.

Many-to-One Class Mappings

LDAP Class Name	eDirectory Class Name
alias aliasObject	Alias
groupOfNames groupOfUniqueNames group	Group
mailGroup rfc822mailgroup	NSCP:mailGroup1

Many-to-One Attribute Mappings

LDAP Attribute Name	eDirectory Attribute Name
c countryName	C
cn commonName	CN
uid userId	uniqueID
description multiLineDescription	Description
l localityname	L
member uniqueMember	Member
o organizationname	O
ou organizationalUnitName	OU
sn surname	Surname

LDAP Attribute Name	eDirectory Attribute Name
st stateOrProvinceName	S
certificateRevocationList;binary certificateRevocationList	certificateRevocationList
authorityRevocationList;binary authorityRevocationList	authorityRevocationList
deltaRevocationList;binary deltaRevocationList	deltaRevocationList
cACertificate;binary cACertificate	cACertificate
crossCertificatePair;binary crossCertificatePair	crossCertificatePair
userCertificate;binary userCertificate	userCertificate

NOTE: The attributes with ;binary are security related. They are in the mapping table in case your application needs the name retrieved with ;binary. If you need it retrieved without ;binary, you can change the order of the mappings.

Enabling Nonstandard Schema Output

eDirectory contains a compatibility mode switch that allows nonstandard schema output so that current ADSI and old Netscape clients can read the schema. This is implemented by setting an attribute in the LDAP Server object. The attribute name is nonStdClientSchemaCompatMode. The LDAP Server object is usually in the same container as the Server object.


The nonstandard output does not conform to the current IETF standards for LDAP, but it will work with the current version of ADSI and old Netscape clients.

In nonstandard output format:

- ◆ SYNTAX OID is single quoted.
- ◆ No upper bounds are output.
- ◆ No X- options are output.
- ◆ If more than one name is present, only the first encountered is output.
- ◆ Any attributes or classes without an OID defined will be output “attributename-oid” or “classname-oid” in lowercase.
- ◆ Attributes or classes with a hyphen in the name and no defined OID are not output.

OID or Object Identifier is a string of octet digits that is required to add an attribute or objectclass of your own to an LDAP server.

To enable nonstandard schema output:

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click LDAP > LDAP Overview.

- 3** Click View LDAP Servers, then click an LDAP Server object.
- 4** Click Searches, then click Enable old ADSI and Netscape Schema Output.

The nonstandard output does not conform to the current IETF defined standards for LDAP, but it works with the current ADSI and old Netscape clients.

- 5** Click Apply, click Information, then click Refresh.

Syntax Differences

LDAP and eDirectory use different syntaxes. Some important differences include the following:

- ♦ “Commas” on page 269
- ♦ “Typeful Names” on page 269
- ♦ “Escape Character” on page 269
- ♦ “Multiple Naming Attributes” on page 270

Commas

LDAP uses commas as delimiters rather than periods. For example, a distinguished (or complete) name in eDirectory looks like this:

```
CN=JANEBOU=MKTGO=EMA
```

Using LDAP syntax, the same distinguished name would be

```
CN=JANEBOU=MKTGO=EMA
```

Some additional examples of LDAP distinguished names:

```
CN=Bill Williams,OU=PR,O=Bella Notte Corp
```

```
CN=Susan Jones,OU=Humanities,O=University College London,C=GB
```

Typeful Names

eDirectory uses both typeless (.JOHN.MARKETING.ABCCORP) and typeful (CN=JOHN.OU=MARKETING.O=ABCCORP) names. LDAP uses only typeful names with commas as the delimiters (CN=JOHN,OU=MARKETING,O=ABCCORP).

Escape Character

The backslash (\) is used in LDAP distinguished names as an escape character. If you use the plus sign (+) or the comma (,), you can escape them with a single backslash character.

For example:

```
CN=Pralines\+Cream,OU=Flavors,O=MFG (CN is Pralines+Cream)
```

```
CN=DCardinal,O=Lionel\,Turner and Kaye,C=US (O is Lionel, Turner, and Kaye)
```

See Internet Engineering Task Force [RFC 232](http://www.ietf.org/rfc/rfc2253.txt?number=2253) (<http://www.ietf.org/rfc/rfc2253.txt?number=2253>) for more information.

Multiple Naming Attributes

Objects can be defined with multiple naming attributes in the schema. In both LDAP and eDirectory, the User object has two: CN and UID. The plus sign (+) separates the naming attributes in the distinguished name. If the attributes are not explicitly labeled, the schema determines which string goes with which attribute (the first would be CN, the second is UID for eDirectory and LDAP). You can reorder them in a distinguished name if you manually label each portion.

For example, the following are two relative distinguished names:

Smith (CN is Smith CN=Smith)

Smith+Lisa (CN is Smith, the OU is Lisa CN=Smith UID=Lisa)

Both relative distinguished names (Smith and Smith+Lisa) can exist in the same context because they must be referenced by two completely different relative distinguished names.

Supported Novell LDAP Controls and Extensions

The LDAP 3 protocol allows LDAP clients and LDAP servers to use controls and extensions for extending an LDAP operation. Controls and extensions allow you to specify additional information as part of a request or a response. Each extended operation is identified by an Object Identifier (OID), which is a string of octet digits that are required to add an attribute or objectclass of your own to an LDAP server. LDAP clients can send extended operation requests specifying the OID of the extended operation that should be performed and the data specific to that extended operation. When the LDAP server receives the request, it performs the extended operation and sends a response containing an OID and any additional data to the client.

For example, a client can include a control that specifies a sort with the search request that it sends to the server. When the server receives the search request, it sorts the search results before sending the search results back to the client. Servers can also send controls to clients. For example, a server can send a control with the authentication request that informs the client about password expiration.

By default, the eDirectory LDAP server loads all system extensions and selected optional extensions and controls when the LDAP server starts up. The extensionInfo attribute of LDAP Server object for optional extensions allows the system administrator to select or deselect the optional extensions and controls.

To enable extended operations, LDAP 3 protocol requires servers to provide a list of supported controls and extensions in the supportedControl attribute and supportedExtension attribute in the rootDSE. RootDSE (DSA [Directory System Agent] Specific Entry) is an entry that is located at the root of the Directory Information Tree (DIT). For more information, see [“Getting Information about the LDAP Server” on page 313](#).

For a list of supported LDAP controls and extensions, see [LDAP Controls \(http://developer.novell.com/ndk/doc/ldapover/ldap_enu/data/cchbehhc.html\)](http://developer.novell.com/ndk/doc/ldapover/ldap_enu/data/cchbehhc.html) and [LDAP Extensions \(http://developer.novell.com/ndk/doc/ldapover/ldap_enu/data/a6ik7oi.html\)](http://developer.novell.com/ndk/doc/ldapover/ldap_enu/data/a6ik7oi.html) in the *LDAP and NDS Integration Guide*.

Using LDAP Tools on Linux, Solaris, AIX, or HP-UX

eDirectory includes the following LDAP tools, stored in /usr/ldaptools/bin (except ice which is stored in /usr/bin), to help you manage the LDAP directory sever.

Tool	Description
ice	Imports entries from a file to an LDAP directory, modifies the entries in a directory from a file, exports the entries to a file, and adds attribute and class definitions from a file.
ldapadd	Adds new entries to an LDAP directory.
ldapdelete	Deletes entries from an LDAP directory server. The ldapdelete tool opens a connection to an LDAP server, binds, and deletes one or more entries.
ldapmodify	Opens a connection to an LDAP server, binds, and modifies or adds entries.
ldapmodrdn	Modifies the relative distinguished name (RDN) of entries in an LDAP directory server. Opens a connection to an LDAP server, binds, and modifies the RDN of entries.
ldapsearch	Searches entries in an LDAP directory server. Opens a connection to an LDAP server, binds, and performs a search using the specified filter. The filter should conform to the string representation for LDAP filters as defined in RFC 2254 (http://www.ietf.org/rfc/rfc2254.txt).
ndsindex	Creates, lists, suspends, resumes, or deletes indexes.

For more information, see [LDAP Tools](http://developer.novell.com/ndk/doc/cldap/lttoolenu/data/hevgtl7k.html) (<http://developer.novell.com/ndk/doc/cldap/lttoolenu/data/hevgtl7k.html>) in the *LDAP Libraries for C Guide*.

To perform secure LDAP tools operations, refer to “[Ensuring Secure eDirectory Operations on Linux, Solaris, AIX, and HP-UX Systems](#)” on page 81 and include the DER file in all command line LDAP operations that establish secure LDAP connections to eDirectory.

LDAP Tools

The LDAP utilities can be used to delete entries, modify entries, add entries, extend the schema, modify relative distinguished names, move entries to new containers, create search indexes, or perform searches.

ldapadd

The ldapadd utility adds new entries. It has the following syntax:

```
ldapadd [-c] [-C] [-l] [-M] [-P] [-r] [-n] [-v] [-F] [-l limit] [-M[M]] [-d
debuglevel] [-e key filename] [-D binddn] [[-W ]| [-w passwd]] [-h ldaphost]
[-p ldapport] [-P version] [-Z[Z]] [-f file]
```

NOTE: On a NetWare server, this utility is called ladd.

If the -f option is specified, ldapadd reads the modifications from a file. If the -f option is not specified, ldapadd reads the modifications from stdin.

TIP: Output from the ldap utilities is sent to stdout. If the utility exits before you can view the output, redirect the output to a file, for example, ldapadd [options] > out.txt.

Option	Description
-a	Adds new entries. The default for <code>ldapmodify</code> is to modify existing entries. If invoked as <code>ldapadd</code> , this flag is always set.
-r	Replaces existing values by default.
-c	Continuous operation mode. Errors are reported, but <code>ldapmodify</code> will continue with modifications. The default is to exit after reporting an error.
-f <i>file</i>	Reads the entry modification information from an LDIF file instead of from standard input. The maximum length of a record is 4096 lines.
-F	Forces the application of all changes regardless of the contents of input lines that begin with <code>replica:</code> . (By default, <code>replica:</code> lines are compared against the LDAP server host and port in use to decide if a relog record should actually be applied.)

Common Options for All LDAP Tools

There are some options that are common to all `ldap` tools. These are listed in the following table:

Option	Description
-C	Enables referral following. (anonymous bind)
-d <i>debuglevel</i>	Sets the LDAP debugging level to <i>debuglevel</i> . The <code>ldapmodify</code> tool must be compiled with <code>LDAP_DEBUG</code> defined for this option to have any effect.
-D <i>binddn</i>	Uses <i>binddn</i> to bind to the LDAP directory. <i>binddn</i> should be a string-represented DN as defined in RFC 1779.
-e <i>key filename</i>	Files the certificate filename for SSL bind.
-f <i>file</i>	Reads a series of lines from <i>file</i> , performing one LDAP search for each line. In this case, the filter given on the command line is treated as a pattern, where the first occurrence of <code>%s</code> is replaced with a line from the file. If the file is a single hyphen (<code>-</code>) character, then the lines are read from standard input.
-h <i>ldaphost</i>	Specifies an alternate host on which the <code>ldap</code> server is running.
-l <i>limit</i>	Specifies the connection timeout (in seconds).
-M	Enables Manage DSA IT control. (non-critical)
-MM	Enables Manage DSA IT control. (critical)
-n	Shows what would be done, but does not actually modify entries. Useful for debugging in conjunction with <code>-v</code> .
-p <i>ldappport</i>	Specifies an alternate TCP™ port where the <code>ldap</code> server is listening.
-P <i>version</i>	Specifies the LDAP version (2 or 3).
-v	Uses verbose mode with many diagnostics written to standard output.
-w <i>passwd</i>	Uses <i>passwd</i> as the password for simple authentication.

Option	Description
-W	Prompts for simple authentication. This option is used instead of specifying the password on the command line.
-Z	<p>Starts TLS before binding to perform the operation. If an error occurs during the Start TLS operation the error is ignored and the operation continues. It is recommended that the -ZZ option be used in place of this option to cause the operation to abort if an error occurs.</p> <p>If a port is specified with this option, it must accept clear text connections.</p> <p>To verify the server identity, this option should be used in conjunction with the -e option to specify a server certificate file. This validates the server trusted root certificate when TLS is started. If the -e option is not specified, any certificate from the server is accepted.</p>
-ZZ	<p>Starts TLS before binding to perform the operation. If an error occurs during the Start TLS operation, the operation is aborted.</p> <p>If a port is specified with this option, it must accept clear text connections.</p> <p>To verify server identity, this option should be used in conjunction with the -e option to specify a server certificate file. This validates the server trusted root certificate when TLS is started. If the -e option is not specified, any certificate from the server is accepted.</p>

Examples

Assume that the file /tmp/entrymods exists and has the following contents:

```
dn: cn=Modify Me, o=University of Michigan, c=US
changetype: modify
replace: mail
mail: modme@terminator.rs.itd.umich.edu
-
add: title
title: Manager
-
add: jpegPhoto
jpegPhoto: /tmp/modme.jpeg
-
delete: description
-
```

In this case, the command `ldapmodify -b -r -f /tmp/entrymods` will replace the contents of the Modify Me entry's mail attribute with the value `modme@terminator.rs.itd.umich.edu`, add a title of Manager, add the contents of the file `/tmp/modme.jpeg` as a jpegPhoto, and completely remove the description attribute.

The same modifications as above can be performed using the older `ldapmodify` input format:

```

cn=Modify Me, o=University of Michigan, c=US
mail=modme@terminator.rs.itd.umich.edu
+title=Manager
+jpegPhoto=/tmp/modme.jpeg
-description

```

and the command:

```
ldapmodify -b -r -f /tmp/entrymods
```

Assume that the file /tmp/newentry exists and has the following contents:

```

dn: cn=Barbara Jensen, o=University of Michigan, c=US
objectClass: person
cn: Barbara Jensen
cn: B Jensen
sn: Jensen
title: Manager
mail: bjensen@terminator.rs.itd.umich.edu
uid: bjensen

```

In this case, the command `ldapadd -f /tmp/entrymods` will add a new entry for B Jensen, using the values from the file /tmp/newentry.

Assume that the file /tmp/newentry exists and has the following contents:

```

dn: cn=Barbara Jensen, o=University of Michigan, c=US
changetype: delete

```

In this case, the command `ldapmodify -f /tmp/entrymods` will remove B Jensen's entry.

ldapdelete

The `ldapdelete` utility deletes the specified entry. It opens a connection to an LDAP server, binds, and then deletes. It has the following syntax:

```

ldapdelete [-n] [-v] [-c] [-r] [-l] [-C] [-M] [-d debuglevel] [-e key
filename] [-f file] [-D binddn] [[-W] | [-w passwd]] [-h ldaphost] [-p
ldapport] [-Z[Z]] [dn]...

```

NOTE: On a NetWare server, the utility is called `ldelete`.

The `dn` parameter is a list of distinguished names of the entries to be deleted.

It interacts with the `-f` option in the following ways:

- ◆ If the `-f` option is missing from the command line and `dn`'s are specified on the command line, the utility deletes the specified entries.
- ◆ If both `dn` and the `-f` option are in the command line, the utility reads the file for the `dn`'s to delete and ignores any `dn`'s in the command line.
- ◆ If both `dn` and the `-f` option are missing in the command line, the utility reads the `dn` from stdin.

TIP: Output from the ldap utilities is sent to stdout. If the utility exits before you can view the output, redirect the output to a file, for example, `ldapdelete [options] > out.txt`.

Option	Description
-c	Continuous operation mode. Errors are reported, but ldapdelete will continue with deletions. The default is to exit after reporting an error.
-f <i>file</i>	Reads a series of lines from the file, performing one LDAP search for each line. In this case, the filter given on the command line is treated as a pattern, where the first occurrence of %s is replaced with a line from the file.
-r	Delete recursively.

NOTE: Refer to ["Common Options for All LDAP Tools" on page 272](#) for more details on common options.

Example

The command `ldapdelete "cn=Delete Me, o=University of Michigan, c=US"` will attempt to delete the entry named with the commonName Delete Me directly below the University of Michigan organizational entry. In this case, it would be necessary to supply a binddn and passwd for the deletion to be allowed (see the -D and -w options).

Idapmodify

The ldapmodify utility modifies the attributes of an existing entry or adds new entries. It has the following syntax:

```
ldapmodify [-a] [-c] [-C] [-M] [-P] [-r] [-n] [-v] [-F] [-l limit] [-M[M]] [-d debuglevel] [-e key filename] [-D binddn] [[-W] | [-w passwd]] [-h ldaphost] [-p ldap-port] [-P version] [-Z[Z]] [-f file]
```

NOTE: On a NetWare server, the utility is called lmodify.

If the -f option is specified, ldapmodify reads the modifications from a file. If the -f option is not specified, ldapmodify reads the modifications from stdin.

TIP: Output from the ldap utilities is sent to stdout. If the utility exits before you can view the output, redirect the output to a file, for example, `ldapmodify [options] > out.txt`.

Option	Description
-a	Adds new entries. The default for ldapmodify is to modify existing entries. If invoked as ldapadd, this flag is always set.
-r	Replaces existing values by default.
-c	Continuous operation mode. Errors are reported, but ldapmodify will continue with modifications. The default is to exit after reporting an error.
-f <i>file</i>	Reads the entry modification information from an LDIF file instead of from standard input. The maximum length of a record is 4096 lines.
-F	Forces the application of all changes regardless of the contents of input lines that begin with replica:. (By default, replica: lines are compared against the LDAP server host and port in use to decide if a relog record should actually be applied.)

NOTE: Refer to ["Common Options for All LDAP Tools" on page 272](#) for more details on common options.

ldapmodrdn

The ldapmodrdn modifies the relative distinguished name of an entry. It can also move the entry to a new container. It has the following syntax:

```
ldapmodrdn [-r] [-n] [-v] [-c] [-C] [-l] [-M] [-s newsuperior] [-d debuglevel]
[-e key filename] [-D binddn] [[-W] | [-w passwd]] [-h ldaphost] [-p ldapport]
[-Z[Z]] [-f file] [dn newrdn]
```

NOTE: On a NetWare server, the utility is called lmodrdn dn <newrdn>).

Output from the ldap utilities is sent to stdout. If the utility exits before you can view the output, redirect the output to a file, for example, ldapmodrdn [options] > out.txt.

Option	Description
-c	Continuous operation mode. Errors are reported, but ldapmodify will continue with modifications. The default is to exit after reporting an error.
-f <i>file</i>	Reads the entry modification information from the file instead of from standard input or the command line. Make sure that there are no blank lines between the old RDN and new RDN, or the -f option will fail.
-r	Removes old RDN values from the entry. The default is to keep old values.
-s <i>newsuperior</i>	Specifies the distinguished name of the container to which the entry is moving.

NOTE: Refer to “Common Options for All LDAP Tools” on page 272 for more details on common options.

Example

Assume that the file /tmp/entrymods exists and has the following contents:

```
cn=Modify Me, o=University of Michigan, c=US
cn=The New Me
```

ldapsearch

The ldapsearch utility searches the directory for specified attributes and object classes. It has the following syntax:

```
ldapsearch [-n] [-u] [-v] [-t] [-A] [-T] [-C] [-V] [-M] [-P] [-L] [-d
debuglevel] [-e key filename] [-f file] [-D binddn] [[-W] | [-w bindpasswd]]
[-h ldaphost] [-p ldapport] [-b searchbase] [-s scope] [-a deref] [-l time
limit] [-z size limit] [-Z[Z]] filter [attrs....]
```

NOTE: On a NetWare server, the utility is called lsearch.

The ldapsearch tool opens a connection to an LDAP server, binds, and performs a search using the filter. The filter should conform to the string representation for LDAP filters as defined in RFC 2254 (<http://www.ietf.org/rfc/rfc2254.txt>).

If ldapsearch finds one or more entries, the attributes specified by attrs are retrieved and the entries and values are printed to standard output. If no attributes are listed, all attributes are returned.

TIP: Output from the ldap utilities is sent to stdout. If the utility exits before you can view the output, redirect the output to a file, for example, ldapsearch [options] filter [attribute list] > out.txt.

Option	Description
-a <i>deref</i>	Specifies how to handle the dereferencing of an alias. It uses the following values: <ul style="list-style-type: none"> ◆ Never: Aliases are never dereferenced while locating the base object or searching. ◆ Always: Aliases are always dereferenced when locating the base object and searching. ◆ Search: Aliases are dereferenced when searching subordinates of the base object but not when locating the base object. ◆ Find: Aliases are dereferenced when locating the base object but not when searching for the subordinates of the base object.
-A	Retrieves attributes only (no values). This is useful when you want to see if an attribute is present in an entry and when you are not interested in the specific values.
-CC	Enables referral following. (authenticated bind with same bind DN and password)
-b <i>searchbase</i>	Use <i>searchbase</i> as the starting point for the search.
-L	Prints entries in the LDIF format.
-LL	Prints entries in the LDIF format without comments.
-LLL	Prints entries in the LDIF format without comments and version.
-s <i>scope</i>	Specifies the scope of the search. Scope should be base, one, or sub to specify a base object, one-level, or subtree search. The default is sub.
-S <i>attribute</i>	Sorts the entries returned, based on attribute. The default is not to sort entries returned. If an attribute is a zero-length string (""), the entries are sorted by the components of their distinguished name. See <i>ldap_sort</i> for more details. <i>ldapsearch</i> normally prints out entries as it receives them. The use of the -S option defeats this behavior, causing all entries to be retrieved, sorted, and then printed.
-t	Writes retrieved binary values to a set of temporary files. This is useful for dealing with non-ASCII values such as jpegPhoto or audio.
-tt	Writes all values to temporary files.
-T <i>path</i>	Writes files to directory specified by path (default: "/tmp").
-u	Includes the user-friendly form of the distinguished name (DN) in the output.
-V	URL prefix for files.
-V <i>prefix</i>	Specifies the URL prefix for files (default: "file://tmp/").
-z <i>sizelimit</i>	Waits at most <i>sizelimit</i> entries for a search to complete.

NOTE: Refer to ["Common Options for All LDAP Tools"](#) on page 272 for more details on common options.

Examples

The following command:

```
ldapsearch "cn=mark smith" cn telephoneNumber
```

will perform a subtree search (using the default search base) for entries with a commonName of mark smith. The commonName and telephoneNumber values will be retrieved and printed to standard output. The output might look like the following if two entries are found:

```
cn=Mark D Smith, ou="College of Literature, Science, and the Arts",
ou=Students, ou=People, o=University of Michigan, c=US

cn=Mark Smith

cn=Mark David Smith

cn=Mark D Smith 1

cn=Mark D Smith

telephoneNumber=+1 313 930-9489

cn=Mark C Smith, ou=Information Technology Division, ou=Faculty and Staff,
ou=People,

o=University of Michigan, c=US

cn=Mark Smith

cn=Mark C Smith 1

cn=Mark C Smith

telephoneNumber=+1 313 764-2277
```

The command:

```
ldapsearch -u -t "uid=mcs" jpegPhoto audio
```

will perform a subtree search using the default search base for entries with user IDs of mcs. The user-friendly form of the entry's DN will be output after the line that contains the DN itself, and the jpegPhoto and audio values will be retrieved and written to temporary files. The output might look like the following if one entry with one value for each of the requested attributes is found:

```
cn=Mark C Smith, ou=Information Technology Division, ou=Faculty and Staff,
ou=People, o=University of Michigan, c=US

Mark C Smith, Information Technology Division, Faculty and Staff, People,
University of Michigan, US

audio=/tmp/ldapsearch-audio-a19924

jpegPhoto=/tmp/ldapsearch-jpegPhoto-a19924
```

The following command will perform a one-level search at the c=US level for all organizations whose organizationName begins with university.:

```
ldapsearch -L -s one -b "c=US" "o=university*" o description
```

Search results will be displayed in the LDIF format. The organizationName and description attribute values will be retrieved and printed to standard output, resulting in output similar to the following:

```
dn: o=University of Alaska Fairbanks, c=US
```

```

o: University of Alaska Fairbanks
description: Preparing Alaska for a brave new yesterday.
description: leaf node only
dn: o=University of Colorado at Boulder, c=US
o: University of Colorado at Boulder
description: No personnel information
description: Institution of education and research
dn: o=University of Colorado at Denver, c=US
o: University of Colorado at D

```

ndsindex

The `ndsindex` utility creates, lists, suspends, resumes, or deletes indexes. It has the following syntax:

```

ndsindex list [-h <hostname>] [-p <port>] -D <bind DN> -W|[-w <password>] [-l
limit] -s <eDirectory Server DN> [-Z[Z]] [<indexName1>, <indexName2>.....]

ndsindex add [-h <hostname>] [-p <port>] -D <bind DN> -W|[-w <password>] [-l
limit] -s <eDirectory Server DN> [-Z[Z]] <indexDefinintion1>
[<indexDefinintion2>.....]

ndsindex delete [-h <hostname>] [-p <port>] -D <bind DN> -W|[-w <password>]
[-l limit] -s <eDirectory Server DN> [-Z[Z]] <indexName1> [<indexName2>.....]

ndsindex resume [-h <hostname>] [-p <port>] -D <bind DN> -W|[-w <password>]
[-l limit] -s <eDirectory Server DN> [-Z[Z]] <indexName1> [<indexName2>.....]

ndsindex suspend [-h <hostname>] [-p <port>] -D <bind DN> -W|[-w <password>]
[-l limit] -s <eDirectory Server DN> [-Z[Z]] <indexName1> [<indexName2>.....]

```

NOTE: On a NetWare server, the utility is called `index`.

Option	Description
<code>list</code>	Lists the specified indexes. If the index is not specified, <code>ndsindex</code> lists all existing indexes on the server.
<code>add</code>	Creates new indexes.
<code>delete</code>	Deletes the specified indexes.
<code>resume</code>	Resumes the specified indexes from an off-line state.
<code>suspend</code>	Suspends the specified indexes to an off-line state.
<code>-s eDirectory Server DN</code>	Specifies the eDirectory Server DN.

NOTE: Refer to [“Common Options for All LDAP Tools”](#) on page 272 for more details on common options.

Examples

To list the indexes on the server `MyHost`, enter the following command:

```
ndsindex list -h MyHost -D cn=admin, o=mycompany -w password -s cn=MyHost,
o=novell
```

To create a substring index with the name MyIndex on the email address attribute, enter the following command:

```
ndsindex add -h myhost -D cn=admin, o=mycompany -w password -s cn=myhost,
o=novell "MyIndex;email address;substring"
```

To create a value index with the name MyIndex on the city attribute, enter the following command:

```
ndsindex add -h myhost -D cn=admin, o=mycompany -w password -s cn=myhost,
o=novell "MyIndex;city;value"
```

To create a presence index with the name MyIndex on the homephone attribute, enter the following command:

```
ndsindex add -h myhost -D cn=admin, o=mycompany -w password -s cn=myhost,
o=novell "MyIndex;homephone;presence"
```

To delete the index named MyIndex, enter the following command:

```
ndsindex delete -h myhost -D cn=admin, o=mycompany -w password -s
cn=myhost,o=novell MyIndex
```

To suspend the index named MyIndex, enter the following command:

```
ndsindex suspend -h myhost -D cn=admin, o=mycompany -w password -s cn=myhost,
o=novell MyIndex
```

To resume the index named MyIndex, enter the following command:

```
ndsindex resume -h myhost -D cn=admin, o=mycompany -w password -s cn=myhost,
o=novell MyIndex
```

Extensible Match Search Filter

The LDAP 3 core protocol specification defined in [RFC 2251 \(http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2251.html\)](http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2251.html) requires LDAP servers to recognize a search element called an extensible match filter. An extensible match allows an LDAP client to specify the following items in a search filter:

- ◆ An optional attribute name
- ◆ An optional matching rule
- ◆ A flag to indicate if the dn attributes should be considered a part of the entry
- ◆ The value to be used for the match

The following is the string representation of the extensible match search filter:

```
extensible = attr [":dn"] [":" matchingrule] ":@" value /
[:dn"] ":" matchingrule ":@" value
```

The following table lists the Extensible Match search filter parameters:

Parameter	Description
attr	Specifies the attribute to match on.

Parameter	Description
[":dn"]	Indicates that the matching rule should be included in the comparison match.
[": matchingrule]	Designates the matching rule to be used.
":="	Without a matching rule results in an equality match.
value	Comparison value

The extensibleMatch is a new filter provided in LDAP 3. If the matchingRule field is absent, the attribute field **MUST** be present, and the equality match is performed for that attribute. If the attribute field is absent and matchingRule is present, the matchValue is compared against all attributes in an entry that supports that matchingRule, and the matchingRule determines the syntax for the assertion value.

The filter item evaluates as

- ◆ TRUE if it matches with at least one attribute in the entry.
- ◆ FALSE if it does not match any attribute in the entry.
- ◆ Undefined if the matchingRule is not recognized or the assertionValue cannot be parsed.

If the type field along with the matchingRule is present, the matchingRule must be one permitted for use with that type, otherwise the filter item is undefined. If the :dn is specified in the search filter, the match is applied against all the attributes in an entry's distinguished name as well, and also evaluates to TRUE if there is at least one attribute in the distinguished name for which the filter item evaluates to TRUE. The dnAttributes field is present so that there does not need to be multiple versions of generic matching rules such as for word matching, one to apply to entries and another to apply to entries and dn attributes as well.

Essentially, an extensible match filter allows an LDAP client to achieve two objectives:

- ◆ Support multiple matching rules for same type of data
- ◆ Include DN elements in the search criteria

The DN specification allows matching on specific elements of the DN.

Novell eDirectory 8.7.3 supports the extensible match filter for matching on the DN attributes. The other elements of the extensible match search filter, namely the matching rule, are treated as undefined and ignored. The DN matching allows an LDAP client to drastically reduce the searches required to locate an object in an eDirectory tree. For example, a complex LDAP search filter such as

```
( & ( ou:dn:=sales ) ( objectclass=user ) )
```

would let you have a listing of all the User objects in the sales function (that is, anywhere under the sales containers).

Usage Examples

The following are examples of the string representations of extensible match search filter that are supported in eDirectory 8.7.3.

```
( o:dn:=Ace Industry )
```

This example illustrates the use of the `:dn` notation. The attributes of an entry's distinguished name should be considered part of the entry when evaluating the match. It denotes an equality match.

```
(:dn:2.4.8.10:=Dino)
```

This example is a filter that should be applied to any attribute of an entry. Attributes contained in the DN with the matching rule 2.4.8.10 should also be considered.

The following are some examples of the string representation of extensible match search filter that are *not* supported in eDirectory 8.7.3:

```
(cn:1.2.3.4.5:=John Smith)
```

This example illustrates a filter that specifies the attributes type `cn` and value `John Smith`. It mandates that the match should be performed by the directory server according to the matching rule identified by the oid 1.2.3.4.5.

```
(sn:dn:2.4.6.8.10:=Barbara Jones)
```

This 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.

12

Configuring LDAP Services for Novell eDirectory

The eDirectory™ installation program automatically installs LDAP Services for Novell® eDirectory. For information on installing eDirectory, see the *Novell eDirectory 8.7.3 Installation Guide*.

This section explains the following:

- ◆ “Loading and Unloading LDAP Services for eDirectory” on page 283
- ◆ “Verifying That the LDAP Server Is Loaded” on page 284
- ◆ “Verifying That the LDAP Server Is Running” on page 285
- ◆ “Configuring LDAP Objects” on page 288
- ◆ “Refreshing the LDAP Server” on page 292
- ◆ “Authentication and Security” on page 293
- ◆ “Using the LDAP Server to Search the Directory” on page 300
- ◆ “Configuring for Superior Referrals” on page 307
- ◆ “Persistent Search: Configuring for eDirectory Events” on page 311
- ◆ “Getting Information about the LDAP Server” on page 313

For information on LDAP tools, see [LDAP Tools \(http://developer.novell.com/ndk/doc/cldap/index.html?ldaplibc/data/a6eup29.html\)](http://developer.novell.com/ndk/doc/cldap/index.html?ldaplibc/data/a6eup29.html).

Loading and Unloading LDAP Services for eDirectory

To load LDAP Services for eDirectory, enter the following commands:

Server	Command
NetWare®	At the console prompt, enter <code>load nldap.nlm</code>
Windows	In the DHOST (NDSCONS) screen, click Nldap.dlm > Start.
Linux, Solaris, AIX, or HP-UX	At the Linux, Solaris, AIX, or HP-UX prompt, enter <code>/usr/sbin/nldap -l</code>

To unload LDAP Services for eDirectory, enter the following commands:

Server	Command
NetWare	At the console prompt, enter <code>unload nldap.nlm</code>
Windows	In the DHOST (NDSCONS) screen, click nldap.dlm > Stop.
Linux, Solaris, AIX, and HP-UX	In the DHOST remote management page, to unload LDAP, click the <i>LDAP v3 for Novell eDirectory 8.7.3</i> action icon to stop. or At the Linux, Solaris, AIX, or HP-UX prompt, enter <code>/usr/sbin/nldap -u</code>

Verifying That the LDAP Server Is Loaded

Before configuring LDAP objects, verify that the LDAP server is loaded and functional. This section explains how to verify that the LDAP server is loaded. To verify that the server is running and functional, see [“Verifying That the LDAP Server Is Running” on page 285](#).

On NetWare

To find out whether nldap.nlm is loaded on a NetWare server, enter one of the following at the server console:

- ◆ **ldap display activity**


If nldap.nlm is not loaded, the server displays `Unknown command`.

In NetWare 6.x, the display is written to the logger screen, not to the console screen.

- ◆ **ldap display config**

- ◆ **modules nldap.nlm**

You can also use Novell iManager.


- 1** Click the Roles and Tasks button .
- 2** Click eDirectory Maintenance > Service Manager.
- 3** Select a connection, server, or DNS name or IP address, then click OK.
- 4** Provide your password, then click OK.
- 5** Click LDAP Agent for Novell eDirectory 8.7.3.

The Module Information section displays nldap.nlm in the filename field.

On Windows 2000/NT

- 1** On a Windows server, open ndscons.exe.
Click Start > Settings > Control Panel > Novell eDirectory Services.
- 2** On the Services tab, scroll to nldap.dlm, then view the Status column.
The column displays Running.

You can also use Novell iManager.

- 1 Click the Roles and Tasks button .
- 2 Click eDirectory Maintenance > Service Manager.
- 3 Select a connection, server, or DNS name or IP address, then click OK.
- 4 Provide your password, then click OK.
- 5 Click LDAP Agent for Novell eDirectory 8.7.3.

The Module Information section displays `nldap.nlm` in the filename field.

Loaded on UNIX

Identify `libnldap.so` or `libnldap.sl`. This name might only be a symbolic link to a longer filename that has version information appended.

Also, each `libnldap.so` or `libnldap.sl` file is a different binary for each UNIX platform.

You can also use the `ndsd.log` file or `ndstrace` to check whether the LDAP server is loaded.

Verifying That the LDAP Server Is Running

After the LDAP server is loaded, verify that it is running. Then verify that a device is listening.

- ♦ [“Scenarios” on page 285](#)
- ♦ [“Verifying That The LDAP Server Is Running” on page 286](#)
- ♦ [“Verifying That A Device Is Listening” on page 287](#)

Scenarios

Typically, the LDAP server runs as soon as it is loaded. However, either of two scenarios can prevent the server from running properly.

Scenario: The Server Is in a Zombie State. The LDAP server loads as long as the NetWare or DHost Loaders can resolve external dependencies. However, the LDAP server doesn't run properly until it can get a valid configuration from the two configuration objects (the LDAP Server and LDAP Group objects).

While the LDAP server is in a loaded-but-not-running (zombie) state, it periodically tries to find and read the configuration objects. If the objects are misconfigured or corrupted, the LDAP server stays in the zombie state until the server (`nldap.nlm`, `nldap.dlm`, `libnldap.so`, or `libnldap.sl`) is unloaded or taken down.

The Loaders show that the LDAP server is loaded, but no LDAP ports (389, 636) are opened by `nldap.nlm` (or `nldap.dlm`, `libnldap.so`, or `libnldap.sl`). Also, no LDAP client requests are serviced.

DSTrace messages will show the periodic attempts and the reason why the server cannot come up to the running state.

Scenario: Denial of Service. At Digital Airlines, the server is processing a very long (20 minutes or more) search operation. The search is, in effect, looking for a needle in a haystack.

During this search, Henri does one of the following:

- ♦ Changes a configuration parameter and updates a configuration object.
- ♦ Clicks Refresh Server Now.

- ◆ Unloads the LDAP server (nldap.nlm, nldap.dlm, libnldap.so, or libnldap.sl).
- ◆ Tries to take the entire server down.

The LDAP server waits until all current operations complete before applying any new update. The server also postpones new operations from running until the update is complete. This delay can cause the server to appear to stop responding to new requests until the search is done and the server can refresh itself. Or the server appears to hang during the unload.

If the search request is long but has many hits, and Henri unloads the LDAP server, it aborts the search and quickly unloads when the next hit is returned to the client. However, if the search request has only one or no hits in 20 minutes, the LDAP server isn't able to abandon the NDS® or eDirectory request in progress.

For a refresh or update, the search will not be aborted even if it has many hits to return to the client.

Verifying That The LDAP Server Is Running

To verify that the LDAP service is running, use the Novell Import Conversion Export Utility (ICE). At a workstation, run `ice.exe` from the command line or use Novell iManager or ConsoleOne®.

At the Command Line

- 1 Go to the directory that contains `ice.exe` (for example, `c:\novell\consoleone\1.2\bin`).
- 2 Run `ice.exe`.

Search the rootDSE. Include parameters that identify the source handler and the export handler. For example enter

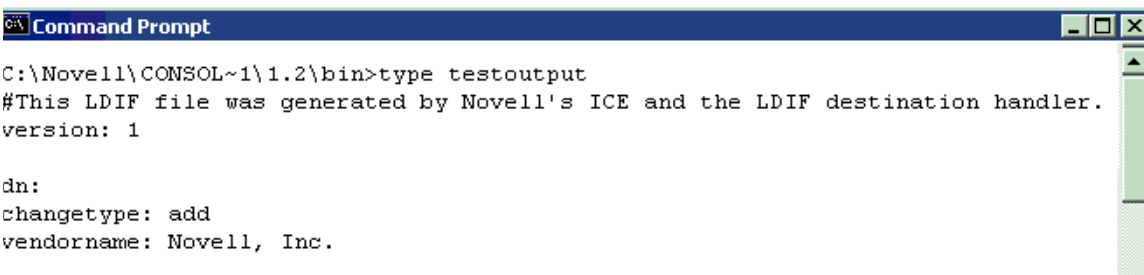
```
ice -S LDAP -s 10.128.45.0 -p 389 -c base -a vendorname -D LDIF -f
testoutput
```

Parameter and Value	Description
-S LDAP	Specifies LDAP as the source handler.
-s 10.128.45.0	Specifies the server's DNS name or IP address.
-p 389	Specifies the port number of the LDAP server that the LDAP source handler parameter identified. The default port is 389. If 389 is not the installed port, specify the clear-text port number.
-c base	Specifies that the scope of the search request is only the base object entry itself.
-a vendorname	Specifies that the search is to retrieve the vendorname attribute.
-D LDIF	Specifies LDIF as the destination handler.
-f testoutput	Specifies the filename where LDIF records can be written.

This example sends output to a `testoutput` file.

For more information on using ICE, see [“Novell Import Conversion Export Utility” on page 125](#). For information specific to LDAP source handlers, see [“LDAP Source Handler Options” on page 132](#). For information specific to LDIF destination handlers, see [“LDIF Destination Handler Options” on page 132](#).

3 View results of the ICE command.



```

C:\Novell\CONSOL~1\1.2\bin>type testoutput
#This LDIF file was generated by Novell's ICE and the LDIF destination handler.
version: 1

dn:
changetype: add
vendorname: Novell, Inc.

```

The example (Steps 2 and 3) limits the output from the rootDSE entry to the Vendor Name attribute. Because the example reads information from a Novell eDirectory server, the vendor information displays as Novell, Inc.

Using Novell iManager

To verify that the LDAP server is functional by using Novell iManager, follow steps in [“Exporting Data to a File” on page 126](#).

If you enter an IP address and a port number and then get a connection, the server is functional. Otherwise, you receive an error message. Download (view) either the log file or the export file.

Using ConsoleOne

To verify that the LDAP server is functional by using ConsoleOne, see [“Performing an LDIF Export” on page 140](#).

Specify a path and filename for the Select Destination LDIF File field (for example, c:\ldap\textoutput.txt). If you enter only a filename, the LDAP snap-in for ConsoleOne writes the file to the default directory (typically, c:\novell\consoleone\1.2\bin).

Verifying That A Device Is Listening

Verify that a device is listening on port 389.

For NetWare:

- 1 At the server console, enter
`tcpcon`
- 2 Select Protocol Information > TCP > TCP Connections.
- 3 Select 389 in the Port column.
If the State column displays Listen, a device is listening on that port.
If a device is not listening, the port will be missing altogether.

For Windows 2000/NT and UNIX

- 1 At the command line, enter
`netstat -a`
- 2 Find a line where the local address is *servername*:389 and the state is LISTENING.

If one of the following situations occurs, run Novell iMonitor:

- ◆ You are unable to get information from the ICE utility
- ◆ You are uncertain that the LDAP server is handling LDAP requests

For information on Novell iMonitor, see [“Configuration Files” on page 168](#) and [“Configuring Trace Settings” on page 175](#).

For information on LDAP requests, see [“Communicating with eDirectory through LDAP”](#) in the *Novell eDirectory 8.7.3 Installation Guide*.

Configuring LDAP Objects

An eDirectory installation creates an LDAP Server object and an LDAP Group object. The default configuration for LDAP Services is located in the directory on these two objects. You can modify the default configuration by using either the ConsoleOne LDAP snap-in or the LDAP Management task in Novell iManager.

The LDAP Server object represents server-specific configuration data.

The LDAP Group object contains configuration information that can be conveniently shared among multiple LDAP servers. This object provides common configuration data and represents a group of LDAP servers. The servers have common data.

You can associate multiple LDAP Server objects with one LDAP Group object. All the associated LDAP servers then get their server-specific configuration from their LDAP Server object but get common or shared information from the LDAP Group object.

By default, the eDirectory installation program installs a single LDAP Group object and a single LDAP Server object for each `nldap.nlm` or `nldap.dlm`. Later, you can associate multiple LDAP Server objects with a single LDAP Group object.

IMPORTANT: Although it is possible to associate newer versions of an LDAP Server object with older versions of LDAP Group objects, we recommend that you don't mix versions. For example, avoid associating an LDAP Group object in eDirectory 8.5 with an LDAP Server object in eDirectory 8.6.

The amount of common information held in an LDAP Group object is limited. LDAP doesn't need to read many attributes because the data contained in the attributes is incredibly common. Many LDAP servers will need to use the same data. Without a common or shared Group object, you would have to replicate that data across each LDAP server.

The LDAP Server object allows more server-specific configuration options and data than the LDAP Group object allows.

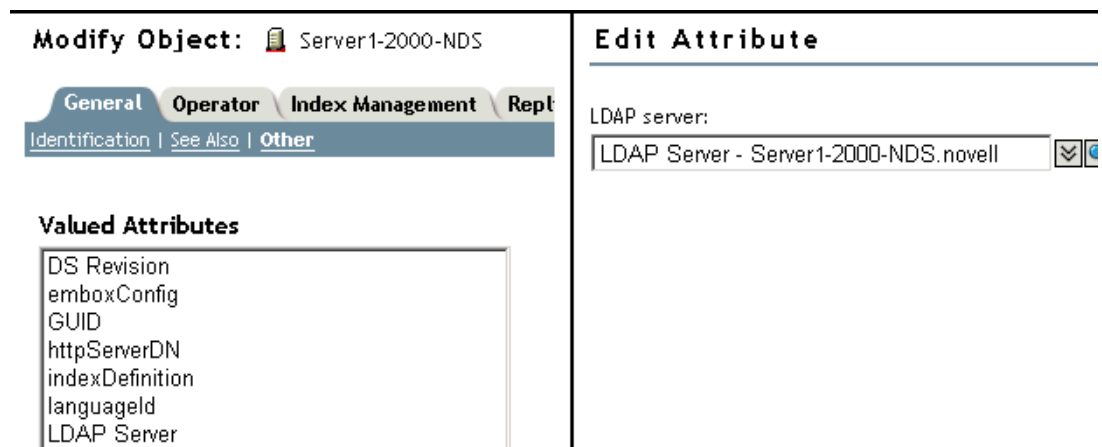
Both objects have DN-syntax attributes that point to each other.

An additional association must be made so that the LDAP server can find its configuration data. This association is through the NCP™ server, which holds the customary eDirectory configuration data. The eDirectory installation program automatically makes the association.

Every eDirectory server has an NCP Server object. In the following figure, server Lundi illustrates this object as displayed in iManager:



This object has an LDAP Server attribute, which points to the LDAP Server object for a particular host eDirectory server. The following figure illustrates this attribute:



Typically, the LDAP Server object, the LDAP Group object, and the NCP Server object are located in the same container. You name this container during the eDirectory installation, when you name the server and Admin context.

If you move the LDAP Server object, you must place it in a writable replica.

Configuring LDAP Server and LDAP Group Objects on Linux, Solaris, AIX, or HP-UX Systems

The LDAP configuration utility is `ldapconfig`. You can use `ldapconfig` on Linux, Solaris, AIX, or HP-UX systems to modify, view, and refresh the attributes of LDAP Server and LDAP Group objects.

Use the following syntax to view LDAP attribute values on Linux, Solaris, AIX, and HP-UX systems:

```
ldapconfig get [...] | set attribute-value-list [-t treename | -p
hostname[:port]] [-w password] [-a user FDN] [-f]
```

```
ldapconfig [-t tree_name | -p host_name[:port]] [-w password] [-a user FDN]
[-V] [-R] [-H] [-f] -v attribute,attribute2...
```

Use the following syntax to modify values of LDAP attributes on Linux, Solaris, AIX, and HP-UX systems:

```
ldapconfig [-t tree_name | -p host_name[:port]] [-w password] [-a admin_FDN]
-s attribute=value,...
```

Parameter	Description
-t <i>treename</i>	Name of the eDirectory tree where the component will be installed.
-p <i>hostname</i>	The name of the host. You could specify the DNS name or IP address also.
-w	The password of the user having administration rights.
-a	The fully distinguished name of the user having administration rights. For example: cn=user.o=org1
get -V	Lets you view all LDAP server/group attributes.
get -v <i>attribute list</i>	Displays the current values of the attributes in the attribute list.
set -s <i>attribute-value pairs</i>	Sets the attributes to the specified values.
-v	Lets you view the value of the LDAP attribute.
-s	Sets a value for an attribute of the installed components.
-R	Refreshes the LDAP server.
-V	Lets you view the current LDAP configuration settings.
-H	Lets you view the usage and help strings.
-f	Allows operations on a filtered replica.
<i>attribute</i>	A configurable LDAP server or group attribute name. For more information, see "Attributes on the LDAP Server Object" on page 290 and "Attributes on the LDAP Group Object" on page 292 .

Examples

To view the value of the attribute in the attribute list, enter the following command:

```
ldapconfig [-t tree_name | -p host_name[:port]]
  [-w password] [-a user_FDN] -v "Require TLS for simple binds with
  password", "searchTimeLimit"
```

To configure the LDAP TCP port number and search size limit to 1000, enter the following command:

```
ldapconfig [-t tree_name | -p host_name[:port]]
  [-w password] [-a admin_FDN] -s "LDAP TCP
  Port=389", "searchSizeLimit=1000"
```

Attributes on the LDAP Server Object

Use the LDAP Server object to set up and manage the Novell LDAP server properties.

The following table provides a description of the LDAP server attributes:

Attribute	Description
LDAP Server	The fully distinguished name of the LDAP server object in eDirectory.
LDAP Host Server	The fully distinguished name of the host eDirectory server that the LDAP server runs on.
LDAP Group	The LDAP Group object in eDirectory that this LDAP server is a member of.
LDAP Server Bind Limit	The number of clients that can simultaneously bind to the LDAP server. A value of 0 (zero) indicates no limit.
LDAP Server Idle Timeout	The period of inactivity from a client after which LDAP server terminates the connection with this client. A value of 0 (zero) indicates no limit.
LDAP Enable TCP	Indicates whether TCP (non-TLS) connections are enabled for this LDAP server. Value=1 (yes), 0 (no)
LDAP Enable TLS	Indicates whether TLS connections are enabled for this LDAP server. Value=1 (yes), 0 (no)
LDAP TCP Port	The port number that the LDAP server listens on for TCP (non-SSL) connections. Range=0 to 65535
LDAP TLS Port	The port number that the LDAP server listens on for TLS connections. Range=0 to 65535, the maximum number of connections allowed on the LDAP server.
keyMaterialName	The name of the Certificate object in eDirectory that is associated with this LDAP server and will be used for SSL LDAP connections.
searchSizeLimit	The maximum number of entries that the LDAP server will return to an LDAP client in response to a search. A value of 0 (zero) indicates no limit.
searchTimeLimit	The maximum number of seconds after which an LDAP search will be timed out by the LDAP server. A value of 0 (zero) indicates no limit.
filteredReplicaUsage	Specifies whether the LDAP server should use a filtered replica for an LDAP search. Values=1 (use filtered replica), 0 (do not use filtered replica)
sslEnableMutualAuthentication	Specifies whether SSL-based mutual authentication (Certificate-based client authentication) is enabled on the LDAP server.

Attribute	Description
IdapTLSVerifyClientCertificate	Enables or disables verification of the client certificate for a TLS operation through LDAP.
IdapNonStdAllUserAttrsMode	Enables or disables the non standard, all user, and operational attributes.
IdapBindRestrictions	<p>Sets the LDAP bind restrictions on the LDAP client connections. You can allow or disallow anonymous binds from the LDAP clients.</p> <p>Values=0, 1</p> <p>0 allows anonymous binds from the clients. 1 restricts the clients from doing anonymous binds.</p>
IdapEnablePSearch	<p>Specifies whether or not the persistent search feature is enabled on the LDAP server.</p> <p>Values= true, false</p>
IdapMaximumPSearchOperations	An integer value that limits the number of concurrent persistent search operations possible. A value of 0 specifies unlimited search operations.
IdapIgnorePSearchLimitsForEvents	<p>Indicates whether size and time limits should be ignored after the persistent search request has sent the initial result set.</p> <p>Values= true, false</p> <p>If this attribute is set to false, the entire persistent search operation is subject to the search limits. If either limit is reached, the search fails with the appropriate error message.</p>

Attributes on the LDAP Group Object

Use the LDAP Group object to set up and manage the way LDAP clients access and use the information on the Novell LDAP server.

To require TLS for simple binds, see [“Requiring TLS for Simple Binds with Passwords” on page 294](#). This attribute specifies whether the LDAP server allows transmission of passwords in clear text from an LDAP client. Values=0 (no) or 1 (yes).

To specify a default referral as well as how LDAP servers process LDAP referrals, see [“Using Referrals” on page 301](#).

Refreshing the LDAP Server

After you change a configuration option or setting on an LDAP server, you must refresh the server so that the changes can take effect.

However, you can't refresh the server while LDAP requests are being serviced. For example, if an operation requires a 15-minute walk of the eDirectory tree, the refresh won't occur until after that operation is complete.

Similarly, you can't take the LDAP server down while LDAP server threads are at work.

When a refresh is scheduled to occur, the LDAP server delays new LDAP requests from starting until after the refresh occurs.

By default, at 30-minute intervals the LDAP server checks the time stamps on the LDAP Server object and the LDAP Group object for changes to settings. If settings have changed, the server then implements the changes.

If the server discovers that time stamps on the settings have not changed, no refresh occurs. (If you force a refresh, the server ignores time stamps and makes the changes.)

To refresh the LDAP server, do one of the following:

- ◆ Use Novell iManager.
 1. On the Roles and Tasks page, click LDAP > LDAP Overview > View LDAP Servers.
 2. Click the LDAP server, then click Refresh.
- ◆ Wait for the server to reconfigure itself at the refresh interval.
- ◆ Unload and then reload nldap.nlm.

You don't have to unload any prerequisite NLM™ programs before unloading nldap.nlm.

Nldap.nlm unloads and then reloads dependent NLM programs.

- ◆ At the command line, change the refresh interval.

This option might be useful if you have WAN links that are not up continuously. You can temporarily make the server's heartbeat longer or shorter, as needed.

This change is not persistent. You must re-enter the command each time that you load nldap.nlm.

At the server console, enter

```
ldap refresh [=] [date] [time] [interval]
```

- ◆ The format for the date variable is mm:dd:yyyy. If you enter zeros for all date fields, the current date is used.
- ◆ The format for the time variable is hh:mm:ss. If you enter zeros for all time fields, the current time is used.
- ◆ The format for the interval variable is 0 or between 1 and 2147483647 minutes. If you enter zero, the default of 30 minutes is used.

You can add this command to the autoexec.ncf file in the sys:\system directory. Place the command after the line that loads nldap.nlm.

Authentication and Security

This section contains information on the following:

- ◆ [“Requiring TLS for Simple Binds with Passwords” on page 294](#)
- ◆ [“Starting and Stopping TLS” on page 294](#)
- ◆ [“Configuring the Server for TLS” on page 295](#)
- ◆ [“Configuring the Client for TLS” on page 296](#)
- ◆ [“Exporting the Trusted Root” on page 296](#)
- ◆ [“Authenticating with a Client Certificate” on page 297](#)

- ♦ “Using Certificate Authorities from Third-Party Providers” on page 297
- ♦ “Using SASL” on page 299


Requiring TLS for Simple Binds with Passwords

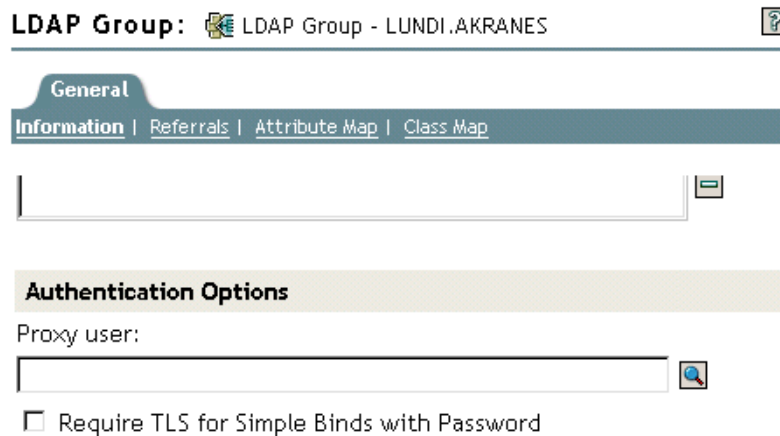
Secure Socket Layer (SSL) 3.1 was released through Netscape. IETF took ownership for that standard by implementing Transport Layer Security (TLS) 1.0.

TLS allows for connections to be encrypted in the Session layer. The encrypted port doesn't have to be used to get a TLS connection. There's another way: port 636 is the implied TLS port and the LDAP server automatically starts a TLS session when a client connects to the secure port.

A client can also connect to the clear-text port and later use TLS to upgrade the connection to an encrypted connection.

To require TLS for simple binds with passwords:

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click LDAP > LDAP Overview > View LDAP Groups.
- 3 Click the LDAP Group object, then click Information on the General tab.
- 4 Check the Require TLS for Simple Binds with Passwords check box.



LDAP Group: LDAP Group - LUNDI.AKRANES

General

Information | Referrals | Attribute Map | Class Map

Authentication Options

Proxy user:

Require TLS for Simple Binds with Password

- 5 Click Apply, then click OK.

Starting and Stopping TLS

The extended LDAP operation STARTTLS enables you to upgrade from a clear connection to an encrypted connection. This upgrade was new to eDirectory 8.7.

When you use the encrypted connection, the entire packet is encrypted. Therefore, sniffers are unable to diagnose data sent across the network.

Scenario: Using STARTTLS—You create a clear connection (to port 389) and do some anonymous searches. However, when you get into secure data, you prefer to start a TLS session. You issue a STARTTLS extended operation to upgrade from a clear connection to an encrypted connection. Your data is secure.

You stop TLS to turn an encrypted session into a clear connection. A clear connection requires less overhead because data to and from the client is not encrypted and decrypted. Therefore, data moves faster when you use a clear connection. At this point, the connection is downgraded to Anonymous.

When you authenticate, you use the LDAP Bind operation. Bind establishes your ID based on your provided credentials. When you stop TLS, the LDAP service removes any authentication previously established. Your authentication state changes to Anonymous. Therefore, if you want a state other than Anonymous you must reauthenticate.

Scenario: Reauthenticating—Henri runs STOPTLS. His status changes to Anonymous. To access and use his files on the Net, Henri runs the Bind command, provides his login credentials, is authenticated, and continues working in clear text on the Internet.

Configuring the Server for TLS

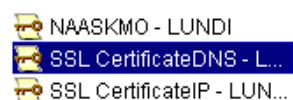
When a TLS session is instantiated, a handshake occurs. The server and the client exchange data. The server determines how the handshake occurs. To establish that the server is legitimate, the server always sends the server's certificate to the client. This handshake guarantees to the client that the server is indeed the expected server.

To require that the client also establish legitimacy, you set a value on the server. This attribute is `ldapTLSVerifyClientCertificate`.

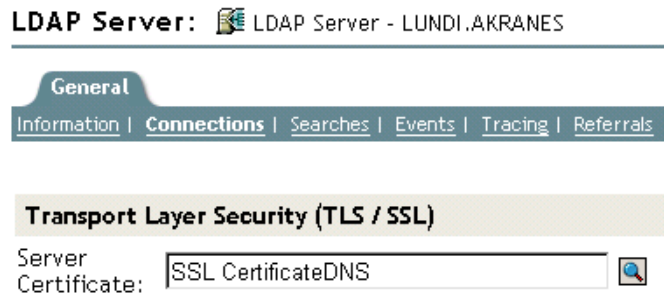
Value	Description
0	Off. During a handshake, the server provides a certificate to the client. The server never requires the client to send a certificate. The client can use or ignore the certificate. A secure session is established.
1	During the handshake, the server provides a certificate to the client and requests a certificate from the client. The client can choose to send its certificate back. The client's certificate is validated. If the server cannot validate the client's certificate, the connection is terminated. If the client doesn't send a certificate, the server maintains the connection.
2	During the handshake, the server requests and requires a certificate from the client. If the client does not provide a certificate, or if the certificate can't be validated, the connection is terminated.

Before the server can support TLS, you must provide the server with an X.509 certificate that the server can use to establish its legitimacy.

This certificate is automatically provided during the eDirectory installation. During installation, Key Material objects are created as part of Public Key Infrastructure (PKI) and Novell Modular Authentication Services (NMAS™). The following figure illustrates these objects in iManager:



The installation automatically associates one of those certificates with the LDAP server. In Novell iManager, the Connections tab for the LDAP Server object displays a DN. This DN represents the X.509 certificate. The Server Certificate field in the following figure illustrates this DN.



In Novell iManager, you can browse to the Key Material object (KMO) certificates. Using the drop-down list, you can change to a different certificate. Either the DNS or the IP certificate will work.

As part of the validation, the server should validate the name (the hard IP address or the DN) that is in the certificate.

To establish a TLS connection, ensure the following:

- ◆ The LDAP server must know the server's KMO
- ◆ You connect to the secure port or start TLS after connecting to the clear port

After you reconfigure the LDAP server, refresh the server. See [“Refreshing the LDAP Server”](#) on [page 292](#). ConsoleOne and Novell iManager automatically refresh the server.

Configuring the Client for TLS

An LDAP client is an application (for example, Netscape Communicator, Internet Explorer, or ICE). The client must understand the certificate authority that LDAP server uses.

When a server is added into an eDirectory tree, by default the installation creates

- ◆ A certificate authority for the tree (the tree CA).
- ◆ A KMO from the tree CA.

The LDAP server uses this certificate provider.

The client needs to import a certificate that the client will trust so that the client can validate the tree CA that the LDAP server claims to be using. The client must import a certificate from the server so that whenever the server sends its certificate, the client can validate it and verify that the server is who it claims to be.

So that the client can get a secure connection, the client must be configured before the connection.

The way that the client imports the certificate differs, based on the kind of application being used. Each application must have a method to import a certificate. Netscape browser has one way, IE has another way, and ICE has a third way. These are three different LDAP clients. Each client has its method for locating the certificates that it trusts.

Exporting the Trusted Root

You can automatically export the trusted root while accepting the certificate server.

To manually export the trusted root, see [Exporting a Trusted Root or Public Key Certificate \(http://www.novell.com/documentation/lg/crt27/crtadmin/data/a2ebopb.html#a2ebopd\)](http://www.novell.com/documentation/lg/crt27/crtadmin/data/a2ebopb.html#a2ebopd).

The Export functionality will create the specified file. Although you can modify the filename, it's a good idea to leave "DNS" or "IP" in the filename, so that you can recognize the type of material object. Also leave the servername.

Install the self-assigned CA in all browsers that establish secure LDAP connections to eDirectory.

If you are using the certificate with Microsoft products (for example, Internet Explorer), leave the .der extension.


If applications or SDKs require the certificate, import it into a certificate database.

Internet Explorer 5 exports root certificates automatically with a registry update. The traditional .X509 extension used by Microsoft is required.

Authenticating with a Client Certificate

Mutual Authentication requires a TLS session and a client certificate. Both the server and the client must verify that they are the objects that they claim to be. The client certificate was validated at the Transport layer. However, at the LDAP protocol layer, the client is anonymous until the client issues an LDAP bind request.

Up to this point, the client has proven its authenticity to the server but not to LDAP. If a client wants to authenticate as the identity contained in the client certificate, the client binds by using the SASL EXTERNAL mechanism.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click LDAP > LDAP Overview.
- 3 Click View LDAP Servers, then click the name of an LDAP Server object.
- 4 Click Connections.
- 5 In the Transport Layer Security section, select the drop-down menu for Client Certificate, then select Required.

This enables Mutual Authentication.

- 6 Click Apply, then click OK.

Using Certificate Authorities from Third-Party Providers

During the eDirectory installation, the LDAP server receives a tree Certificate Authority (CA). The LDAP Key Material object is based on that CA. Any certificate that a client sends to the LDAP server must be able to be validated through that tree CA.

LDAP Services for eDirectory 8.7.3 supports multiple certificate authorities. Novell's tree CA is just one certificate authority. The LDAP server might have other CAs (for example, from VeriSign*, an external company.) This additional CA is also a trusted root.

To configure the LDAP server to use multiple certificate authorities, set the ldapTLSTrustedRootContainer attribute on the LDAP Server object. By referencing multiple certificate authorities, the LDAP server allows a client to use a certificate from an external authority.

Creating and Using LDAP Proxy Users

Novell eDirectory assigns a [Public] identity to users who are not authenticated. In the LDAP protocol, an unauthenticated user is an Anonymous user. By default, the LDAP server grants Anonymous users the rights of the [Public] identity. These rights enable unauthenticated eDirectory and Anonymous LDAP users to browse eDirectory by using [Public] rights.

The LDAP server also allows Anonymous users to use the rights of a different proxy user. That value is located on the LDAP Group object. In Novell iManager, the value is named the Proxy User field. In ConsoleOne, the value is named the Proxy Username field. The following figure illustrates this field in Novell iManager.


The screenshot shows the configuration page for an LDAP Group in Novell iManager. At the top, it says "LDAP Group: LDAP Group - LUNDI.AKRANES" with a help icon. Below this is a tabbed interface with "General" selected. Under "General", there are sub-tabs: "Information", "Referrals", "Attribute Map", and "Class Map". The "Information" tab is active, showing a search bar. Below that, the "Authentication Options" section is expanded, showing a "Proxy user:" label followed by a search bar and a search icon. At the bottom of this section, there is a checkbox labeled "Require TLS for Simple Binds with Password" which is currently unchecked.

The proxy user is a Distinguished Name. You can grant that proxy identity different rights than the Public identity has. With the proxy user, you can control LDAP Anonymous access to specific containers in the eDirectory tree.

NOTE: Don't set login restrictions for the proxy user unless you want to have them apply to all Anonymous LDAP users.

Scenario: Setting Up an NLDAP Proxy User—Digital Airlines has contracted with DataSure, a research group. DataSure will use LDAP to access and store research on DigitalAir43, a NetWare 6 server at Digital Airlines. You don't want DataSure to have Public rights to directories on DigitalAir43.

Therefore, you create an LDAP proxy user and assign that user specific rights to the DataSure directory. You populate the proxy Distinguished Name on the LDAP Group object and refresh the server. The server automatically starts using the proxy user rights for any new or existing Anonymous users.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Administration > Create Object, then create a proxy user (for example, LDAPPProxy).
- 3 Assign a null password to that user.
- 4 (Optional) Assign the proxy user rights to specified directories.
- 5 Click LDAP > LDAP Overview > View LDAP Groups > the LDAP Group object.
- 6 In the Proxy User field, click the Browse button, browse to and select the LDAPPProxy user, then click OK.

Using SASL

Simple Authentication and Security Layer (SASL) defines various authentication mechanisms that must be registered with the Internet Assigned Numbers Authority (IANA). The LDAP server supports the following mechanisms:

- ◆ DIGEST-MD5
- ◆ EXTERNAL
- ◆ NMAS_LOGIN

These mechanisms are installed on the server during an eDirectory installation or upgrade. However, on UNIX, you have to run the `nmasinst` utility to install NMAS methods.

The LDAP server queries SASL for the installed mechanisms when it gets its configuration, and automatically supports whatever is installed. The LDAP server also reports the current supported SASL mechanisms in its rootDSE by using the `supportedSASLMechanisms` attribute.

Because these mechanisms are registered, you must enter them using all uppercase characters. Otherwise, the LDAP server won't recognize them.

The LDAP bind protocol allows the client to use various SASL mechanisms for authentication. When the application uses the LDAP bind API, it would either need to choose the simple bind and supply a DN and password, or choose the SASL bind and supply the SASL mechanism name in upper case, and any associated SASL credentials required by the mechanism.

DIGEST-MD5

The DIGEST-MD5 mechanism does not require TLS. The LDAP server supports DIGEST-MD5 over clear and secure connections.

LDAP supports SASL mechanisms in the bind request. Instead of requesting an LDAP simple bind (DN and clear-text password), you request an LDAP SASL bind. This request provides a DN and MD5 credentials.

MD5 provides an encrypted hash of passwords. Passwords are encrypted even on clear connections. Therefore, the LDAP server accepts passwords that use MD5 on either the clear-text or encrypted port.

If someone sniffs this connection, the password can't be detected. However, the entire connection can be spoofed or hijacked.

This mechanism is an LDAP SASL bind (and not a simple bind). Therefore, the LDAP server accepts these requests, even if you checked the Require TLS for Simple Binds with Passwords check box during installation.

EXTERNAL

The EXTERNAL mechanism informs the LDAP server that a user's DN and credentials have already been supplied to the server. Therefore, the DN and credentials don't need to come across in the bind request.

The LDAP bind request using the SASL EXTERNAL mechanism instructs the server to do the following:

- ◆ Ask an EXTERNAL layer what the credentials were
- ◆ Authenticate the user as those credentials and user

A secure handshake has occurred. The server requested credentials from the client and the client passed them to the server. The LDAP server received the certificate that was passed from the client, passed the certificate to the NMAS module, and authenticated the user as whatever DN was supplied in the certificate.

Having a certificate with a usable DN requires some setup on the client. For information about setting up the certificate, see the [NMAS \(http://www.novell.com/documentation/lg/nmas23/index.html\)](http://www.novell.com/documentation/lg/nmas23/index.html) online documentation.

Even if the client sends an EXTERNAL mechanism, the LDAP server could fail the request. Novell iMonitor can provide the reasons for failure:

- ◆ The connection is not secure.
- ◆ Although the connection is secure, the client did not provide the required certificate during the handshake.
- ◆ The SASL module is unavailable.
- ◆ The client failed to check the rootDSE before sending the request.

NMAS_LOGIN

The NMAS_LOGIN mechanism provides the LDAP server with the biometrics capability of NMAS. For more information, see the Novell NDK.

When the server comes up, the LDAP server initializes with the SASL module and then asks the SASL module which mechanisms are available to the LDAP server.

The client can query the rootDSE to find out the supported attribute for the mechanism. Then the LDAP server displays the supported mechanisms.

Using the LDAP Server to Search the Directory

This section contains information on the following:

- ◆ “Setting Search Limits” on page 300
- ◆ “Using Referrals” on page 301
- ◆ “Searching Filtered Replicas” on page 306

Setting Search Limits

The following attributes on the LDAP Server object control how the LDAP server searches the Directory:

- ◆ Search Entry Limit



Limits the size of a search. The default is 0, for no limit on size. So that the LDAP server isn't overloaded, you can limit the number of entries that the LDAP server returns from a search request.

Scenario: Limiting the Size of a Search—Henri requests a search that could result in thousands of replies concerning objects that the search finds. However, you have set a limit of 10 results. LDAP Server stops searching after returning 10 results. A system message informs Henri that the search has ended even though more data is available.

- ◆ Search Time Limit

Limits the time that the server searches. The default is 0 seconds, for no time limit.

The following figure illustrates these attributes in Novell iManager.

LDAP Server:  LDAP Server - LUNDI.AKRANES 

General

Information | Connections | **Searches** | Events | Tracing | Referrals


Maximum concurrent persistent searches: operations ('0' for unlimited)

Ignore size and time limits when monitoring persistent search events

Restrictions

Entry Limit: entries ('0' for no limit)

Time Limit: seconds ('0' for no timeout)

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click LDAP > LDAP Overview > View LDAP Servers.
- 3** Click the LDAP Server object > Searches.
- 4** Scroll to the Restrictions section, enter values, then click OK.

The client can also set limit search requests (for example, limiting the search to two seconds). If the client limit conflicts with the server limit, the LDAP server uses the lowest or smallest value from either request.

The search is based on Access Control Lists (ACLs). Therefore, an Anonymous search could yield the few entries that Public is allowed to view, even though thousands of entries exist in the Directory.

Using Referrals

A referral is a client-centric method to resolve names. An LDAP client sends a request to an LDAP server, which attempts to find the target entry of the operation locally. If the server can't find the target entry, the server uses the knowledge references that it has to generate a referral to a second LDAP server that knows more about the entry. The first server sends the referral information to the LDAP client.

The LDAP client then establishes a connection to the second LDAP server and retries the operation. If the second LDAP server has the target entry of the operation, it performs the operation. Otherwise, the second server also sends a referral back to the client. This process continues until one of the following occurs:

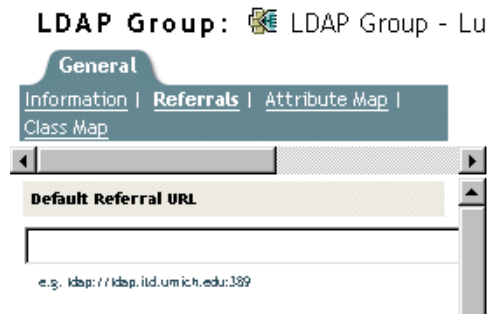
- ◆ The client contacts a server that has the entry and can perform the desired operation
- ◆ The LDAP server returns an error indicating that the entry doesn't exist
- ◆ The LDAP server indicates that no more referrals can be followed

Functionality introduced in LDAP for eDirectory 8.7 causes referrals to behave slightly differently than with earlier versions of eDirectory and NDS. The differences influence the way you configure LDAP Services.

Default Referrals

Typically, a default referral URL contains an LDAP URL that points to a server that holds the root of the tree. An LDAP URL has the following form: `ldap://host:port`.

You enter a default referral in the Default Referral URL field:



Historically, the eDirectory LDAP server sent the default referral in a number of failover situations. Many users find these behaviors strange and sometimes unpredictable. LDAP Services for eDirectory 8.7.3 let you control when the default referral is sent for any kind of subordinate referral.

The new option is a value (setting) held on the `ldapDefaultReferralBehavior` attribute on the LDAP Server and LDAP Group objects. The value is an integer which is a bitmask of the following bits.

Bits	Value
0x00000001	The base DN is not found
0x00000002	The base DN is on an unavailable eDirectory server
0x00000004	An entry in the search scope is on an unavailable eDirectory server

If the LDAP server is configured to Always Refer for the operation, and if any of the conditions listed are met and the corresponding value is set, the default referral is returned.

Setting Referrals for Search Operations

Functionality introduced to LDAP for eDirectory 8.7 causes referrals to behave slightly differently than with earlier versions of eDirectory and NDS. The differences influence the way you configure LDAP Services.

You can configure the eDirectory LDAP server to return referrals to other eDirectory servers within the eDirectory tree. By default, the LDAP server chains all operations to other eDirectory servers on behalf of the user, and referrals are never returned.

Prior to eDirectory 8.7, the referral options only existed as settings on the LDAP Group object. eDirectory 8.7.3 allows you to set these options on the LDAP Server object also. Any setting on the LDAP Server object overrides that setting on the LDAP Group object.

You set the Referral Option by manipulating the `ldapSearchReferralOption` attribute. Previous to LDAP Services for eDirectory 8.7, you could set this attribute to the following options:

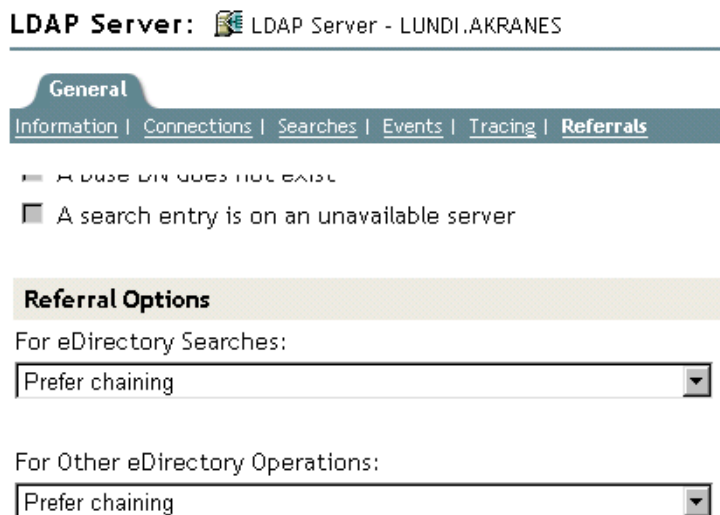
- ◆ [“Prefer Chaining” on page 304](#) (the default option)

- ♦ “Prefer Referrals” on page 304
- ♦ “Always Refer” on page 305

These referral options apply only to referring and chaining to other eDirectory servers within the eDirectory tree. These configuration settings don't control referrals that come from a nonauthoritative partition. Therefore, even though you select an option (for example, Always Chain) from the Referral Options drop-down list, referrals will still come from nonauthoritative partitions to other servers.

To support superior referrals to non-eDirectory DSAs, LDAP Services for eDirectory 8.7.a has an Always Chain option. See “Always Chain” on page 303.

The following figure illustrates the LDAP referral drop-down lists for searches and other operations.



“Other” eDirectory operations include referrals for the Add, Delete, Modify, and Bind operations.

Always Chain

The Always Chain option is a “never refer” option. If you select this option, the eDirectory LDAP server never returns referrals to other eDirectory servers in the eDirectory tree. The LDAP server checks with other LDAP servers on behalf of the requesting client and sends the referral to the client.

The Always Chain option will be most beneficial to you if you have an eDirectory deployment that participates as subordinate servers in a global federated tree.

These referral options only apply to the way referrals are handled within the eDirectory tree. They have no bearing on referral behavior to non-eDirectory servers.

The reason for blocking referrals to other eDirectory servers is subtle, but may prove invaluable. If the nonauthoritative data on an eDirectory 8.7 or later server is replicated to another, older eDirectory server, a referral to the older server might cause a client application to get a distorted view of the global tree.

For example, assume that an LDAP client caches referrals to LDAP servers and sends requests to the server it last communicated with. If the client is configured to send requests to an eDirectory server that supports superior referrals, the client's view of the global tree should be normal.

However, LDAP servers earlier than eDirectory 8.7 don't understand nonauthoritative areas and superior referrals. Therefore, if the client follows a referral to an earlier-version eDirectory server in the eDirectory tree, and continues to send requests to that earlier-version server, the earlier-version LDAP server will present the nonauthoritative data as if it were the actual directory tree data.

An intelligent client should, however, interrogate the supportedFeatures attribute of the RootDSE to ascertain whether or not the server supports superior referrals.

Prefer Chaining

The Prefer Chaining option indicates that search operations will not normally return referrals. Instead, the LDAP server progresses the search operation across all eDirectory DSAs required to complete it.

The exception is a search operation that is accompanied by the persistent search control. In this case, because the Novell implementation of persistent search does not support chaining, referrals are sent if the scope of the search operation is not all held locally.

The LDAP server receives a search operation. If the entry in the tree is not stored locally, the server automatically chains to other servers. After the entry has been located, the LDAP server acts as proxy for the LDAP client. Using the same identify that the LDAP client is bound with, the LDAP server authenticates to the remote server and continues the search operation there.

The LDAP server that received the original search request sends the LDAP client all search entries and the search result. Because the LDAP server fully takes care of the request, the LDAP client is unaware that other servers were involved.

Through chaining on eDirectory, an LDAP server that doesn't have much data can appear to hold the data of the entire tree.

Prefer Chaining is important concerning partitions.

Scenario: Finding Information in another Partition—At the Digital Airlines Company, Luc selects the Prefer Chaining option for LDAP server DAir43. DAir43 is in Partition A. Partition B is a subpartition of A and contains LDAP server DAir44.

An LDAP client requests a search. DAir43 searches locally for the entry but only finds part of the data. DAir43 automatically chains to DigitalAir44, which has the needed entry. DAir44 sends the data to DAir43, and DAir43 sends the entry to the LDAP client.

The Prefer Chaining option causes the LDAP server to chain to other servers for search requests (when needed) unless the operation is a Persistent Search. For information on Persistent Search, see [“Persistent Search: Configuring for eDirectory Events” on page 311](#).

Prefer Referrals

The Prefer Referrals options indicates that search operations will return referrals to other eDirectory servers in the eDirectory tree when needed. Referrals are sent only if the local server can ensure that the server holding the data is operational and that the LDAP service is running. Otherwise, the operation is chained to the other server, or the operation fails if the other server is inoperable.

You have two partitions and are doing a subtree search. You get down to a point where the search entries are no longer held on the local server. Therefore, the search must go to another server. If the server that holds the replica of that data (that partition) is also running nldap.nlm, the LDAP server builds an LDAP referral and sends it back to the LDAP client.

If the server holding the replica isn't running `nldap.nlm`, LDAP server chains the request to the other server, thereby completing the search.

When `nldap.nlm` starts up, the LDAP server communicates to eDirectory that the LDAP server is a referral point. If a client has received referrals but the referrals stop, the LDAP server is not running.

Always Refer

The Always Refer option follows the same logic as Prefer Referrals, except that the Default Referral is sent under various failover situations (for example, an object is not found or the server is down).

If another server that holds the rest of the data isn't running the LDAP service, the first LDAP server won't chain the request to the second server.

If you mark the Always Refer option, you are allowed to enter a default referral. The Default Referral field enables you to glue two different vendor LDAP servers together and build your own Directory tree.

Scenario: Using a Default Server—You have an LDAP tree. One part of the tree is serviced by eDirectory. A subordinate partition is serviced by iPlanet. In the Default Referral field, you place a URL that references the iPlanet server. An LDAP client requests a search.

Unable to resolve the base DN, the LDAP server sends the client the string in the Default Referral field. The referral instructs the LDAP client to look in the place specified in the URL. The LDAP client contacts the iPlanet server, which completes the search.

Whenever a default referral is configured and the server doesn't find the base DN being searched for, the client receives the default referral.

The format for a referral is an LDAP URL (for example, `LDAP://123.23.45.6:389`).

When the LDAP server sends a default referral to a client (because the base DN was unavailable), the server appends an additional forward slash (/) and the DN that the client was looking for. The default referral and the appended information go to the client. The client sends the search request to the server specified in the default referral.

The LDAP Group object has a string field for the default referral. The LDAP server treats that data as a string. There is no validation. Whatever is entered is prepended to the referral. Some data is appended to the referral. The LDAP server expects the string to look like a URL.

When clients get referrals to other eDirectory servers that are running LDAP, the client receives two referrals per server:

- ◆ A referral directing the client to the clear-text port
- ◆ A referral directing the client to the secure port

To differentiate between the two referrals, the clear-text referral states `ldap://` and the secure port displays `ldaps//`.

A referral from the server appends the port number.

Setting Referrals for Other Operations

The historical referral option setting only applied to the search operation. To provide a comparable option for other operations, the `ldapOtherReferralOption` attribute is used. This attribute allows the same values and controls the behavior for non-search operations (excluding bind, which never sends a referral).

No Support for ManageDsaIT

In LDAP Services for eDirectory 8.7.3, the distributed relationships between eDirectory servers in an eDirectory tree are managed by means other than the use of the ManageDsaIT control. The ManageDsaIT control won't allow the LDAP client to interrogate or update eDirectory subordinate or cross references.

Functionality Not Supported

LDAP Services for eDirectory 8.7.3 doesn't support subordinate references. You cannot reliably create a nonauthoritative partition that is subordinate to an authoritative partition and have it send referrals. If you elect to do this, referrals are only sent when resolving the base DN for an operation. SearchResultReferences are not sent.

There is no support for distributed updates of data in the nonauthoritative area. If a name change occurs on the root server, there is no built-in mechanism to copy that name change to the eDirectory server holding that same data in a nonauthoritative area.

Searching Filtered Replicas

A filter restricts the amount of data that the replica holds. Therefore, a filtered replica does not have complete view of real data held in the directory. The following are examples of filters applied to a replica:


- ◆ The replica only contains User objects.
- ◆ The replica contains all User objects, but the objects only contain telephone numbers and mailing addresses.

Because data in a filtered replica is incomplete, an LDAP search could produce constrained results. Therefore, by default an LDAP search request does not examine filtered replicas.

While performing filtered replica search, the search might not return the results as per the replica filter in the following cases:

- ◆ If the objects matching the search filter are not present on the local filtered replica server then the results may not match with the filter of the local replica as the results may be fetched from a full replica server.
- ◆ When the search base is not local to the filtered replica server, the objects matching the search filter may be obtained from a full replica server and these might not match with the filter of the local replica.

However, if you are certain that a filtered replica holds data that you need, you can configure an LDAP server to search filtered replicas.

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click LDAP > LDAP Overview.
- 3** Click View LDAP Server, then click the name of an LDAP server.
- 4** Click Searches.
- 5** Select Include Filtered Replicas in Search, then click Apply.

LDAP Server: LDAP Server - Server1-2000-NDS

General

Information | Connections | **Searches** | Events | Tracing | Referrals

Filtered Replica

Include filtered replicas in search

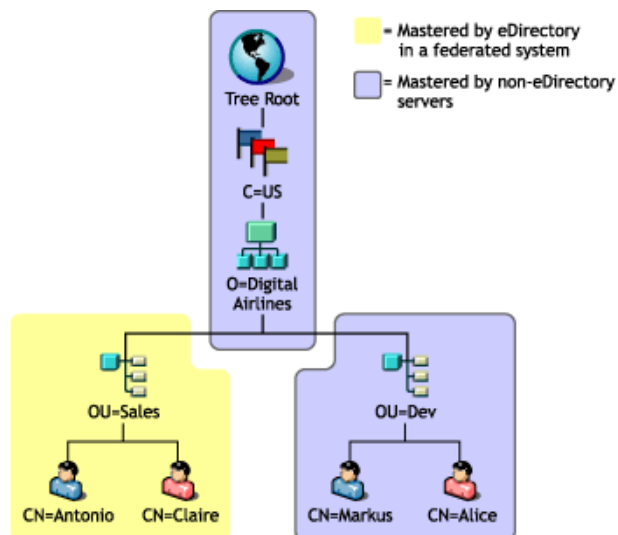
Configuring for Superior Referrals

Often, larger deployments need a directory tree that uses LDAP server software from different vendors. Such a tree is a global federated tree. LDAP Services for eDirectory 8.7.3 has the capability to return referrals to a superior DSA in a federated tree.

Scenario: Superior Referrals in a Federated Tree

Luc is responsible for networks at Digital Airlines. An OpenLDAP server is being used to master the root of a directory tree at Digital Airlines (from the tree root down to O=Digital Airlines). An organization (OU=Sales) is mastered by an eDirectory server, and another organization (OU=Dev) is held on an iPlanet server.

The following figure illustrates this tree:



eDirectory masters only the data within the partition for OU=Sales. The data in the other areas are mastered on non-eDirectory DSAs. Luc configures LDAP Services to return superior referrals whenever an operation is rooted at O=Digital Airlines or above, or anywhere under O=Digital Airlines that is not part of the OU=Sales hierarchy.

An operation is sent to the eDirectory LDAP server with a base DN of OU=Dev,O=Digital Airlines,C=US. A referral is returned pointing to the servers holding that entry or to servers that have knowledge of the servers holding that entry.

Likewise, a subtree search rooted at O=Digital Airlines,C=US results in a referral to the root DSA. The root DSA in turn returns referrals to the DSAs mastering OU=Sales and OU=Dev.

So that the eDirectory server can participate in this tree, LDAP Services allows eDirectory to hold the hierarchical data above it in a partition marked “nonauthoritative.” The objects in the nonauthoritative area consist only of those entries needed to build the correct DN hierarchy. These entries are analogous to X.500 “Glue” entries.

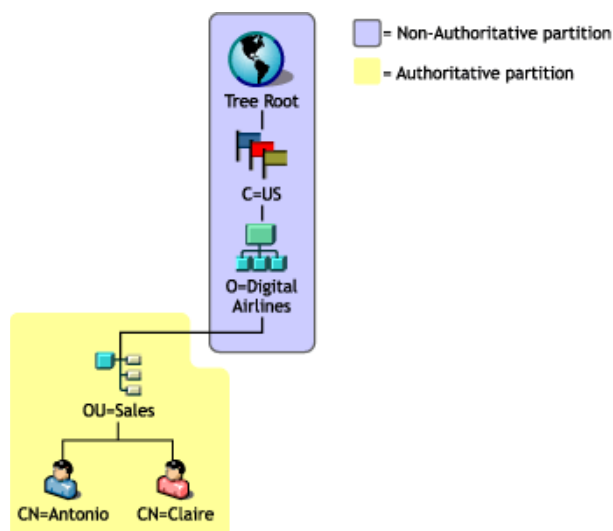
In this scenario, the Root, C=US, and O=Digital Airlines objects are held on the eDirectory server in a nonauthoritative area.

eDirectory allows knowledge information (referral data) to be placed within nonauthoritative areas. This information is used to return referrals to the LDAP client.

When an LDAP operation takes place in a nonauthoritative area of the eDirectory tree, the LDAP server locates the correct reference data and returns a referral to the client.

Creating a Nonauthoritative Area

The following figure illustrates the actual data held on the eDirectory server in the federated tree shown in “[Scenario: Superior Referrals in a Federated Tree](#)” on page 307.



Notice that entries are placed above OU=Sales, even though these entries are mastered by another DSA. This placement is necessary to provide the proper DNs for the entries mastered by the eDirectory server.

To create a nonauthoritative area:

- 1** Segregate the nonauthoritative data from the authoritative data.
 - Create a partition boundary at the top of the authoritative area. An eDirectory server considers itself authoritative for all data that it holds unless otherwise specified.
- 2** Mark the root partition as nonauthoritative.
 - 2a** Add the authoritative attribute to the rootmost entry in the partition.
 - 2b** Populate the authoritative attribute with a value of zero.
- 3** Draw a boundary at the bottom of the nonauthoritative area.

Create partition roots at the areas of the subtree that this server is to be authoritative for. For example, in the figure above, a partition root exists at the OU=Sales entry. The new partitions won't have the authoritative attribute set to zero. Therefore, the server will be authoritative for the partitions.

4 Refresh the LDAP server.

The LDAP server caches the authoritative and nonauthoritative area boundaries whenever its configuration is refreshed. If you don't manually refresh the server configuration, the server will automatically refresh itself on a 30-minute background task.

Multiple partitions can be stacked in a chain of nonauthoritative areas. However, LDAP Services for eDirectory 8.7.3 requires that all nonauthoritative partitions must be contiguous and held in local replicas.

Specifying Reference Data

When the LDAP server finds that an operation is taking place in a nonauthoritative area, it looks for information it can use to return a referral to the client. This referral information might be at one of the following:

- ◆ Located on any or all of the entries in the nonauthoritative area
- ◆ Specified as a default referral on the LDAP Server or LDAP Group object that holds the configuration data for the server

Referral information held on entries in the nonauthoritative area is an Immediate Superior Reference. Such referral information consists of a multi-valued ref attribute. (For a description of this attribute, see [RFC 3296 \(http://www.ietf.org/rfc/rfc3296.txt\)](http://www.ietf.org/rfc/rfc3296.txt). Referral information held in the Default Referral configuration setting is a Superior Reference and is single-valued. (See immSupr and supr DSE types in X.501.)

Reference data is held in the form of an LDAP URL, but only specifies the host and (optionally) the port of the DSA being referred to. The following example illustrates this reference data:

```
ldap://ldap.digital_airlines.com:389
```

The LDAP server looks at the base DN for the operation (or if not found, the matched DN). If the base DN contains reference information, the LDAP server returns that information as a referral.

If no reference information is found, the LDAP server traverses the tree upwards, looking for reference information. If no reference information is found after exhausting all entries, the LDAP server returns the superior reference. (This reference is held in the default referral setting on the LDAP Group or LDAP Server object.)

Adding an Immediate Superior Reference

You can add an auxiliary object class called `immediateSuperiorReference` to an entry in the nonauthoritative area. This auxiliary class adds a `ref` attribute, which is populated with one or more LDAP URLs. Each URL points to a DSA's host name and (optionally) port.

Adding a Superior Reference

Historically, the LDAP Group object has had an `ldapReferral` attribute. This attribute held a default reference that was used for various failover situations when returning referrals to other eDirectory servers in an eDirectory tree. In LDAP Services for eDirectory 8.7.3, this attribute is used to hold a single default referral to a superior DSA in a federated tree.

Additionally, the `ldapReferral` attribute has been added to the LDAP Server object. If the `ldapReferral` attribute contains a value on the LDAP Server object, that setting overrides the value held in the same attribute on the LDAP Group object. This behavior allows you to configure all LDAP servers participating in a group to have a particular default referral, while one or two servers override that value with a different default referral.

The value on the `ldapReferral` attribute is an LDAP URL. The URL holds the host and optional port of the DSA being referred to.

Updating Reference Information through LDAP

If you followed the steps above, in order, and used LDAP to perform the tasks, you were likely unable to add an immediate superior reference. This is because the root partition had already been marked nonauthoritative, so LDAP sends referrals for any operation acting on data within that partition.

To update or interrogate information in a nonauthoritative area, the `ManageDsaIT` control must accompany the LDAP request. For information on this control, see [RFC 3296 \(http://www.ietf.org/rfc/rfc3296.txt\)](http://www.ietf.org/rfc/rfc3296.txt). This control effectively causes the LDAP server to treat the entire nonauthoritative area as though it is authoritative.

NOTE: The superior reference feature is only available through LDAP. Other protocols (for example, NDAP) are not affected by the presence of the authoritative attribute. Therefore, the use of `ConsoleOne` or `Novell iManager` to interrogate and update data in the nonauthoritative area is unhindered.

Affected Operations

Nonauthoritative areas and superior referrals affect the following LDAP operations:

- ◆ Search and Compare
- ◆ Modify and Add

DN-syntax attribute values are not checked. Therefore, a group member attribute can contain DNs that point to entries in a nonauthoritative area.

- ◆ Delete
- ◆ Rename (`moddn`)
- ◆ Move (`moddn`)

If the parent DN falls within a nonauthoritative area, an error affects `MultipleDSAs` should be returned.

- ◆ Extended

Discovering Support for Superior References

Support for superior referrals is available only in LDAP Services for eDirectory 8.7 and later. To discover whether an eDirectory server supports this functionality, you can read the `supportedFeatures` attribute on the root DSE. If the `supportedFeatures` attribute lists the OID 2.16.840.1.113719.1.27.99.1, these features are available. Additional discovery-related changes to the root DSE object include the following:

- ◆ `namingContexts`

This attribute only lists the partition roots held on the local DSA that the server is authoritative for. No nonauthoritative partition roots are listed.

- ◆ altServer
This attribute won't list other eDirectory servers that share only nonauthoritative partitions with the local server.
- ◆ superiorReference
This attribute advertises the superior referral for the DSA. This value is administered by updating the ldapReferral attribute on the LDAP Server or LDAP Group object.

Persistent Search: Configuring for eDirectory Events

Novell eDirectory has an event service that enables applications to be notified of significant events that occur within the Directory. Some of these events are general events that can pertain to any Directory service. Other events are specific to eDirectory and its special features.

eDirectory events are exposed to applications through two different extensions to the LDAP protocol:

- ◆ An implementation of the Persistent Search Control

The Persistent Search feature of Novell eDirectory is a search operation that keeps going after the initial set of matching entries is returned. Persistent Search is an extension to the LDAP v3 search operation that moves the burden of checking for updates within a search result set from the client to the server. The Persistent Search control allows the client to perform a normal LDAP search operation (specifying the base DN, scope of search, search filter, and so on) and then, rather than having the server return a SearchResultDone message at the end, the operation maintains a connection so the client can be updated each time an entry in the result set changes. This allows the client to maintain a cache of the entries it is interested in, or trigger some logic whenever an update occurs.

The “Persistent Search” document on the Internet (<http://www.ietf.org/proceedings/01mar/I-D/ldapext-psearch-03.txt>) describes this extension in further detail.

- ◆ Monitor Events (an extended LDAP operation that is specific to eDirectory)



Applications that use eDirectory event services can place a heavy computational load on the directory. Various administrative parameters are available to help control how event services are used on individual eDirectory servers. These parameters are stored on the LDAP Server object. You use ConsoleOne or Novell iManager to set these parameters.

Specific applications that use the event service might require that you set these parameters to specific values. The documentation for such applications will indicate specific requirements for the application.

For more information, see [Understanding and Using Persistent Search in Novell eDirectory \(http://developer.novell.com/research/appnotes/2003/february/04/a030204.htm\)](http://developer.novell.com/research/appnotes/2003/february/04/a030204.htm).

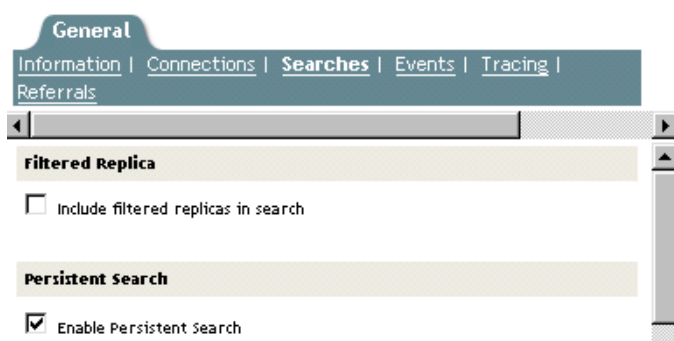
Managing Persistent Searches

You can use Novell iManager to view or edit persistent searches.

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Administration > Modify Object.
- 3** Enter the name and context of the LDAP Server object you want to modify, or click  and browse or search for the LDAP Server object.



- Click OK, then click Searches on the General tab.



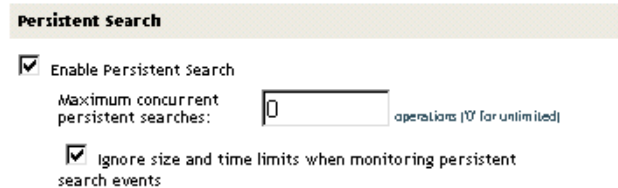
- Enable persistent searches.

By default, the Enable Persistent Search check box is checked. To disable and prevent persistent searches on this server, uncheck the check box.

NOTE: If you disable a previously established persistent search operation, the operation might continue even after this option is disabled and the server is refreshed.

- Control the number of concurrent persistent searches on this server.

Specify a value in the Maximum Concurrent Persistent Searches field. A value of zero allows unlimited concurrent persistent searches.




- Control whether to ignore size and time limits.

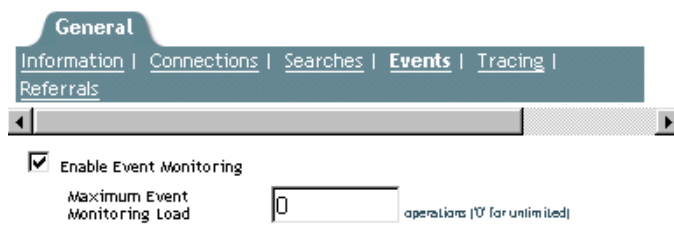
To control whether size and time limits should be ignored after the persistent search request has sent the initial search result set, check the Ignore Size and Time Limits When Monitoring Persistent Search Events check box.

If you don't select this option, the entire persistent search operation is subject to the search restrictions. If either limit is reached, the search will fail, with the appropriate error message.

- Click Apply, then click OK.

Controlling Use of the Monitor Events Extended Operation

- In Novell iManager, click the Roles and Tasks button .
- Click LDAP > LDAP Overview.
- Click View LDAP Servers, then click the name of an LDAP server.
- Click Events.



5 Control whether client applications can monitor events on this LDAP server.

To enable client applications to monitor events on this LDAP server, check the Enable Event Monitoring check box.

To disable the monitoring of events, uncheck the check box.

6 Control the maximum load that event monitoring applications can place on the server.

Enter a value in the Maximum Event Monitoring Load field.

Processing event data and sending event notifications to monitoring applications involves computational overhead on the LDAP server. For a given event, the exact load on the server depends on the frequency of the event being monitored, the data associated with the event, and the number of client applications monitoring the event.

The Maximum Event Monitoring Load is a relative value that reflects how much of a load the event monitoring extension is allowed to place on the server. A zero value indicates no limit. To find an appropriate value for this attribute, experiment.

7 Click Apply, then click OK.

Getting Information about the LDAP Server

To get information about an LDAP server, you use ICE or an LDAP search. These utilities request information from rootDSE (Directory Service Agent, specific entry).

RootDSE is a pseudo object in a directory tree. It is an unnamed entry at the root of the tree. RootDSE holds information that is specific to the server that you are connected to. For example, rootDSE knows where the schema is located and the extensions and controls that the schema supports.

Because rootDSE is not a named entry in the tree, an LDAP server does not return rootDSE to the client as part of any normal search operation.

The following table lists information from rootDSE.

Information and Description	Excerpt
The schema's location: You find where the schema for the LDAP server or tree is located by reading the subschemaSubentry. For eDirectory, cn=schema is the base for the search.	subschemaSubentry: cn=schema

Information and Description	Excerpt
Supported extensions: Extensions enable you to manage the server (for example, creating or merging contexts, adding new replicas, refreshing the LDAP server, removing replicas, changing the replica type from master to read/write or read-only) and identities. Extensions are in ASN.1OID format. For names of extensions, see LDAP Extensions (http://developer.novell.com/ndk/doc/ldapover/ldap_enu/data/a6ik7oi.html) .	supportedExtension: 2.16.840.1.113719.1.27.100.12 supportedExtension: 2.16.840.1.113719.1.27.100.7 supportedExtension: 2.16.840.1.113719.1.27.100.8
Which vendor is providing the LDAP server.	vendorName: Novell, Inc.
Which directory version the LDAP server supports.	vendorVersion: eDirectory v8.7.0 (10410.29)
Which version of eDirectory is running.	vendorVersion: eDirectory v8.7.0 (10410.29)
The directory server name and the directory tree name.	dsaName: cn=WestWindNDS,o=westwind directoryTreeName: t=WESTWINDTREE
Supported SASL mechanisms.	supported SASLMechanisms: EXTERNAL supported SASLMechanisms: DIGEST-MD5 supported SASLMechanisms: NMAS LOGIN
Which version of LDAP Server is supported.	supportedLDAPVersion: 2 supportedLDAPVersion: 3
Server statistics: RootDSE provides a variety of statistics about the LDAP server (for example, the number of strong authentication binds).	errors: 0 securityErrors: 0 chainings: 3 referralsReturned: 6 extendedOps: 0 abandonOps: 0 wholeSubtreeSearchOps: 1

Information from rootDSE is useful for application developers.

Scenario: Developing an Application—Henri is writing an application that creates a new replica. Henri reads rootDSE and finds supportedExtension: 2.16.840.1.113719.1.27.100.7 in the list. Henri knows that the server supports the call to create a new replica.

Also, Novell iManager checks to see what functionality is available in rootDSE and then behaves according to that information.

To search rootDSE, enter the following at a workstation:

```
ldapsearch -h hostname -p 389 -b "" -s base "objectclass=*"
```

This search can be performed by any application using the ldap_search APIs.

The key to the search is that the base is null and the filter is set to objectclass=*. (In the case of this client, the base is -b.)

For more information on reading the rootDSE, refer to one of the following:

- ♦ [LDAP Libraries for C \(http://developer.novell.com/ndk/doc/cldap/ldaplibc/data/hevgtl7k.html\)](http://developer.novell.com/ndk/doc/cldap/ldaplibc/data/hevgtl7k.html)

- ◆ [LDAP Classes for Java \(http://developer.novell.com/ndk/doc/jldap/jldapenu/data/hevgtl7k.html\)](http://developer.novell.com/ndk/doc/jldap/jldapenu/data/hevgtl7k.html)

For information on LDAP search filters, see [LDAP Search Filters \(http://developer.novell.com/ndk/doc/ldapover/ldap_enu/data/a3saoeg.html\)](http://developer.novell.com/ndk/doc/ldapover/ldap_enu/data/a3saoeg.html). This section is in the LDAP and NDS Integration section of the NDK documentation.

13 SNMP Support for Novell eDirectory

The Simple Network Management Protocol (SNMP) is the standard operations and maintenance protocol for the Internet for exchanging management information between the management console applications and managed devices. Management console application are application such as HP* Openview, Novell® NMS, IBM* NetView, or Sun* Net Manager. The managed devices includes hosts, routers, bridges, and hubs and also network applications like Novell eDirectory™.

This section describes SNMP services for Novell eDirectory 8.7.3. It contains the following topics:

- ♦ [“Definitions and Terminology for SNMP” on page 317](#)
- ♦ [“Understanding SNMP Services” on page 318](#)
- ♦ [“eDirectory and SNMP” on page 320](#)
- ♦ [“Installing and Configuring SNMP Services for eDirectory” on page 323](#)
- ♦ [“Monitoring eDirectory Using SNMP” on page 336](#)
- ♦ [“Troubleshooting” on page 363](#)

Definitions and Terminology for SNMP

The following tables contain terminologies used in this chapter.

Terminology	Definition
EMANATE	Enhanced Management Agent Through Extensions is a product from SNMP Research International, Inc.
SNMP	Simple Network Management Protocol is used to exchange data about the network activity.
NAA	Native Agent Adapter
NMS	Network Management Station
MA	Management Agent
SA	Subagent
MIB	Management Information Base
NCP™	NetWare® Core Protocol™
NMA	Network Management Application
edir.mib	Novell eDirectory server Monitoring MIB, which has MIB objects and traps relevant to Novell eDirectory.

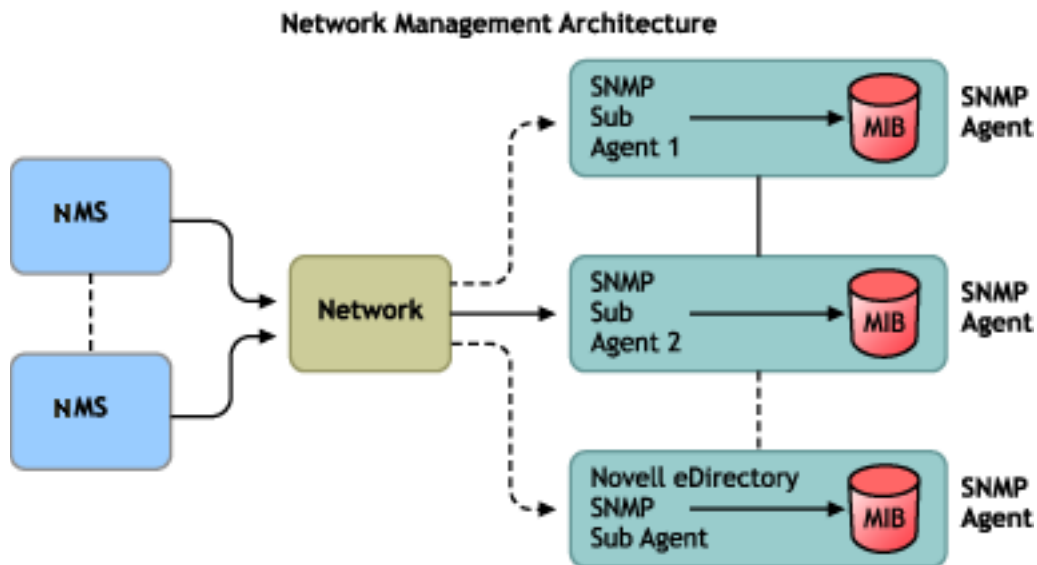
Terminology	Definition
traps	Alerts generated by agents on a managed device when eDirectory events occur on the server. These conditions are defined in the Management Information Base (MIB) provided by Novell.

Understanding SNMP Services

SNMP is based on a manager/agent architecture. The architecture of network management with SNMP includes the following elements:

- ◆ Network Management Station (NMS)
- ◆ Managed Device
- ◆ Master Agent
- ◆ Subagent
- ◆ Management Information Base (MIB)
- ◆ Network Management Protocol

Figure 33 Network Management Architecture



Network Management Station

The network management station is a workstation with one or more network management applications installed, to graphically show information about managed devices.

NMS features:

- ◆ Provides the user interface to the entire network management system, thus providing a powerful, flexible and easy to use tool for network management
- ◆ Allows you to perform SNMP Get, Get Next, SNMP Get Response and Set operations. NMS also allows you to capture SNMP Traps sent from managed devices on the network.

- ◆ Monitors one or more network management applications (NMA) simultaneously; it has facilities to graphically show information about managed devices, table viewing, and logging.
- ◆ Allows you to compile the MIB file using the MIB compiler present in the NMS.

Managed Devices

A managed device is any device that has SNMP installed on it. A managed device could be a host, router, bridge, hub, etc. NMS can monitor and communicate with managed devices.

The information between the NMS and the managed device is transferred through two types of agents: subagent and master agent.

Subagent

The subagent gathers information about the managed device and passes the information to the master agent.

Master Agent

The master agent exchanges information between the various subagents and the NMS. The master agent runs on the same host machine as the subagents with which it communicates.

Management Information Base

SNMP exchanges network information in the form of protocol data units (PDUs). PDUs contain information about variables stored on the managed device. These variables are known as managed objects and have values and titles that are reported to the NMS. All managed objects are defined in the Management Information Base (MIB). MIB is a virtual database with a tree-like hierarchy.

SNMP Network Management Protocol

The basic functions of SNMP are listed in the following table.

Function	Description
Get	Used by the manager to request information from an agent.
Get Next	Used by the manager to obtain information from an array or a table.
Get Response	Used by the queried agent to satisfy a request made by the manager.
Set	Used by the manager to modify the value of the variable which resides on the agent's MIB.
Trap	Used by the agent to notify the manager that a certain event has occurred.

For more information about SNMP, refer to the following Web sites:

- ◆ [NET-SNMP Home Page \(http://net-snmp.sourceforge.net\)](http://net-snmp.sourceforge.net)
- ◆ [SNMP FAQ \(http://www.faqs.org/faqs/snmp-faq/part1\)](http://www.faqs.org/faqs/snmp-faq/part1)
- ◆ [RFC 1157 \(http://www.ietf.org/rfc/rfc1157.txt\)](http://www.ietf.org/rfc/rfc1157.txt)

- ◆ [SNMPLink \(http://www.snmplink.org\)](http://www.snmplink.org)
- ◆ [SNMPInfo \(http://www.snmpinfo.com\)](http://www.snmpinfo.com)
- ◆ [SNMP RFC Standard MIBs and Informative Links \(http://www.wtcs.org/snmp4tpc/snmp_rfc.htm\)](http://www.wtcs.org/snmp4tpc/snmp_rfc.htm)
- ◆ [RFC 2605 \(http://ietf.org/rfc/rfc2605.txt?number=2605\)](http://ietf.org/rfc/rfc2605.txt?number=2605)

eDirectory and SNMP

eDirectory can store and manage millions of objects, such as users, applications, network devices, and data. With the increase in objects, the need to track down the additions and modifications to the eDirectory increases. SNMP renders a solution to this problem by helping you monitor eDirectory servers and thus keep track of the changes.

Benefits of SNMP Instrumentation on eDirectory

- ◆ Real time monitoring for an eDirectory server
- ◆ Monitoring of eDirectory from any third party SNMP MIB browser
- ◆ Tracking the status of eDirectory to verify normal operations
- ◆ Spotting and reacting to potential problems once they are detected
- ◆ Configuring traps and statistics for selective monitoring
- ◆ Plotting a trend on the access of eDirectory
- ◆ Storing and analyzing historical data that has been obtained through SNMP
- ◆ SNMP Get, GetNext request support for statistics
- ◆ Using SNMP native master agent on all the platform

Understanding How SNMP Works with eDirectory

SNMP implementation on eDirectory provides useful eDirectory information on statistics on the accesses, operations, errors, and cache performance.

- ◆ “[Directory Service Monitoring MIB](#)” on page 320
- ◆ “[SNMP Group Object](#)” on page 321

Directory Service Monitoring MIB

Traps on the occurrence of events can also be sent with SNMP implementation. Traps and statistics are defined in the MIB.

The eDirectory MIB defines statistics and traps to monitor eDirectory. This MIB is assigned the following oid:

```
iso(1).org(3).dod(6).internet(1).private(4).enterprise(1).novell(23).mibDoc(2).ndsMIB(98)
```

Statistics

The eDirectory MIB is divided into four distinct tables of managed objects:

- ◆ **The Cache Database Statistics Table - ndsDbCacheTable:** Contains a description of the directory servers as well as summary statistics on the entries cached by these servers.

- ♦ **The Config Database Statistics Table - `ndsDbConfigTable`:** Contains a description of the directory servers as well as summary statistics on the entries configured by these servers.
- ♦ **The Protocol Statistics Table - `ndsProtoIfOpsTable`:** Provides summary statistics on the accesses, operations, and errors for each application protocol interface of a directory server.
- ♦ **The Interaction Statistics Table - `ndsServerIntTable`:** Keeps track of the last “N” directory server with which the monitored directory has interacted or attempted to interact. “N” is a locally defined constant.

NOTE: For more information on statistics, see [“Statistics” on page 359](#).

Traps - `ndsTrapVariables`

The eDirectory MIB defines 119 traps. Out of this, 117 traps map to eDirectory events and 2 additional traps `ndsServerStart` and `ndsServerStop` are directly generated by the SNMP subagent. These 2 traps cannot be configured.

For more information on traps, see [“Traps” on page 336](#). For more information on statistics and traps, see the `edir.mib` file located in the following directories:

Platform	Directory
NetWare	<code>sys:\etc</code>
Windows	<code>C:\novell\NDS\snmp</code>
UNIX	<code>/etc/ndssnmp/</code>

SNMP Group Object

The SNMP group object is used to set up and manage the eDirectory SNMP traps. During installation, an SNMP group object named “SNMP Group - *server_name*” is created (where *server_name* is the name of the server on which SNMP services for eDirectory are installed). The SNMP group object is created in the same container as the server object. This SNMP configuration utility is used to configure SNMP traps.

- ♦ [“On Windows” on page 321](#)
- ♦ [“On NetWare” on page 322](#)
- ♦ [“On UNIX” on page 322](#)

On Windows

The utility to create and delete an SNMP group object is `snmpinst`. This utility is located in the `C:\Program Files\Common Files\novell\ni\bin` directory.

To create an SNMP group object, enter the following command:

```
rundll32 snmpinst, snmpinst -c <createobj> -a <userFDN> -p  
<password> -h <hostname or IP address>
```

Parameter	Description
<code>-c <createobj></code>	Trap command that specifies the creation of an object.
<code>-a <userFDN></code>	Fully distinguished name of a user having administrative rights

Parameter	Description
-p <password>	userFDN password for authentication
-h <hostname or IP address>	DNS host name or IP address

Example:

```
rundll32 snmpinst, snmpinst -c createobj -a admin.mycontext -p mypassword -h 160.98.146.26
```

To delete an SNMP group object, enter the following command:

```
rundll32 snmpinst, snmpinst -c <deleteobj> -a <userFDN> -p  
<password> -h <hostname or IP address>
```

See the table above for more information.

Example:

```
rundll32 snmpinst, snmpinst -c deleteobj -a admin.mycontext -p mypassword -h 160.98.146.26
```

On NetWare

The utility to create and delete an SNMP group object is `snmpinst`. This utility is located in the `sys:\system\` directory.

To create an SNMP group object, enter the following command:

```
SNMPINST -c <adminContext> <password> <ServerDN>
```

Parameter	Description
-c	Trap command that specifies the creation of an object. For deletion, it is -d.
<adminContext>	Fully distinguished name of a user having administrative rights
<password>	userFDN password for authentication
<ServerDN>	FDN of the server object

Example:

```
SNMPINST -c admin.mycontext.treename mypassword myserver
```

To delete an SNMP group object, enter the following command:

```
SNMPINST -d <adminContext> <password> <ServerDN>
```

Refer to the table above for more details.

Example:

```
SNMPINST -d admin.mycontext.treename mypassword myserver
```

On UNIX

To create an SNMP group object, enter the following command:

```
ndsconfig add -m <modulename> -a <userFDN>
```

Example:

```
ndsconfig add -m snmp -a admin.mycontext
```

Installing and Configuring SNMP Services for eDirectory

SNMP service for eDirectory is installed when eDirectory is installed. You can modify the default configuration of SNMP services for eDirectory using iManager. For more information, see [“Dynamic Configuration” on page 325](#).

A new object called SNMP Group-Object is added to the directory tree when eDirectory is installed. This object is used to set up and manage the Novell eDirectory SNMP traps. See [“SNMP Group Object” on page 321](#) for more information.

Installing SNMP after eDirectory Installation on Windows

If the SNMP service is not installed with eDirectory, the eDirectory install copies only the required SNMP subagent files and does not update the registry.

If you want to use SNMP services on eDirectory at a later point in time, you can install the SNMP service and update the registry using the following command:

```
rundll32 snmpinst, snmpinst -c createreg
```

Loading and Unloading the SNMP Server Module

The SNMP server module can be manually loaded and unloaded. By default, the SNMP server module loads automatically on all platforms. However, you can manually load the server module on Windows and UNIX platforms.

To load the SNMP server module, enter the following commands:

Server	Command
NetWare	N.A
Windows	In the DHOST (NDSCONS) screen, select Ndssnmp.dlm > click Start.
Linux, Solaris, AIX, and HP-UX	In the DHOST remote management page, to load the SNMP trap server click on the SNMP Trap Server for Novell eDirectory 8.7.3 action icon to start. or At the prompt, enter <code>/usr/bin/ndssnmp -l</code> .

To unload the SNMP server module, enter the following commands:

Server	Command
NetWare	N.A
Windows	In the DHOST (NDSCONS) screen, select ndssnmp.dlm, then click Stop.

Server	Command
Linux, Solaris, AIX, and HP-UX	In the DHOST remote management page, to unload the SNMP trap server, click the SNMP Trap Server for Novell eDirectory 8.7.3 action icon to stop. or At the prompt, enter <code>/usr/bin/ndssnmp -u</code> .

Subagent Configuration

Static Configuration

Static configuration is used before bringing up the subagent. You can manually configure it by editing the `ndssnmp.cfg` file on Windows, Solaris, Linux, AIX, or the `dssnmp.cfg` file on NetWare. The `ndssnmp.cfg` file is located in the following directories:

Windows: `install_directory\SNMP\`

NetWare: `sys:\etc\`

UNIX: `/etc/ndssnmp/`

NOTE: If changes are made to the `ndssnmp.cfg` file, the subagent must be restarted.

You can provide configuration information to the subagent such as the following:

- ◆ INTERACTIVE *status*

Where *status* is either on or off. If the status is on, you are prompted to enter the username and password when starting the subagent. If the status is off, then the username and password will be taken from the secure store. Default = Off.

Examples:

INTERACTIVE on

INTERACTIVE off

- ◆ INTERACTION *value*

Where *value* is the number of interaction table entries. Range = 1 to 10. Default = 4.

Examples:

INTERACTION 4

INTERACTION 2

- ◆ MONITOR *status*

Where *status* is either on or off. Default = On.

Examples:

MONITOR on

MONITOR off

- ◆ SSLKEY *certificate_file*

Where *certificate_file* is the exported certificate along with the path. You must enter the path where this exported certificate exists.

Examples:

SSLKEY /home/guest/snmp-cert.der (UNIX)

SSLKEY c:\home\guest\snmp-cert.der (Windows NT and NetWare)

- ◆ SERVER *hostname/ipaddr*

Where *hostname* is the name of the host where the eDirectory server is installed and configured. Only the locally installed server is supported.

This is a required command in the file, otherwise none of the servers are monitored. Default: *hostname* of the local server.

Examples:

SERVER myserver

SERVER myserver:1524

NOTE: No spaces are allowed before or after ':' as part of the server command.

Dynamic Configuration

Dynamic configuration can be done in either of the following ways, anytime after the Directory service is up and running.

Command Line

A trap configuration command line utility can be used to configure SNMP traps for eDirectory.


The command line configuration utility can be used to:

- ◆ Enable or disable traps
- ◆ Set the trap interval
- ◆ Enable or disable failure traps
- ◆ List the enabled, disabled or all traps

NOTE: For more details, see [“Configuring Traps” on page 350](#).

iManager Plug-In

Traps can also be configured using Novell iManager. Novell iManager is a browser-based tool used for administering, managing, and configuring eDirectory objects. Novell iManager gives you the ability to assign specific tasks or responsibilities to users and to present the user with only the tools (with the accompanying rights) necessary to perform those sets of tasks.

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click SNMP Management > SNMP Overview.
- 3** Click View SNMP Group Objects, then click the name of the SNMP Group object you want to configure.
- 4** Specify the configurable parameters in the General/Traps page.
- 5** Click Apply, then click OK to save the new configuration settings.

NOTE: For more information, see the Novell iManager online help.

Setting Up SNMP Services for eDirectory

Setting up SNMP services for eDirectory requires the following steps:

1. Configuring the master agent
2. Starting the master agent
3. Configuring the subagent
4. Starting the subagent

NetWare

On NetWare, the native master agent (snmp.nlm) is installed by default with the operating system.

TIP: NetWare provides the default SNMP master agent. See [SNMP Developers Components \(http://developer.novell.com/ndk/snmpcomp.htm\)](http://developer.novell.com/ndk/snmpcomp.htm) for more information.

- ♦ “Configuring the Master Agent” on page 326
- ♦ “Starting the Master Agent” on page 326
- ♦ “Loading the Subagent” on page 326

Configuring the Master Agent

Community Name

- 1 Enter **inetcfg** at the command prompt.
- 2 Select the Manage Configuration option.
- 3 Select the Configure SNMP parameters option.
- 4 Edit the community string accordingly.

Trap Destination

- 1 Edit the file `sys:\etc\traptarg.cfg` and specify the IP address or hostname of the destination computer that the traps are sent to.

Starting the Master Agent

The master agent `snmp.nlm` is started by default.

Loading the Subagent

- 1 To load the subagent, enter **dssnmpsa** at the command prompt.
A dialog box is displayed with the Login and Exit options.
- 2 Select Login to proceed or Exit to discontinue.
- 3 (Conditional) If you selected Login, you are prompted for the login information. Enter the username and password.
- 4 Type **Y** in the Remember Password field to remember the password. When you start the subagent the next time, you are not prompted for the password. Type **N** to enter the password when the subagent is started the next time.
- 5 Press Enter after entering Y or N.
- 6 Press the function key F10 to log in to the tree.
- 7 Press Enter to continue.

8 The subagent is started.

NOTE: If INTERACTION is set to ON in the sys:\etc\ndssnmp.cfg file, this dialog box is displayed. If INTERACTION is OFF, it is not displayed.

Windows

- ♦ “Configuring the Master Agent” on page 327
- ♦ “Starting the Master Agent” on page 327
- ♦ “Stopping the Master Agent” on page 327
- ♦ “Starting the Subagent” on page 328

Configuring the Master Agent

NOTE: The SNMP master agent should be installed before eDirectory is installed. Refer to [SNMP Installation on Windows \(http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/prodtechnolog/winntas/maintain/feasability/getting.asp\)](http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/prodtechnolog/winntas/maintain/feasability/getting.asp) for more details.

- 1** In the Microsoft SNMP Properties dialog box, click the Agent tab.
- 2** Enter the Contact and Location information.
- 3** Click the Traps, then enter the Community Name and Trap destination details.
 - 3a** Enter the Community Name, then click Add.
 - 3b** Enter the IP address or hostname of the destination computer that traps are generated for.
 - 3c** Click Add to add the IP address or hostname.
- 4** Enable the Allow Service to Interact with Desktop option.

If it is not enabled, you will be unable to connect to SNMP on Windows.

- ♦ On Windows NT: Click Start > Settings > Control Panel > Services. Then click SNMP > Startup and select the Allow Service to Interact with Desktop option.
- ♦ On Windows 2000: Click Start > Settings > Control Panel > Administrative Tools > Services. Then right-click SNMP and select Properties. At the Log On tab, select the Allow Service to Interact with Desktop option.

Starting the Master Agent

To start the master agent, do either of the following:

- ♦ For Windows NT: Click Start > Settings > Control Panel > Services > SNMP > Start.
For Windows 2000: Click Start > Settings > Control Panel > Administrative Tools > Services > SNMP > Start.
- ♦ Enter the following at the command prompt:
Net start SNMP

Stopping the Master Agent

To stop the master agent, do either of the following:

- ♦ For Windows NT: Click Start > Settings > Control Panel > Services > SNMP > Stop.
For Windows 2000: Click Start > Settings > Control Panel > Administrative Tools > Services > SNMP > Stop.
- ♦ Enter the following at the command prompt:

Net stop SNMP**Starting the Subagent**

When the master agent starts on Windows, the subagent also starts.

IMPORTANT: The latest updated Service Pack needs to be installed after the installation of the SNMP service.

Solaris

- ◆ “Configuring the Master Agent” on page 328
- ◆ “Starting the Master Agent” on page 328
- ◆ “Configuring the Subagent” on page 328
- ◆ “Starting the Subagent” on page 329
- ◆ “Stopping the Subagent” on page 329

Configuring the Master Agent

Before you load SNMP Package, Solstice Enterprise master agent 1.0.3 should be installed in the system. If it is not installed, you need to download it from the [Solstice Enterprise Agents \(http://www.sun.com/software/entagents\)](http://www.sun.com/software/entagents) Web site.

- 1 In the `/etc/snmp/conf/snmpd.conf` file, identify a hostname. Add the following trap entry:

```
trap myserver
```

Where *myserver* is the hostname for the trap destination.

- 2 In the `/etc/snmp/conf/snmpdx.acl` file, add the following under the trap parameter section:

```
trap-community = public
hosts = myserver {
enterprise = "Novell eDirectory"
trap-num = 1-117, 2001, 2002 }
```

where *trap-community* is the community name used in traps, *myserver* is the trap destination host name, *Novell eDirectory* is the enterprise MIB, and *trap-num* is the trap range.

IMPORTANT: If any configuration files are changed, the master agent and subagent should be restarted.

Starting the Master Agent

To start the master agent, execute the following command:

```
/usr/lib/snmp/snmpdx -y -c /etc/snmp/conf
```

Configuring the Subagent

On Solaris, the subagent `ndssnmpsa` is a daemon process.

To configure the subagent, the following configuration files (located in `/etc/snmp/conf/`) are required:

- ◆ **ndsmib.reg** is the registration file for the subagent
- ◆ **ndsmib.acl** is the configuration file of the SNMP subagent

Starting the Subagent

You cannot invoke the subagent using Master agent resource file. You can invoke subagent only after master agent has been invoked.

To start the subagent, execute the following command:

```
/etc/init.d/ndssnmpsa start
```

Enter the username and password when prompted. Upon successful authentication, the following message is displayed if INTERACTION = ON in the /etc/ndssnmp/ndssnmp.cfg file:

```
Do you want to remember password? (Y/N)
```

Enter **Y** to remember the password. When you start the subagent the next time, you are not prompted for the password.

Enter **N** to enter the password when the subagent is started the next time.

Stopping the Subagent

To stop the subagent, execute the following command:

```
/etc/init.d/ndssnmpsa stop
```

Linux

On Linux (except SLES 9 32-bit or OES Linux, but including SLES 9 64-bit), net-snmp-5.0.9-4.rh73.i386.rpm should be installed. On SLES 9 32-bit (OES Linux) the default master agent on the system (net-snmp-5.1-80.xx) is used.

The procedure to configure for SLES 9 (OES Linux) and other flavors of Linux vary. For more information, refer to:

- ◆ [“Setting up SNMP Services on SLES 9 32-Bit or OES Linux” on page 329](#)
- ◆ [“Setting up SNMP Services on Linux \(Other than SLES 9 32-Bit or OES\)” on page 330](#)

Setting up SNMP Services on SLES 9 32-Bit or OES Linux

- ◆ [“Configuring the Master Agent” on page 329](#)
- ◆ [“Starting the Master Agent” on page 330](#)
- ◆ [“Starting the Subagent” on page 330](#)
- ◆ [“Stopping the Subagent” on page 330](#)

Configuring the Master Agent

To configure the master agent on SLES 9 32-bit or OES Linux, make the changes to your snmpd.conf file as mentioned in [“Snmpd.conf Changes” on page 329](#).

The snmpd.conf file is located in the /etc/snmp directory on OES Linux or SLES 9 and in the /etc directory on other Linux platforms.

Snmpd.conf Changes

In the snmpd.conf file, enter the hostname

```
trapsink myserver public
```

Where, *myserver* is the hostname for the trap destination.

In the `snmpd.conf` file, add the following line:

```
master agentx
```

Additionally, make the following changes:

Original Content	Changed Content
<code>com2sec notConfigUser default public</code>	<code>com2sec demouser default public</code>
<code>group notConfigGroup v1 notConfigUser</code>	<code>group demogroup v1 demouser</code>
<code>view systemview included system</code>	<code>view all included .1</code>
<code>access notConfigGroup "" any noauth exact systemview none none</code>	<code>access demogroup "" any noauth exact all all all</code>

If the above content is not present in the `snmpd.conf` file, add it.

IMPORTANT: If any configuration files are changed, the master agent and subagent should be restarted.

Starting the Master Agent

To start the master agent, execute the following command:

```
/usr/sbin/snmpd -C -c /etc/snmpd.conf
```

Starting the Subagent

To start the subagent, execute the following command:

```
/etc/init.d/ndssnmpsa start
```

Enter the username and password when prompted. Upon successful authentication, the following message is displayed if `INTERACTION = ON` in the `/etc/ndssnmp/ndssnmp.cfg` file:

```
Do you want to remember password? (Y/N)
```

Enter **Y** to remember the password. When you start the subagent the next time, you are not prompted for the password.

Enter **N** to enter the password when the subagent is started the next time.

IMPORTANT: For SLES 9 32-Bit or OES Linux, refer to the Readme for known issues while starting the subagent.

Stopping the Subagent

To stop the subagent, execute the following command:

```
/etc/init.d/ndssnmpsa stop
```

Setting up SNMP Services on Linux (Other than SLES 9 32-Bit or OES)

This section includes the SLES 9 64-bit configuration.

- ◆ [“Configuring the Master Agent” on page 331](#)
- ◆ [“Starting the Master Agent” on page 331](#)
- ◆ [“Starting the Subagent” on page 332](#)
- ◆ [“Stopping the Subagent” on page 332](#)

Configuring the Master Agent

Download net-snmp-5.0.9-4.rh73.i386.rpm from <http://sourceforge.net/projects/net-snmp> (<http://sourceforge.net/projects/net-snmp>).

The net-snmp-5.0.9-4.rh73.i386.rpm requires rpm-4.0.4-7x.i386.rpm to be installed on the system. You can download this from <http://rpmfind.net/linux/RPM/rpm.org/rpm/dist/rpm-4.0.x/rpm-4.0.4-7x.i386.html> (<http://rpmfind.net/linux/RPM/rpm.org/rpm/dist/rpm-4.0.x/rpm-4.0.4-7x.i386.html>).

Additionally, you need to make changes to the snmpd.conf file as specified in “[Snmpd.conf Changes](#)” on page 329.

Starting the Master Agent

To start the master agent, firstly install and configure net-snmp-5.0.9-4.rh73.i386.rpm.

You can do so using any of the two options mentioned below. However, we recommend you to use Option 1 as the second option requires you to uninstall the system installed SNMP packages and this may need you to uninstall all the dependent rpms too.

Option 1

- 1 Install net-snmp-5.0.9-4.rh73.i386.rpm and rpm-4.0.4-7x.i386.rpm on to a custom location for example, /home/ndssnmp.

Install net-snmp-5.0.9-4.rh73.i386.rpm as follows:

```
# cd /home/ndssnmp
# rpm2cpio net-snmp-5.0.9-4.rh73.i386.rpm | cpio -ivd
```

- 2 Install rpm-4.0.4-7x.i386.rpm (this is dependent rpm which snmpd requires)

```
# cd /home/ndssnmp
# rpm2cpio rpm-4.0.4-7x.i386.rpm | cpio -ivd
```

- 3 Export LD_LIBRARY_PATH as follows:

```
# export LD_LIBRARY_PATH=/home/ndssnmp/usr/lib
```

- 4 Start the master agent as follows:

```
# /home/ndssnmp/usr/sbin/snmpd -C -c snmpd.conf
```

For example, if your snmpd.conf file is present in the /etc directory, the command would be similar to the following:

```
# /home/ndssnmp/usr/sbin/snmpd -C -c /etc/snmpd.conf
```

NOTE: Ensure that the snmpd.conf file has the relevant information required for ndssnmpsa to start. Refer to “[Setting up SNMP Services on SLES 9 32-Bit or OES Linux](#)” on page 329 for more information.

- 5 (Conditional) While starting master agent you may encounter the following error:

```
snmpd: error while loading shared libraries: libcrypto.so.2: cannot open
shared object file: No such file or directory
```

You will get this error if libcrypto.so.2 not being installed on your system.

For this you have to make an explicit link to system installed crypto library as mentioned below:

```
# cd /usr/lib
```

Additionally, add any one of the following based on your Linux version:

- ♦ **For Red Hat Advanced Server 3.0:**

```
# ln -s libcrypto.so libcrypto.so.2
```

- ♦ **For SUSE Linux Enterprise Server 8:**

```
# ln -s libcrypto.so.0.9.6 libcrypto.so.2
```

- 6** (Conditional) While starting master agent on SLES 9 64-bit, you may encounter the following error:

```
error while loading shared libraries:
libdb.so.2: cannot open shared object file: No such file or directory
```

You will get this error if libdb.so.2 not being installed on your system.

For this you have to make an explicit link to system installed db library as mentioned below (using libdb.so.3 as an example):

```
# cd /usr/lib
# ln -s libdb.so.3 libdb.so.2
```

- 7** (Conditional) If the SNMP master agent is already configured on a default port #161 then start the master agent on different port as:

```
# /home/ndssnmp/usr/sbin/snmpd -C -c /etc/snmpd.conf 1161
```

Option 2

- 1** Uninstall system installed snmp package
- 2** If the SNMP package is already installed and the version is anything other than net-snmp-5.0.9-4.rh73.i386.rpm, then uninstall the SNMP package and install net-snmp-5.0.9-4.rh73.i386.rpm.

NOTE: If any dependent RPM is required, then download those and install them as well.
- 3** Start the master agent as follows:

```
/usr/sbin/snmpd -C -c /etc/snmpd.conf
```

Starting the Subagent

To start the subagent, execute the following command:

```
/etc/init.d/ndssnmpsa start
```

Enter the username and password when prompted. Upon successful authentication, the following message is displayed if INTERACTION = ON in the /etc/ndssnmp/ndssnmp.cfg file:

```
Do you want to remember password? (Y/N)
```

Enter **Y** to remember the password. When you start the subagent the next time, you are not prompted for the password.

Enter **N** to enter the password when the subagent is started the next time.

Stopping the Subagent

To stop the subagent, execute the following command:

```
/etc/init.d/ndssnmpsa stop
```

AIX

- ◆ “Configuring the Master Agent” on page 333
- ◆ “Starting the Master Agent” on page 333
- ◆ “Starting the Subagent” on page 333
- ◆ “Stopping the Subagent” on page 334

Configuring the Master Agent

In the `/etc/snmpd.conf` file, add the following trap destination entry:

```
trap community myserver view_name trap_mask
```

where

- ◆ *community* is the community name that will be encoded in the trap packet
- ◆ *myserver* is the hostname for trap destination
- ◆ *view_name* is the unique object identifier in dotted numeric notation
For example: 1.3.6.1.4.1.23.2.98. This is an optional parameter. If this is not included, the view defaults to the entire MIB tree.
- ◆ *trap_mask* is in the hexadecimal format
The bits from left to right stand for coldStart trap, warmStart trap, linkDown trap, linkUp trap, authenticationFailure trap, egpNeighborLoss trap, and enterpriseSpecific trap. In the example, the value “98” on the right does not have any meaning. The value “1” enables the corresponding trap to be sent. Otherwise, the trap is blocked.

Example:

```
fe  block no traps (1111 1110)
7e  block coldStart trap (0111 1110)
be  block warmStart trap (1011 1110)
3e  block coldStart trap and warmStart trap (0011 1110)
```

Starting the Master Agent

To start the master, execute the following command:

```
/usr/sbin/snmpd
```

Starting the Subagent

To start the subagent, execute the following command:

```
/etc/ndssnmppsa start
```

Enter the username and password when prompted. Upon successful authentication, the following message is displayed if INTERACTION= ON in the `/etc/ndssnmp/ndssnmp.cfg` file:

```
Do you want to remember password? (Y/N)
```

Enter **Y** to remember the password. When you start the subagent the next time, you are not prompted for the password.

Enter **N** to enter the password when the subagent is started the next time.

Stopping the Subagent

To stop the subagent, execute the following command:

```
/etc/ndssnmpsa stop
```

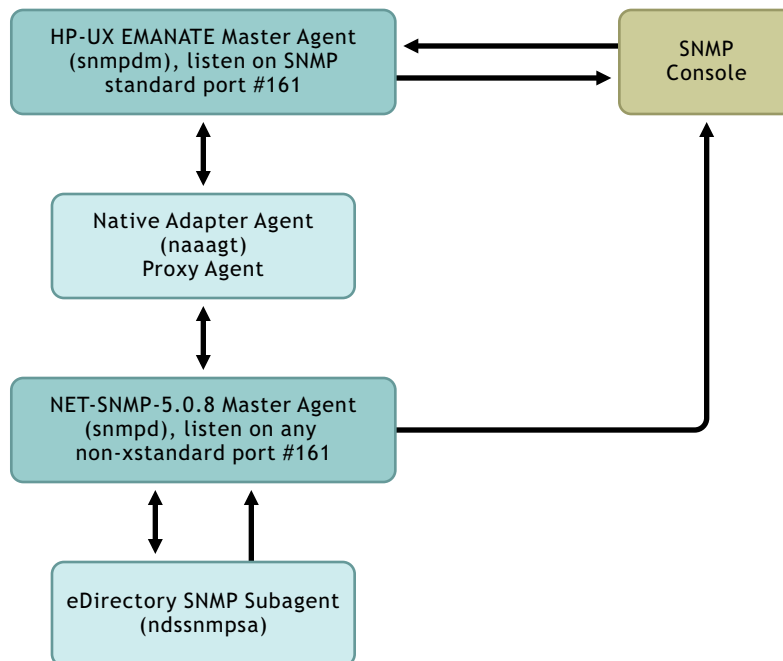
HP-UX

On HP-UX, the native master agent is EMANATE SNMP master agent. Configuring the master agent on HP-UX involves proxy SNMP agent configuration. The Proxy agent configuration is done through Native Adapter Agent (NAA). This NAA allows third-party SNMP agents to work with the HP-UX SNMP master agent (snmpdm). The third-party SNMP agent in our case is NET-SNMP master agent. The NET-SNMP master agent must listen on the same non-standard UDP port that NAA has been configured.

For details refer to section [“Starting/Configuring the Native Agent Adapter \(NAA\)” on page 335](#) and [“Starting/Configuring the NET-SNMP Master Agent” on page 335](#).

The following figure illustrates the flow of data between the eDirectory SNMP subagent, NET-SNMP master agent, NAA agent, the HP-UX EMANATE master agent, and the SNMP console.

Figure 34 SNMP Data Flow



- ◆ [“Starting the HP-UX SNMP Master Agent” on page 335](#)
- ◆ [“Starting/Configuring the Native Agent Adapter \(NAA\)” on page 335](#)
- ◆ [“Starting/Configuring the NET-SNMP Master Agent” on page 335](#)
- ◆ [“Starting the Subagent” on page 336](#)
- ◆ [“Stopping the Subagent” on page 336](#)

Starting the HP-UX SNMP Master Agent

To start the HP-UX SNMP master agent, execute the following command:

```
/etc/snmpd
```

or

```
/usr/sbin/snmpdm
```

NOTE: To stop the HP-UX SNMP master agent, enter `/etc/snmpd -k`

Starting/Configuring the Native Agent Adapter (NAA)

Before starting the NAA agent (`naaagt`), export the following environment variables:

- ◆ `HP_NAA_CNF` - the NAA configuration file
- ◆ `HP_NAA_PORT` - a non-standard UDP port that the net-snmp master agent listens to
- ◆ `HP_NAA_GET_COMMUNITY` - the community name to be used in the SNMP requests forwarded from NAA to the net-snmp master agent

For example:

```
export HP_NAA_CNF=/etc/ndssnmp/ndssnmpNAA.cfg
export HP_NAA_PORT=8161 ## Specify any non-standard UDP port
export HP_NAA_GET_COMMUNITY=public
```

For details on the NAA agent, refer to the `naaagt` man page.

Enter the following command to start the NAA agent:

```
/usr/sbin/naaagt
```

NOTE: Root access is required to start the NAA agent.

Starting/Configuring the NET-SNMP Master Agent

Before configuring the NET-SNMP master agent, you need to first download and install it.

- 1** Download the NET-SNMP version 5.0.8 tar file (`net-snmp-5.0.8-HP-UX_B.11.00_9000_712.tar.gz`) from [SourceForge.net](http://sourceforge.net/project/showfiles.php?group_id=12694) (http://sourceforge.net/project/showfiles.php?group_id=12694).
- 2** Install NET-SNMP version 5.0.8 binaries by untaring the above mentioned tar file.
After untaring the tar file, NET-SNMP version 5.0.8 binaries are installed to `current_working_directory/usr/local`.

To configure the NET-SNMP master agent:

- ◆ In the `/etc/ndssnmp/snmpd-net-snmp.conf` file, enter the hostname `trapsink myserver public`
where `myserver` is the hostname for the trap destination.
- ◆ In the `/etc/ndssnmp/snmpd-net-snmp.conf` file, add the following line if it is not already added:
`master agentx`

NOTE: Because the NET-SNMP-5.0.8 binary download does not come with a sample master agent configuration file, the NET-SNMP sample master agent configuration file is bundled with the eDirectory SNMP component. After eDirectory is installed, you can get the sample NET-SNMP configuration file (snmpd-net-snmf.conf file) from the /etc/ndssnmp directory.

To start NET-SNMP-5.0.8 Master Agent, use the following syntax:

```
installed_NET-SNMP_directory/usr/local/sbin/snmpd -C -c /etc/ndssnmp/snmpd-net-snmf.conf 8161
```

IMPORTANT: If any configuration files are changed, the master agent and subagent should be restarted.

Starting the Subagent

To start the subagent, execute the following command:

```
/sbin/init.d/ndssnmfpa start
```

Enter the username and password when prompted. Upon successful authentication, the following message is displayed if INTERACTION = ON in the /etc/ndssnmp/ndssnmf.cfm file:

```
Do you want to remember password? (Y/N)
```

Enter **Y** to remember the password. When you start the subagent the next time, you are not prompted for the password.

Enter **N** to enter the password when the subagent is started the next time.

Stopping the Subagent

To stop the subagent, execute the following command:

```
/sbin/init.d/ndssnmfpa stop
```

Monitoring eDirectory Using SNMP

eDirectory is monitored using the traps and statistics feature of SNMP.

To monitor an eDirectory server using SNMP, you need the following rights over the NCP server, LDAP group and LDAP server objects:

- ◆ Supervisor rights over the NCP server object
- ◆ Read rights over the LDAP Allow Clear Text Password attribute of the LDAP Group object
- ◆ Read rights over the LDAP TCP Port and LDAP SSL Port attributes of the LDAP Server object

By default a user who has logged in with the administrative rights does not face any problem in monitoring an eDirectory server using SNMP.

Traps

The SNMP component generates a total of 119 traps out of which traps ndsServerStart (2001) and ndsServerStop (2002) cannot be configured. These traps are enabled by default.

You can use a MIB browser to check the generated traps.

NOTE: Trap numbers 42, 92 and 100 are specific to NetWare.

Trap Number	Trap Name	Trap Is Generated When
1	ndsCreateEntry	<p>A new object is added in the directory.</p> <p>Example:</p> <p>Create an object using LDAP tools, ICE, ConsoleOne®, or iManager.</p>
2	ndsDeleteEntry	<p>An existing object is deleted.</p> <p>Example:</p> <p>Create an object using LDAP tools, ICE, ConsoleOne, or iManager.</p>
3	ndsRenameEntry	<p>An existing object is renamed.</p> <p>Example:</p> <p>Rename an object using LDAP tools, ICE, ConsoleOne, or iManager.</p>
4	ndsMoveSourceEntry	<p>An object is moved to a different context. The trap gives the context of the object before movement.</p> <p>Example:</p> <p>Move an object using ldapmodrdn or ldap sdk.</p>
5	ndsAddValue	<p>A value is added to an object attribute.</p> <p>Example:</p> <p>Add new values to attributes using LDAP tools, ICE, ConsoleOne, or iManager.</p>
6	ndsDeleteValue	<p>A value is deleted from an object attribute.</p> <p>Example:</p> <p>Delete new values to attributes using LDAP tools, ICE, ConsoleOne, or iManager.</p>
7	ndsCloseStream	A stream attribute is modified.
8	ndsDeleteAttribute	<p>A value is deleted from a single-value attribute.</p> <p>Example:</p> <p>Delete an attribute using LDAP tools, ICE, ConsoleOne, or iManager.</p>
9	ndsCheckSecurityEquiv	<p>The security equivalence vector for the particular entry is checked.</p> <p>Example:</p> <p>Change the security equivalence attribute using LDAP tools, ICE, ConsoleOne, or iManager.</p>

Trap Number	Trap Name	Trap Is Generated When
10	ndsUpdateSecurityEquiv	The security equivalence vector for the particular entry is modified. Example: Change the security equivalence attribute using LDAP tools, ICE, ConsoleOne, or iManager.
11	ndsMoveDestEntry	An object is moved to a different context. The trap will give the context that the object is moved to. Example: Move objects using ldapmodrdn or ldapsdk.
12	ndsDeleteUnusedExtref	A backlink object is deleted.
13	ndsAgentOpenLocal	The local directory agent is opened. Example: Run unattended repair.
14	ndsAgentCloseLocal	The local directory agent is closed. Example: Run unattended repair.
15	ndsDSABadVerb	An incorrect verb number is associated with an DSAgent request. Example: Pass a bad verb request to eDirectory using DClient calls.
16	ndsMoveSubtree	A container and its subordinate object are moved. Example: When a partition is moved to a different context using LDAP tools, ICE, ConsoleOne, or iManager.
17	ndsNoReplicaPointer	A replica has no replica pointer associated with it.
18	ndsSynclnEnd	Inbound synchronization is completed.
19	ndsBacklinkSecurEquiv	A backlink operation has updated an object's security equivalence vector. Example: Change the security equivalence attribute using LDAP tools, ICE, ConsoleOne, or iManager.
20	ndsBacklinkOperPrivChg	A backlink operation has changed an object's console operator privileges.
21	ndsDeleteSubtree	A container and its subordinate objects have been deleted.
22	ndsReferral	A referral is created.

Trap Number	Trap Name	Trap Is Generated When
23	ndsUpdateClassDef	A schema class definition is updated. Example: When a new class or attribute is added to a primary and this gets synchronized with the secondary using LDAP tools, ICE, ConsoleOne, or iManager, this trap is generated.
24	ndsUpdateAttributeDef	A schema attribute definition is updated. Example: When a new attribute is added to a primary and this is synchronized with the secondary using LDAP tools, ICE, ConsoleOne, or iManager, this trap is generated.
25	ndsLostEntry	eDirectory encounters a lost entry. A lost entry is an entry that does not exist on the local server, but for which updates are being received.
26	ndsPurgeEntryFail	The purge operation fails.
27	ndsPurgeStart	The purge operation is started. Example: Run dstrace and Set ndstrace=*j.
28	ndsPurgeEnd	The purge operation is completed. Example: Run dstrace and Set ndstrace=*j.
29	ndsLimberDone	The limber operation is completed. Example: Configure dstrace to start limber after a particular interval of time.
30	ndsPartitionSplitDone	The split partition operation is completed. Example: Create a partition using ConsoleOne or iManager.
31	ndsSyncServerOutStart	Outbound synchronization from a particular server is started. Example: Configure dstrace to start outbound synchronization after a particular interval of time.
32	ndsSyncServerOutEnd	Outbound synchronization from a particular server is completed. Example: Configure dstrace to stop outbound synchronization after a particular interval of time.

Trap Number	Trap Name	Trap Is Generated When
33	ndsSyncPartitionStart	Partition synchronization is started. Example: Partition one of the containers.
34	ndsSyncPartitionEnd	Partition synchronization is completed. Example: Partition one of the containers.
35	ndsMoveTreeStart	Movement of a subtree is started. A subtree is moved when a partition is moved. Example: Using ConsoleOne or iManager, create a partition and move the partition to another container.
36	ndsMoveTreeEnd	Movement of a subtree is completed. A subtree is moved when a partition is merged. Example: Using ConsoleOne or iManager, create a partition and move the partition to another container.
37	ndsJoinPartitionDone	Joining of partitions is completed. Example: Using ConsoleOne or iManager, create a partition and merge the partition.
38	ndsPartitionLocked	A partition gets locked (for example, before merging the partitions). Example: Using ConsoleOne or iManager, create a partition.
39	ndsPartitionUnlocked	A partition gets unlocked (for example, after merging the partitions). Example: Using ConsoleOne or iManager, create a partition.
40	ndsSchemaSync	Schema are synchronized. Example: Schedule schema synchronization using ldapsdk schsync.

Trap Number	Trap Name	Trap Is Generated When
41	ndsNameCollision	<p>Two objects on different servers have the same name (they <i>collide</i>).</p> <p>Example:</p> <p>Disable the outbound synchronization of the primary and secondary servers of a tree using iMonitor. Add some User objects to both the servers using LDAP tools. Then enable the outbound synchronization of both servers using iMonitor.</p>
42	ndsNLMLoaded	<p>An NLM™ program is loaded in NetWare.</p> <p>This trap is applicable only for NetWare.</p> <p>Example:</p> <p>Load or unload nldap.nlm.</p>
43	ndsChangeModuleState	<p>An eDirectory module (NLM / DLM) is loaded or unloaded.</p> <p>Example:</p> <p>Load or unload the nldap module.</p>
44	ndsLumberDone	<p>The limber background process is started.</p>
45	ndsBacklinkProcDone	<p>The backlink process is completed.</p> <p>Example:</p> <p>Configure dstrace to start backlink after a particular interval of time.</p>
46	ndsServerRename	<p>A server is renamed.</p> <p>Example:</p> <p>Use ldapmodrtn or ldapsdk to rename the server.</p>
47	ndsSyntheticTime	<p>Objects are created with future time stamps. To synchronize eDirectory servers, synthetic time might be invoked.</p> <p>Example:</p> <p>Add a secondary server to the tree using ndsconfig.</p>
48	ndsServerAddressChange	<p>Limber changes a server referral.</p> <p>Example:</p> <p>Change the IP address of the server and restart ndsd.</p>
49	ndsDSARead	<p>An entry is read.</p> <p>This trap is generated for all operations on eDirectory.</p> <p>Example:</p> <p>Use ldapsearch to generate traps.</p>

Trap Number	Trap Name	Trap Is Generated When
50	ndsLogin	eDirectory is logged in to. Example: Login to the tree using ndslogin.
51	ndsChangePassword	A password is changed. Example: Change the password of a user object using ldapmodify.
52	ndsLogout	eDirectory is logged out of. Example: Detach the connection to the tree from Novell Client.
53	ndsAddReplica	A replica is added to a server partition. Example: Add a new replica to the tree using ndsconfig.
54	ndsRemoveReplica	A replica is deleted. Example: Delete a replica from one of the server using ConsoleOne or iManager.
55	ndsSplitPartition	A partition is split. Example: Create a partition using ConsoleOne or iManager.
56	ndsJoinPartition	A parent partition is joined with a child partition. Example: Create a partition and join the partition using ConsoleOne or iManager.
57	ndsChangeReplicaType	A partition replica's type is changed. Example: Change the replica type from Master replica to Read-Write replica.
58	ndsAddEntry	A new object is added. Example: Add a user object using ConsoleOne or iManager.
59	ndsAbortPartitionOp	A partition operation is aborted. Example: Partition a container and abort the partitioning operation.

Trap Number	Trap Name	Trap Is Generated When
60	ndsRecvReplicaUpdates	<p>A replica receives an update during synchronization.</p> <p>Example:</p> <p>When an eDirectory server in a multiple servers tree setup, requests for updates on the replica that it holds. This operation can be done using ConsoleOne or iManager.</p>
61	ndsRepairTimeStamps	<p>A replica's time stamps are repaired.</p> <p>Example:</p> <p>Perform a DIB repair operation for timestamps using dsrepair (ndsrepair on UNIX, or NDSCons on Windows.)</p>
62	ndsSendReplicaUpdates	<p>A replica is updated during synchronization.</p> <p>Example:</p> <p>When an eDirectory server in a multiple servers tree setup sends for updates on the replica that it holds. This operation can be done using ConsoleOne or iManager.</p>
63	ndsVerifyPass	<p>A password is verified.</p> <p>Example:</p> <p>When the password expires, re-enter the password for confirmation at the change password prompt.</p>
64	ndsBackupEntry	<p>An entry is backed up.</p> <p>Example:</p> <p>Back up Directory objects using the dsbackup utility (ndsbackup on UNIX, NDSCons on Windows).</p>
65	ndsRestoreEntry	<p>An entry is restored.</p> <p>Example:</p> <p>Restore the backed-up Directory objects using the dsbackup utility (ndsbackup on UNIX, NDSCons on Windows).</p>
66	ndsDefineAttributeDef	<p>An attribute definition is added to the schema.</p> <p>Example:</p> <p>Extend the eDirectory tree schema by adding a new attribute definition. The schema can get extended when an eDirectory dependent application is installed such as ZENWorks® or NMASTM. The schema can also be extended using ConsoleOne, iManager, or the schema extension utility ndssch on UNIX.</p>

Trap Number	Trap Name	Trap Is Generated When
67	ndsRemoveAttributeDef	<p>An attribute definition is removed from the schema.</p> <p>Example:</p> <p>Delete an attribute definition from the eDirectory tree schema. The attribute can be deleted using ConsoleOne, iManager or the schema extension utility ndssch on UNIX.</p>
68	ndsRemoveClassDef	<p>A class definition is removed from the schema.</p> <p>Example:</p> <p>Delete an object class definition from the eDirectory tree schema. This can be deleted using ConsoleOne, iManager, or the schema extension utility ndssch on UNIX.</p>
69	ndsDefineClassDef	<p>A class definition is added to the schema.</p> <p>Example:</p> <p>Extend the eDirectory tree schema by adding a new class. The schema can get extended when an eDirectory dependent application is installed such as ZENWorks or NMAS. The schema can also be extended using ConsoleOne, iManager, or the schema extension utility ndssch on UNIX.</p>
70	ndsModifyClassDef	<p>A class definition is modified.</p> <p>Example:</p> <p>Modify an existing object class or attribute definitions.</p>
71	ndsResetDSCounters	The internal eDirectory counters are reset.
72	ndsRemoveEntryDir	A file directory associated with an entry is removed.
73	ndsCompAttributeValue	<p>Attribute values are compared.</p> <p>Example:</p> <p>Compare an attribute value against any object.</p> <p>Perform an LDAP search operation against a User object to check if its telephone number is the same as the input value.</p>
74	ndsOpenStream	<p>A stream attribute is opened or closed.</p> <p>Example:</p> <p>Create or open a stream for read or write operations.</p> <p>Create a login script for a User object. It creates a file under the DIB directory, which results in the generation of this trap.</p>
75	ndsListSubordinates	<p>A List Subordinate Entries operation is performed on a container object. It is a one-level search.</p> <p>Example:</p> <p>Using ConsoleOne or iManager, click a container object to list the objects under it.</p>

Trap Number	Trap Name	Trap Is Generated When
76	ndsListContainerClasses	<p>A List Containable Classes operation is performed on an entry.</p> <p>Example:</p> <p>For a given object, list the container classes that can contain the given object.</p> <p>When queried against a user object, the container classes that can contain it are Organization, Organizational Unit, and Domain Classes.</p>
77	ndsInspectEntry	<p>An Inspect Entry operation is performed on an entry.</p> <p>Example:</p> <p>Inspect any entry to obtain information about the entry and to check if there are any errors that the entry has experienced.</p> <p>This event is generated as part of the Flat Cleaner background process of eDirectory, which results in this trap generation.</p>
78	ndsResendEntry	<p>A Resend Entry operation is performed on an entry.</p> <p>Example:</p> <p>During replication operation when an entry is resent because of a failure in sending the object earlier as a result of connection between the servers.</p>
79	ndsMutateEntry	<p>A Mutate Entry operation is performed on an entry.</p> <p>Example:</p> <p>Mutate a bindery object class to User object class.</p>
80	ndsMergeEntries	<p>Two entries are merged.</p> <p>Example:</p> <p>Merge two User objects. Merge Entry2 (ndsEntryName2) into Entry (ndsEntryName).</p>
81	ndsMergeTree	<p>Two eDirectory trees are merged.</p> <p>Example:</p> <p>Merge two eDirectory trees using dsmerge (ndsmerge on UNIX, NDSCons on Windows).</p>
82	ndsCreateSubref	<p>A subordinate reference is created.</p> <p>Example:</p> <p>Delete the replica of the child partition from a server, the Subordinate Reference replica gets created automatically which results in the generation of this trap.</p>

Trap Number	Trap Name	Trap Is Generated When
83	ndsListPartitions	<p>A List Partitions operation is performed.</p> <p>Example:</p> <p>Using ConsoleOne or iManager, from Partition and Schema view, click the eDirectory Server object to list the partitions held by the server.</p>
84	ndsReadAttribute	<p>A value of an attribute is read.</p> <p>Example:</p> <p>Perform a search operation on the tree.</p>
85	ndsReadReferences	<p>An entry's references are read.</p>
86	ndsUpdateReplica	<p>An Update Replica operation is performed on a partition replica.</p> <p>Example:</p> <p>Delete a user from one of the servers; the other replica is updated for the delete operation.</p>
87	ndsStartUpdateReplica	<p>A Start Update Replica operation is performed on a partition replica.</p> <p>Example:</p> <p>Delete a user from one of the servers; the other replica is updated for the delete operation.</p>
88	ndsEndUpdateReplica	<p>An End Update Replica operation is performed on a partition replica.</p> <p>Example:</p> <p>Delete a user from one of the servers; the other replica is updated for the delete operation.</p>
89	ndsSyncPartition	<p>A Synchronize Partition operation is performed on a partition replica.</p> <p>Example:</p> <p>Delete a user from one of the partitions. The sync can be observed using ndstrace.</p>
90	ndsSyncSchema	<p>The master replica of the root receives a request to synchronize its schema with the server.</p> <p>Example:</p> <p>Add a new class using ConsoleOne > Wizard > Schema, LDAP tools, or ndssch utilities.</p>

Trap Number	Trap Name	Trap Is Generated When
91	ndsCreateBackLink	<p>A backlink is created. (A backlink is created when an object not present locally is being referenced.)</p> <p>Example:</p> <p>In a multi-server scenario, create a partition with some users. Delete this partition from one of the servers; this will create a subordinate reference. A backlink will be created for all the users present in the deleted partition.</p>
92	ndsCheckConsoleOperator	<p>Backlinker checks for console operator privileges.</p> <p>This trap is applicable only for NetWare.</p>
93	ndsChangeTreeName	<p>The tree name is changed.</p> <p>Example:</p> <p>Using the merge utility dsmerge/ndsmerge to rename the tree.</p>
94	ndsStartJoinPartition	<p>A Start Join operation is performed to merge partitions.</p> <p>Example:</p> <p>Merge or join partitions using ConsoleOne or LDAP tools.</p>
95	ndsAbortJoinPartition	<p>A Join Partition operation is aborted to stop merge partition.</p> <p>Example:</p> <p>Merge or join partitions using ConsoleOne or LDAP tools.</p>
96	ndsUpdateSchema	<p>An Update Schema operation is performed.</p> <p>Example:</p> <p>Add a new class using ConsoleOne > Wizard > Schema, LDAP tools, or ndssch.</p>
97	ndsStartUpdateSchema	<p>A Start Update Schema operation is performed.</p> <p>Example:</p> <p>Add a new class using ConsoleOne > Wizard > Schema, LDAP tools, or ndssch.</p>
98	ndsEndUpdateSchema	<p>An End Update Schema operation is performed.</p> <p>Example:</p> <p>Add a new class using ConsoleOne > Wizard > Schema, LDAP tools, or ndssch.</p>
99	ndsMoveTree	<p>A Move Tree operation is performed.</p> <p>Example:</p> <p>Move a partition from one container to another.</p>

Trap Number	Trap Name	Trap Is Generated When
100	ndsReloadDS	DS is reloaded. This trap is applicable only on NetWare. Example: set dstrace=*
101	ndsConnectToAddress	A connection is established with a particular address. Example: Browse the tree using ConsoleOne or iManager.
102	ndsSearch	A Search operation is performed. Example: Perform ldapsearch on the tree using LDAP tools.
103	ndsPartitionStateChange	A partition is created or deleted. Example: Create a new partition.
104	ndsRemoveBacklink	Unused external references are removed and the server sends a remove backlink request to the server holding the object.
105	ndsLowLevelJoinPartition	A low-level join is performed during merge partition operations. Example: Merge or join partitions using ConsoleOne, iManager, or LDAP tools.
106	ndsCreateNameBase	An eDirectory namebase is created.
107	ndsChangeSecurityEquals	The Security Equals attribute is modified. Example: Change the security equivalent of any user and make it equal to admin using ConsoleOne or iManager.
108	ndsRemoveEntry	An entry is removed from eDirectory. Example: Delete any user using ConsoleOne or iManager.
109	ndsCRCFailure	A CRC failure occurs when fragmented NCP requests are being reconstructed.
110	ndsModifyEntry	An eDirectory entry is modified. Example: Modify attributes of any user using ConsoleOne or iManager.

Trap Number	Trap Name	Trap Is Generated When
111	ndsNewSchemaEpoch	The schema is reset using DSRepair. Example: Create a new schema epoch using ndsrepair -S -Ad on UNIX.
112	ndsLowLevelSplitPartition	A low-level split is performed when a partition is being created. Example: Create a partition using ConsoleOne, iManager, or LDAP tools.
113	ndsReplicaInTransition	A replica is added or removed.
114	ndsAclModify	A trustee of an object is changed (an Access Control List (ACL) object is changed). Example: Add, modify, or delete a trustee of an object using LDAP tools, ICE, ConsoleOne, or iManager.
115	ndsLoginEnable	A request for enabling the user account is received by the server. Example: Enable the Account Disable attribute using LDAP tools, ICE, ConsoleOne, or iManager.
116	ndsLoginDisable	A request for disabling the user account is received by the server. Example: Disable the Account Disable attribute using LDAP tools, ICE, ConsoleOne, or iManager.
117	ndsDetectIntruder	A user account is locked out because of intruder detection. Example: Locked by Intruder attribute using LDAP tools, ICE, ConsoleOne, or iManager.
2001	ndsServerStart	The subagent successfully reconnects to the eDirectory server. This trap consists of two variables: <ul style="list-style-type: none"> ◆ ndsTrapTime: This variable contains the total number of seconds since midnight (12 a.m.) of 1 January 1970 GMT (UT), when the subagent successfully reconnected to the eDirectory server. ◆ ndsServerName: eDirectory server to which the subagent reconnected successfully. Example: Bring down and bring up the eDirectory server when the subagent is up and running.

Trap Number	Trap Name	Trap Is Generated When
2002	ndsServerStop	<p>The subagent loses its connection with the eDirectory server. This trap consists of two variables:</p> <ul style="list-style-type: none"> ♦ ndsTrapTime: This variable contains the total number of seconds since midnight (12 a.m.) of 1 January 1970 GMT (UT), when the subagent lost connection with the eDirectory server. ♦ ndsServerName: eDirectory server to which the subagent lost its connection. <p>Example:</p> <p>Bring down the eDirectory server when the subagent is up and running.</p>

Configuring Traps

The method of configuring traps differs from platform to platform.

Platform	Utility
NetWare	dssnmpsa
Windows	ndssnmpcfg
UNIX	ndssnmpconfig

NetWare

The utility to configure traps on NetWare is `dssnmpsa`. This utility is present in the `sys:\etc\` directory. Use this utility to enable and disable traps, set a time interval for individual traps, set a default time interval, enable traps for failure operations, and list all traps.

For help on the `dssnmpsa` usage, type **help dssnmpsa** at command line.

Usage:

`dssnmpsa trap commands`

For NetWare trap commands, see [“NetWare Trap Commands” on page 351](#).

NetWare Trap Commands

Trap Commands	Description	Usage
DISABLE	Disabling a trap refers to the NMS not receiving traps though they are generated.	<p>dssnmpsa "DISABLE <i>trapSpec</i>"</p> <p><i>trapSpec</i> can be any one of the following:</p> <p>To disable specific traps (for example, traps 10, 11, and 100):</p> <p>dssnmpsa "DISABLE 10, 11, 100"</p> <p>To disable all traps except 10, 11, and 100:</p> <p>dssnmpsa "DISABLE ID != 10, 11, 100"</p> <p>To disable all traps in the range 20 to 30:</p> <p>dssnmpsa "DISABLE 20-29"</p> <p>To disable all traps:</p> <p>dssnmpsa "DISABLE ALL"</p>
ENABLE	Enabling a trap refers to the NMS receiving traps when they are generated.	<p>dssnmpsa "ENABLE <i>trapSpec</i>"</p> <p><i>trapSpec</i> can be any one of the following:</p> <p>To enable specific traps (for example, traps 10, 11, and 100):</p> <p>dssnmpsa "ENABLE 10, 11, 100"</p> <p>To enable all traps except 10, 11, and 100:</p> <p>dssnmpsa "ENABLE ID != 10, 11, 100"</p> <p>To enable all traps in the range 20 to 30:</p> <p>dssnmpsa "ENABLE 20-29"</p> <p>To enable all traps:</p> <p>dssnmpsa "ENABLE ALL"</p>

Trap Commands	Description	Usage
INTERVAL	<p>This utility is used to set and view the time interval.</p> <p>The time interval determines how many seconds to delay before sending duplicate traps.</p> <p>The time interval should be between 0 and 2592000 seconds.</p> <p>If the time interval is out of range, then the default time interval is considered.</p> <p>If the time interval is set to zero, all the traps are sent.</p>	<p>To view the time interval: dssnmpsa "213,240,79 INTERVAL"</p> <p>To set the time interval between multiple traps (for example, to set the time interval between traps 12, 17, and 101 to 5): dssnmpsa "12 17 101 INTERVAL 5"</p> <p>To view the default time interval: dssnmpsa "DEFAULT INTERVAL"</p> <p>To set the default time interval: dssnmpsa "DEFAULT INTERVAL = 10"</p>
LIST	<p>Use this utility to view lists of trap numbers that meet specified criteria.</p>	<p>dssnmpsa LIST <i>trapSpec</i></p> <p><i>trapSpec</i> is used to specify groups of trap numbers and can be any of the following keywords:</p> <p>ALL, ENABLED, DISABLED, FAILED, or a logical expression</p> <p>Examples:</p> <p>To list all enabled traps along with trap names: dssnmpsa LIST ENABLED</p> <p>To list all disabled traps along with trap names: dssnmpsa LIST DISABLED</p> <p>To list all traps (117) along with trap names: dssnmpsa LIST ALL</p> <p>To list specific traps such as 12, 224, and 300 along with trap names: dssnmpsa LIST ID = 12,224,300</p> <p>To list all traps except selected traps such as 12, 224, and 300 along with trap names: dssnmpsa LIST ID != 12,224,300</p> <p>To list all traps that have been enabled for failure with trap names: dssnmpsa LIST FAILED</p>

Trap Commands	Description	Usage
READ_CFG	<p>Use this command to reconfigure the directory configuration from the configuration file <code>ndstrap.cfg</code>.</p> <p>Any changes specified in the configuration file will then take effect. This utility is primarily used to put various commands together in the <code>ndstrap.cfg</code> and do the operation in one instance.</p> <p>The <code>ndstrap.cfg</code> is located in <code>sys:\etc\</code>.</p> <p>The <code>ndstrap.cfg</code> file specifies operational parameters to be used for trap configuration and provides a way to configure the operation of SNMP traps. This file is read whenever the trap configuration utility, <code>dssnmpsa</code> is executed with the <code>READ_CFG</code> command.</p>	<code>dssnmpsa "READ_CFG"</code>
FAILURE	<p>This command is used to list all traps enabled for failure.</p> <p>Whenever an event fails, a failure trap is generated.</p> <p>NOTE: If the trap is enabled for failure and then disabled and again enabled using the <code>enable trapid</code> command, the trap is enabled for success and not for failure.</p>	<p><code>dssnmpsa "FAILURE <i>trapSpec</i>"</code></p> <p><i>trapSpec</i> consists of one or more trap numbers separated by commas or spaces, the keyword ALL, or a logical expression.</p> <p>Examples:</p> <p>To set failure for multiple traps:</p> <pre>dssnmpsa "FAILURE 10,11,100"</pre> <p>To set failure for all traps except the traps mentioned:</p> <pre>dssnmpsa "FAILURE ID != 24,30"</pre> <p>To set failure for all traps:</p> <pre>dssnmpsa "FAILURE ALL"</pre>

Windows

The utility to configure traps on Windows is `ndssnmpcfg`. This utility is present in the `install_path\snmp\` directory. Use this utility to enable and disable traps, set a time interval for individual traps, set a default time interval, enable traps for failure operations, and list all traps.

Usage:

```
ndssnmpcfg -h [hostname[:port]] -p password -a userFDN -c command
```

Parameter	Description
-h	DNS host name or IP address
-p	userFDN password for authentication

Parameter	Description
-a	Fully Distinguished Name of a user having administrative rights
-c	Trap Commands (See "Windows Trap Commands" on page 354.)

Windows Trap Commands

Trap Commands	Description	Usage
DISABLE	Disabling a trap refers to the NMS not receiving traps although they are being generated.	<p>To disable specific traps (for example, traps 10, 11, and 100):</p> <pre>ndssnmpcfg "DISABLE 10, 11, 100"</pre> <p>To disable all traps except 10, 11, and 100:</p> <pre>ndssnmpcfg "DISABLE ID != 10, 11, 100"</pre> <p>To disable all traps in the range 20 to 30:</p> <pre>ndssnmpcfg "DISABLE 20-29"</pre> <p>To disable all traps:</p> <pre>ndssnmpcfg "DISABLE ALL"</pre>
ENABLE	Enabling a trap refers to the NMS receiving traps when they are generated.	<pre>ndssnmpcfg "ENABLE trapSpec"</pre> <p><i>trapSpec</i> can be any one of the following:</p> <p>To enable specific traps (for example, traps 10, 11, and 100):</p> <pre>ndssnmpcfg "ENABLE 10, 11, 100"</pre> <p>To enable all traps except 10, 11, and 100:</p> <pre>ndssnmpcfg "ENABLE ID != 10, 11, 100"</pre> <p>To enable all traps in the range 20 to 30:</p> <pre>ndssnmpcfg "ENABLE 20-29"</pre> <p>To enable all traps:</p> <pre>ndssnmpcfg "ENABLE ALL"</pre>

Trap Commands	Description	Usage
INTERVAL	<p>This utility is used to set and view the time interval.</p> <p>The time interval determines how many seconds to delay before sending duplicate traps.</p> <p>The time interval set should be between 0 and 2592000 seconds.</p> <p>If the time interval set is out of range, then the default time interval is considered.</p> <p>If the time interval is set to zero, all the traps are sent.</p>	<p>To view the time interval: ndssnmpcfg "213,240,79 INTERVAL"</p> <p>To set the time interval between multiple traps (for example, to set the time interval between traps 12, 17, and 101 to 5): ndssnmpcfg "12 17 101 INTERVAL 5"</p> <p>To view the default time interval: ndssnmpcfg "DEFAULT INTERVAL"</p> <p>To set the default time interval: ndssnmpcfg "DEFAULT INTERVAL=10"</p>
LIST	<p>Use this utility to view lists of trap numbers that meet specified criteria.</p>	<p>ndssnmpcfg LIST <i>trapSpec</i></p> <p><i>trapSpec</i> is used to specify groups of trap numbers and can be any of the following keywords:</p> <p>ALL, ENABLED, DISABLED, FAILED, or a logical expression</p> <p>Examples:</p> <p>To list all enabled traps along with trap names: ndssnmpcfg LIST ENABLED</p> <p>To list all disabled traps along with trap names: ndssnmpcfg LIST DISABLED</p> <p>To list all traps (117) along with trap names: ndssnmpcfg LIST ALL</p> <p>To list specific traps like 12, 224, and 300 along with trap names: ndssnmpcfg LIST ID = 12,224,300</p> <p>To list all traps except selected traps like 12, 224, and 300 along with trap names: ndssnmpcfg LIST ID != 12,224,300</p> <p>To list all traps which have been enabled for failure with trap names: ndssnmpcfg LIST FAILED</p>

Trap Commands	Description	Usage
READ_CFG	<p>Use this command to reconfigure the directory configuration from the configuration file <code>ndstrap.cfg</code>.</p> <p>Any changes specified in the configuration file will then take effect. This utility is primarily used to put various commands together in the <code>ndstrap.cfg</code> and do the operation in one instance.</p> <p>The <code>ndstrap.cfg</code> is located in <i>install directory</i>\SNMP</p> <p>The <code>ndstrap.cfg</code> file specifies operational parameters to be used for trap configuration and provides a way to configure the operation of SNMP traps. This file is read whenever the trap configuration utility, <code>ndssnmpcfg</code> is executed with the <code>READ_CFG</code> command.</p>	<code>ndssnmpcfg "READ_CFG"</code>
FAILURE	<p>This command is used to list all traps enabled for failure.</p> <p>Whenever an event fails, a failure trap is generated.</p> <p>NOTE: If the trap is enabled for failure and then disabled and again enabled using the <code>enable trapid</code> command, the trap is enabled for success and not for failure.</p>	<p><code>ndssnmpcfg "FAILURE <i>trapSpec</i>"</code></p> <p><i>trapSpec</i> consists of one or more trap numbers separated by commas or spaces, the keyword ALL, or a logical expression.</p> <p>Examples:</p> <p>To set failure for multiple traps:</p> <pre>ndssnmpcfg "FAILURE 10,11,100"</pre> <p>To set failure for all traps except the traps mentioned:</p> <pre>ndssnmpcfg "FAILURE ID != 24,30"</pre> <p>To set failure for all traps:</p> <pre>ndssnmpcfg "FAILURE ALL"</pre>

UNIX

The utility to configure traps on UNIX is `ndssnmpconfig`. This utility is present in the `/etc/ndssnmp/` directory. Use this utility to enable and disable traps, set a time interval for individual traps, set a default time interval, enable traps for failure operations, and list all traps.

Usage:

```
ndssnmpconfig -h [hostname[:port]] -p password -a userFDN -c command
```

Parameter	Description
-h	DNS host name or IP address

Parameter	Description
-p	userFDN password for authentication
-a	Fully distinguished name of a user having administrative rights
-c	Trap commands (See "UNIX Trap Commands" on page 357.)

UNIX Trap Commands

Trap Commands	Description	Usage
DISABLE	Disabling a trap refers to the NMS not receiving traps though they are being generated.	<p>To disable specific traps (for example, traps 10, 11 and 100):</p> <pre>ndssnmpconfig "DISABLE 10, 11, 100"</pre> <p>To disable all traps except 10, 11, and 100:</p> <pre>ndssnmpconfig "DISABLE ID != 10, 11, 100"</pre> <p>To disable all traps in the range 20 to 30:</p> <pre>ndssnmpconfig "DISABLE 20-29"</pre> <p>To disable all traps:</p> <pre>ndssnmpconfig "DISABLE ALL"</pre>
ENABLE	Enabling a trap refers to the NMS receiving traps when they are generated.	<pre>ndssnmpconfig "ENABLE <i>trapSpec</i>"</pre> <p><i>trapSpec</i> can be any one of the following:</p> <p>To enable specific traps (for example, traps 10, 11, and 100):</p> <pre>ndssnmpconfig "ENABLE 10, 11, 100"</pre> <p>To enable all traps except 10, 11, and 100:</p> <pre>ndssnmpconfig "ENABLE ID != 10, 11, 100"</pre> <p>To enable all traps in the range 20 to 30:</p> <pre>ndssnmpconfig "ENABLE 20-29"</pre> <p>To enable all traps:</p> <pre>ndssnmpconfig "ENABLE ALL"</pre>

Trap Commands	Description	Usage
INTERVAL	<p>This utility is used to set and view the time interval.</p> <p>The time interval determines how many seconds to delay before sending duplicate traps.</p> <p>The time interval should be between 0 and 2592000 seconds.</p> <p>If the time interval is out of range, then the default time interval is considered.</p> <p>If the time interval is set to zero, all the traps are sent.</p>	<p>To view the time interval:</p> <pre>ndssnmpconfig "213,240,79 INTERVAL"</pre> <p>To set the time interval between multiple traps (for example, to set the time interval between traps 12, 17, and 101 to 5):</p> <pre>ndssnmpconfig "12 17 101 INTERVAL 5"</pre> <p>To view the default time interval:</p> <pre>ndssnmpconfig "DEFAULT INTERVAL"</pre> <p>To set the default time interval:</p> <pre>ndssnmpconfig "DEFAULT INTERVAL=10"</pre>
LIST	<p>Use this utility to view lists of trap numbers that meet specified criteria.</p>	<pre>ndssnmpconfig LIST <trapSpec></pre> <p><i>trapSpec</i> is used to specify groups of trap numbers and can be any of the following keywords:</p> <p>ALL, ENABLED, DISABLED, FAILED, or a logical expression</p> <p>Examples:</p> <p>To list all enabled traps along with trap names:</p> <pre>ndssnmpconfig LIST ENABLED</pre> <p>To list all disabled traps along with trap names:</p> <pre>ndssnmpconfig LIST DISABLED</pre> <p>To list all traps (117) along with trap names:</p> <pre>ndssnmpconfig LIST ALL</pre> <p>To list specific traps like 12, 224, and 300 along with trap names:</p> <pre>ndssnmpconfig LIST ID = 12,224,300</pre> <p>To list all traps except selected traps like 12, 224, and 300 along with trap names:</p> <pre>ndssnmpconfig LIST ID != 12,224,300</pre> <p>To list all traps that have been enabled for failure with trap names:</p> <pre>ndssnmpconfig LIST FAILED</pre>

Trap Commands	Description	Usage
READ_CFG	<p>Use this command to reconfigure the directory configuration from the configuration file <code>ndstrap.cfg</code>.</p> <p>Any changes specified in the configuration file will then take effect. This utility is primarily used to put various commands together in the <code>ndstrap.cfg</code> file and perform the operation in one instance.</p> <p>The <code>ndstrap.cfg</code> file is located in <code>/etc/ndssnmp/</code>.</p> <p>The <code>ndstrap.cfg</code> file specifies operational parameters to be used for trap configuration and provides a way to configure the operation of SNMP traps. This file is read whenever the trap configuration utility, <code>ndssnmpcfg</code> is executed with the <code>READ_CFG</code> command.</p>	<code>ndssnmpconfig "READ_CFG"</code>
FAILURE	<p>This command is used to list all traps enabled for failure.</p> <p>Whenever an event fails, a failure trap is generated.</p> <p>NOTE: If the trap is enabled for failure and then disabled and again enabled using the <code>enable trapid</code> command, the trap is enabled for success and not for failure.</p>	<p><code>ndssnmpconfig "FAILURE <i>trapSpec</i>"</code></p> <p><i>trapSpec</i> consists of one or more trap numbers separated by commas or spaces, the keyword ALL, or a logical expression.</p> <p>Examples:</p> <p>To set failure for multiple traps:</p> <pre>ndssnmpconfig "FAILURE 10,11,100"</pre> <p>To set failure for all traps except the traps mentioned:</p> <pre>ndssnmpconfig "FAILURE ID != 24,30"</pre> <p>To set failure for all traps:</p> <pre>ndssnmpconfig "FAILURE ALL"</pre>

Statistics

- ◆ [“ndsDbCache” on page 360](#)
- ◆ [“ndsDbConfig” on page 360](#)
- ◆ [“ndsProtolIfOps” on page 361](#)
- ◆ [“ndsServerInt” on page 363](#)

ndsDbCache

Managed Objects in Directory	Description
ndsDbSrvApplIndex	An index to uniquely identify the eDirectory Server Application.
ndsDbDibSize	Current size of the eDirectory Database in KB.
ndsDbBlockSize	Block size of the eDirectory Database in KB.
ndsDbEntryCacheMaxSize	Information on max size of the entry cache in KB.
ndsDbBlockCacheMaxSize	Information on max size of the block cache in KB.
ndsDbEntryCacheCurrentSize	Information on the current entry cache size.
ndsDbBlockCacheCurrentSize	Information on the current block cache size.
ndsDbEntryCacheCount	Information on the number of entries in the cache.
ndsDbBlockCacheCount	Information on the number of blocks in the cache.
ndsDbEntryCacheOldVerCount	Information on prior version entries in the cache.
ndsDbBlockCacheOldVerCount	Information on prior version blocks in the cache.
ndsDbEntryCacheOldVerSize	Information on prior version entry cache size.
ndsDbBlockCacheOldVerSize	Information on prior version block cache size.
ndsDbEntryCacheHits	Information on the number of entry hits.
ndsDbBlockCacheHits	Information on the number of block hits.
ndsDbEntryCacheHitLooks	Information on the number of entries examined to find hits.
ndsDbBlockCacheHitLooks	Information on the number of blocks examined to find hits.
ndsDbEntryCacheFaults	Information on the number of entry faults.
ndsDbBlockCacheFaults	Information on the number of block faults.
ndsDbEntryCacheFaultLooks	Information on the number of entries examined to determine misses.
ndsDbBlockCacheFaultLooks	Information on the number of blocks examined to determine misses.

ndsDbConfig

Managed Objects in Directory	Description
ndsDbCfgSrvApplIndex	An index to uniquely identify the eDirectory Server Application.

Managed Objects in Directory	Description
ndsDbCfgDynamicCacheAdjust	Information on whether Dynamic Cache Adjust is on or off. 0 = off 1 = on
ndsDbCfgDynamicCacheAdjustPercent	Information on the Dynamic Cache Adjust percentage parameter of available memory.
ndsDbCfgDynamicCacheAdjustMin	Information on the Dynamic Cache Adjust Minimum value parameter. This is cache size constraint values in KB.
ndsDbCfgDynamicCacheAdjustMinToLeave	Information on the Dynamic Cache Adjust Minimum value parameter in KB that is to be subtracted from the total available memory in KB.
ndsDbCfgHardLimitCacheAdjust	Information on whether Hard Limit Cache Adjust is on or off. 0 = off 1 = on
ndsDbCfgHardLimitCacheAdjustMax	Information on the cache maximum size in KB. This is a hard limit parameter.
ndsDbCfgBlockCachePercent	Information on the block cache percentage.
ndsDbCfgCacheAdjustInterval	Information on the cache adjust interval in seconds.
ndsDbCfgCacheCleanupInterval	Information on the cache cleanup interval in seconds.
ndsDbCfgPermanentSettings	Information on whether Permanent Settings is on or off. 0 = off 1 = on

ndsProtolfOps

Managed Objects in Directory	Description
ndsProtolfSrvApplIndex	An index to uniquely identify the eDirectory Server Application.
ndsProtolfIndex	An index to uniquely identify an entry corresponding to an eDirectory Server protocol interface.
ndsProtolfDescription	Information on the port being used by the DS protocol interface.
ndsProtolfUnauthBinds	Number of unauthenticated/anonymous bind requests received.
ndsProtolfSimpleAuthBinds	Number of bind requests that were authenticated using simple authentication procedures where the password is sent over the wire in encrypted or clear text format.

Managed Objects in Directory	Description
ndsProtolfStrongAuthBinds	Number of bind requests that were authenticated using SASL and X.500 strong authentication procedures. This includes the binds that were authenticated using external authentication procedures.
ndsProtolfBindSecurityErrors	Number of bind requests that have been rejected due to inappropriate authentication or invalid credentials.
ndsProtolfInOps	Number of requests received from DUAs or other eDirectory servers.
ndsProtolfReadOps	Number of read requests received.
ndsProtolfCompareOps	Number of compare requests received.
ndsProtolfAddEntryOps	Number of addEntry requests received.
ndsProtolfRemoveEntryOps	Number of removeEntry requests received.
ndsProtolfModifyEntryOps	Number of modifyEntry requests received.
ndsProtolfModifyRDNops	Number of modifyRDN requests received.
ndsProtolfListOps	Number of list requests received.
ndsProtolfSearchOps	Number of search requests (baseObject searches, oneLevel searches, and whole subtree searches) received.
ndsProtolfOneLevelSearchOps	Number of oneLevel search requests received.
ndsProtolfWholeSubtreeSearchOps	Number of whole subtree search requests received.
ndsProtolfExtendedOps	Number of extended operations.
ndsProtolfReferrals	Number of referrals returned in response to requests for operations.
ndsProtolfChainings	Number of operations forwarded by this eDirectory server to other eDirectory servers.
ndsProtolfSecurityErrors	Number of requests received that did not meet the security requirements.
ndsProtolfErrors	Number of requests that could not be serviced because of errors other than security errors and referrals. A partially serviced operation is not counted as an error. The errors include naming-related, update-related, attribute-related, and service-related errors.
ndsProtolfReplicationUpdatesIn	Number of replication updates fetched or received from eDirectory servers.
ndsProtolfReplicationUpdatesOut	Number of replication updates sent to or taken by eDirectory servers.
ndsProtolfInBytes	Incoming traffic, in bytes, on the interface. This includes requests from DUAs as well as responses from other eDirectory servers.

Managed Objects in Directory	Description
ndsProtolfOutBytes	Outgoing traffic, in bytes, on the interface. This includes responses to DUAs and eDirectory servers as well as requests to other eDirectory servers.

ndsServerInt

Managed Objects in Directory	Description
ndsSrvIntSrvApplIndex	An index to uniquely identify an eDirectory server application.
ndsSrvIntProtolfIndex	An index to uniquely identify an entry corresponding to an eDirectory server protocol interface.
ndsSrvIntIndex	Together with ndsSrvIntSrvApplIndex and ndsSrvIntProtolfIndex, this object forms the unique key to identify the conceptual row that contains useful information on the (attempted) interaction between the eDirectory server (referred to by applIndex) and a peer eDirectory server using a particular protocol.
ndsSrvIntURL	URL of the peer eDirectory server.
ndsSrvIntTimeOfCreation	The total number of seconds since midnight (12 a.m.) of 1 January 1970 GMT (UT) when this row was created.
ndsSrvIntTimeOfLastAttempt	The total number of seconds since midnight (12 a.m.) of 1 January 1970 GMT (UT) when the last attempt was made to contact the peer eDirectory server.
ndsSrvIntTimeOfLastSuccess	The total number of seconds since midnight (12 a.m.) of 1 January 1970 GMT (UT) when the last attempt made to contact the peer eDirectory server was successful.
ndsSrvIntFailuresSinceLastSuccess	The number of failures since the last time an attempt to contact the peer eDirectory server was successful. If there have been no successful attempts, this counter will contain the number of failures since this entry was created.
ndsSrvIntFailures	Cumulative failures in contacting the peer eDirectory server since the creation of this entry.
ndsSrvIntSuccesses	Cumulative successes in contacting the peer eDirectory server since the creation of this entry.

Troubleshooting

Log files are maintained to troubleshoot the problems that occur. These log files contain information about the errors that occur and can help you solve the problems.

See [“Troubleshooting SNMP” on page 527](#) for more details.

Platform	Subagent	Server	Master
NetWare	sys:\etc\ndssnmp.log, sys:\etc\snmpinst.log	NA	NA
Windows NT / 2000	<i>install_directory</i> \nds\ds snmpsa.log	<i>install_directory</i> \nds\ds snmpsrv.log	NA
Solaris	/var/nds/ ndssnmpsa.log	/var/nds/ndsd.log	/var/adm/messages
Linux	/var/nds/ ndssnmpsa.log	/var/nds/ndsd.log	/var/log/messages
AIX	/var/nds/ ndssnmpsa.log	/var/nds/ndsd.log	/usr/tmp/snmpd.log
HP-UX	/var/nds/ ndssnmpsa.log	/var/nds/ndsd.log	net-snmp-5.0.8 master agent: /usr/adm/snmpd.log NAA agent: /var/adm/ snmpd.log

14 Backing Up and Restoring Novell eDirectory

Novell® eDirectory™ is designed to provide fault tolerance for the tree through replication, so that if one server is not available, other servers can provide access. Replication is the primary method for protecting eDirectory.

Replication, however, is not possible in a single-server environment. Also, replication might not provide a complete restore of individual servers in case of a server hardware failure or other damage, or in the event of a disaster such as a fire or flood in which you lose multiple servers. Backing up eDirectory on each server increases the fault tolerance for your network.

eDirectory 8.7 introduced a new backup and restore utility called the eDirectory Backup eMTool to back up the eDirectory database on your individual servers. It has the following benefits:

- ◆ **Same tool for all platforms.**
- ◆ **Provides hot continuous backup.** You can back up your server without closing the eDirectory database, and you still get a complete backup.
- ◆ **Supports a quick restore of an individual server.** This is especially helpful in the event of hardware failure.
- ◆ **Scalable.** You can back up a server whose eDirectory database contains tens or hundreds of millions of objects. The speed of the backup process is limited mainly by I/O channel bandwidth.
- ◆ **Can support a quick restore of the tree, when used with replica planning and DSMASTER servers.** Even without using DSMASTER servers, some level of recovery for the tree should be possible. See [“Using DSMASTER Servers as Part of Disaster Recovery Planning” on page 377](#).
- ◆ **Lets you perform tasks remotely.** You can perform most backup and restore tasks in a browser using [iManager](#), inside or outside the firewall. You can perform advanced tasks remotely using the [eMBox Client](#), a command line Java client, with access behind the firewall or through a VPN.
- ◆ **Lets you back up related files.** You can back up files on the server that are related to the database, such as stream files, and any files you specify (such as `autoexec.ncf`) in an include file.
- ◆ **Can restore eDirectory to the state it was in at the moment before it went down,** if you use continuous roll-forward logging. See [“Using Roll-Forward Logs” on page 380](#).
- ◆ **Makes hardware upgrade simpler.** Doing a cold backup and then restoring the eDirectory database is an easy way to transfer the server’s identity to a new machine or safeguard it while you make changes such as RAM upgrades. See [“Upgrading Hardware or Replacing a Server” on page 447](#).
- ◆ **Works within the distributed nature of eDirectory.** You can ensure that a restored server matches the synchronization state that other servers in the tree expect by turning on continuous roll-forward logging.

- ◆ **Allows unattended backups.** You can create batch files to run unattended backups through the eMBox Client.

The new eDirectory Backup eMTool is designed to give you a complete backup and restore of the database and associated files on an individual server. It does not support backup and restore for individual objects or sections of the tree.

Also, it must be used in conjunction with file system backups to put the eDirectory backup files safely on tape.

This chapter contains the following topics:

- ◆ [“Checklist for Backing Up eDirectory” on page 366](#)
- ◆ [“Understanding Backup and Restore Services” on page 368](#)
- ◆ [“Using Roll-Forward Logs” on page 380](#)
- ◆ [“Preparing for a Restore” on page 384](#)
- ◆ [“Using Novell iManager for Backup and Restore” on page 386](#)
- ◆ [“Using the eMBox Client for Backup and Restore” on page 393](#)
- ◆ [“Using DSBK.NLM on NetWare” on page 409](#)
- ◆ [“Changes to Server-Specific Information Backup \(NetWare Only\)” on page 410](#)
- ◆ [“Recovering the Database If Restore Verification Fails” on page 411](#)
- ◆ [“Scenarios for Backup and Restore” on page 415](#)
- ◆ [“Backing Up and Restoring NICI” on page 421](#)

Checklist for Backing Up eDirectory

To make sure objects in a multiple-server tree are accessible even if a server is down:

- For multiple-server trees, ensure that all eDirectory partitions are replicated on more than one server, for fault tolerance.

For information on creating replicas, see [“Adding a Replica” on page 117](#).

To allow a quick and complete restore of individual servers (such as after a hardware failure):

- Do a full backup of the eDirectory database regularly (such as weekly).
- Do an incremental backup regularly (such as nightly).
- Do full and incremental tape backups of the file system shortly after full or incremental eDirectory database backups are completed.

Backup eMTool writes the backup files to a directory you specify on the server, but has no way of placing the data directly to tape. File system backup should be set to run after the eDirectory backup has run, to place the database backup files on tape for safe storage.

- Turn on and configure roll-forward logging, if it’s necessary in your environment.

You must turn on roll-forward logging for servers that participate in a replica ring. If you don’t, when you try to restore from your backup files you will get errors and the database will not open. The restore by default won’t open a database that shares replicas with other servers unless it is restored back to the state it was in at the moment before it went down.

In a single-server environment, roll-forward logging is not required for the restore verification process, but you can use it if you want to be able to restore eDirectory to the moment before it went down instead of just to the last backup.

Here is a list of the main issues you must address when you turn on roll-forward logging. For more information, see [“Using Roll-Forward Logs” on page 380](#).

- ◆ Specify a new location for the roll-forward logs (don't use the default).

The logs must be local to the server. For fault tolerance, they must not be stored on the same disk partition/volume or the same storage device as eDirectory. You might want a separate disk partition/volume just for roll-forward logs.

- ◆ Document where the roll-forward logs are placed, so that you can find them in the event of a failure.

To find out the location when the server is healthy, you can look it up in iManager in Backup Configuration, or in the eMBox Client using the getconfig option. But if the server has a failure that affects eDirectory (such as a hardware failure), you won't be able to look up the location of the roll-forward logs.

- ◆ Monitor disk space on the disk partition/volume where the roll-forward logs are stored, so that you can prevent it from filling up.

If roll-forward logs cannot be created because no more disk space is available, eDirectory will stop responding on that server.

- ◆ Restrict access to where the roll-forward logs are kept, so that unauthorized users cannot see them.
- ◆ If a restore is necessary, make sure you re-create the roll-forward log configuration on the server after the restore is complete. The settings are reset to the default during a restore. After turning on the roll-forward logs, you must also do a new full backup.

- ❑ For multiple-server trees, if you are using the Backup eMTool to back up a server, you should upgrade all the servers that share replicas with it to eDirectory 8.5 or later.

The restore verification process is backward compatible only with 8.5 or later. For more information about the restore verification, see [“Overview of How the Backup eMTool Does a Restore” on page 371](#) and [“Restore Verification Is Backward Compatible Only with eDirectory 8.5 or Later” on page 378](#).

- ❑ (NetWare[®] only) Review the issues with file system rights in [“Preserving Rights When Restoring File System Data on NetWare” on page 379](#). Test for potential problems and take preventive action if necessary.
- ❑ Periodically check the backup log file to make sure that unattended backups were successful.
- ❑ Do a cold backup before upgrading a server, as described in [“Upgrading Hardware or Replacing a Server” on page 447](#).
- ❑ For multiple-server trees, ensure that all eDirectory partitions are replicated on more than one server, for fault tolerance.

In addition to making objects available when a server is down, such as during maintenance, replicating your partitions also provides fault tolerance in a case where you lose a server, such as a hardware failure. If a server in a multiple-server tree holds a partition that is not replicated, and the server has a failure, there's a risk that you might not be able to recover the partition. It's best to make sure all partitions are replicated. For more information on why you might not be able to recover an unreplicated partition in a multiple-server tree, see [“Overview of How the Backup eMTool Does a Restore” on page 371](#), [“Using Roll-Forward Logs” on page 380](#), and [“Recovering the Database If Restore Verification Fails” on page 411](#).

For information on replication, see [“Replicas” on page 49](#) and [Chapter 5, “Managing Partitions and Replicas,” on page 113](#).

- Ensure that the backup tapes containing the eDirectory and file system backups are in a safe location.
- Regularly test your backup strategy to make sure it meets your goals.
- (Optional) If you plan to access servers remotely to do cold backups (a full backup with the database closed) or to do advanced backup and restore tasks, install the eMBox Client on the machine you plan to use. Also, arrange for access (such as VPN access) behind the firewall.

iManager lets you do backup and restore tasks remotely, outside the firewall, but it does not support cold backup and advanced tasks.

The eMBox Client is installed with eDirectory on the server, and you can also use it on workstations with Sun JVM 1.3.1. For information on installing and configuring the eMBox Client, see [“Using the eMBox Command Line Client” on page 465](#).

To prepare for a disaster in which you lose multiple servers:

- Address the issues listed above.
- For multiserver trees, consider creating DSMASTER servers to help you prepare for the event of a disaster.

See [“Using DSMASTER Servers as Part of Disaster Recovery Planning” on page 377](#).

- Regularly test your disaster recovery strategy to make sure it meets your goals.

Understanding Backup and Restore Services

- ◆ [“About the eDirectory Backup eMTool” on page 369](#)
- ◆ [“What’s Different about Backup and Restore in eDirectory 8.7.3?” on page 370](#)
- ◆ [“Overview of How the Backup eMTool Does a Restore” on page 371](#)
- ◆ [“Format of the Backup File Header” on page 372](#)
- ◆ [“Format of the Backup Log File” on page 376](#)
- ◆ [“Using DSMASTER Servers as Part of Disaster Recovery Planning” on page 377](#)
- ◆ [“Transitive Vectors and the Restore Verification Process” on page 378](#)
- ◆ [“Restore Verification Is Backward Compatible Only with eDirectory 8.5 or Later” on page 378](#)
- ◆ [“Preserving Rights When Restoring File System Data on NetWare” on page 379](#)

About the eDirectory Backup eMTool

The Backup eMTool provides hot continuous backup of the eDirectory database on an individual server. You can back up eDirectory on your server without closing the database, and you still get a complete backup that is a snapshot of the moment when the backup began. This feature means that you can create a backup at any time and eDirectory will be accessible throughout the process. (Hot continuous backup is the default behavior—you can specify a “cold” backup with the database closed, if required.)

The new backup also lets you turn on roll-forward logging to keep a record of transactions in the database since the last backup, so you can restore a server to the state it was in at the moment before it went down. You must turn on roll-forward logging for servers that participate in a replica ring, so that you can restore a server back to the synchronization state that the other servers expect. If you don't, when you try to restore from your backup files you will get errors and the database will not open. Roll-forward logging is off by default. For more information, see [“Using Roll-Forward Logs” on page 380](#).

The new Backup eMTool does not back up all the objects in eDirectory at once; just the partitions on an individual server. This allows for better restore of an individual server and faster backups than the legacy TSA for NDS[®] backup. (The legacy TSA for NDS backup still works as documented in eDirectory 8.6; both the TSA for NDS and the new backup can be used if necessary.) For a comparison, see [“What's Different about Backup and Restore in eDirectory 8.7.3?” on page 370](#).

The new eDirectory backup tool must be used in conjunction with file system backups to put the eDirectory backup files safely on tape. Novell has partnered with several leading providers of backup solutions. For a list, see [NetWare Partner Products: Backup, Restore, & Recovery \(http://www.novell.com/partnerguid/p100004.html\)](http://www.novell.com/partnerguid/p100004.html).

On NetWare, you also might need to use the eDirectory backup tool in conjunction with backups of file system rights. For more information, see [“Preserving Rights When Restoring File System Data on NetWare” on page 379](#).

In iManager, you can use all the features except cold backup, unattended backup, and advanced restore options, as explained in [“Using Novell iManager for Backup and Restore” on page 386](#). All backup and restore tasks including unattended backups can be done using the eMBox Java command line client, as explained in [“Using the eMBox Client for Backup and Restore” on page 393](#).

For a description of the backup and restore options in iManager, see the online help. For a description of the eMBox Client options, see [“Backup and Restore Command Line Options” on page 403](#).

For a description of what the tool does during a restore, see [“Overview of How the Backup eMTool Does a Restore” on page 371](#).

The eDirectory Backup eMTool is part of the eMBox tool set. The eMBox is a service that is installed on the server as part of eDirectory.

The Backup eMTool comprises the following files:

Filename	Description
backupcr	Core library that contains all backup and restore functionality. This library has no user interface; it is loaded and linked dynamically by the backupfl program.

Filename	Description
backuptl	eMTool interface to the backupcr library. Provides backup and restore functionality through the eMBox architecture. This can be accessed via the iManager plug-in or the eMBox Client, the Java command line client.
dsbackup_en.xlf	Language file containing messages returned by the Backup eMTool.

For a description of the format for the backup files and log files that the Backup eMTool creates, see [“Format of the Backup Log File” on page 376](#) and [“Format of the Backup File Header” on page 372](#).

IMPORTANT: The restore verification process is backward compatible only with eDirectory 8.5 or later. If you want to use the new backup and restore on servers that participate in a replica ring, make sure you upgrade all the servers in the replica ring to eDirectory 8.5 or later. (See also [“Restore Verification Is Backward Compatible Only with eDirectory 8.5 or Later” on page 378](#).)

What’s Different about Backup and Restore in eDirectory 8.7.3?

In previous versions of eDirectory, backup and restore was focused on backing up the tree, object by object.

The new Backup eMTool in eDirectory 8.7 introduced a completely new focus and new architecture. It’s server-centric, not tree-centric; you back up the eDirectory database on each server individually. It’s much faster than the legacy TSA for NDS backup.

The legacy TSA for NDS backup tool can still be used to back up the tree, although we encourage you to use the new backup.

Backup of server-specific information has been implemented using the Backup eMTool. See [“Changes to Server-Specific Information Backup \(NetWare Only\)” on page 410](#).

For more comparison information, see the following table.

Issue	Legacy TSA for NDS Backup	Backup eMTool “Hot Continuous Backup”
Focus	Designed to back up the tree, object by object. For more information about the legacy backup utilities (that still work in 8.7 - both kinds of backup can be used if necessary), see “Backing Up and Restoring Novell eDirectory” (http://www.novell.com/documentation/lg/ndsedir86/taoenu/data/a2n4mb6.html) in the <i>Novell eDirectory 8.6 Administration Guide</i> .	Designed to back up the eDirectory database on each server individually. Fault tolerance for the whole tree should be provided primarily by replication, but backing up each server provides additional fault tolerance. When planning a restore strategy for the tree after a disaster in which many servers are lost, consider using DSMASTER servers and replica planning as outlined in “Using DSMASTER Servers as Part of Disaster Recovery Planning” on page 377 .
Speed	N/A	Significantly improved. Speed is one of the most important features of the new backup.

Issue	Legacy TSA for NDS Backup	Backup eMTool “Hot Continuous Backup”
Where the backup is placed	Allows backup to be placed directly to tape.	Places the backup files on the file system. You must use a file system backup to put them on tape for safe storage.
Cross-platform	Performs differently on each platform.	Works the same way on each platform.
Ability to restore individual servers	Not designed to provide this.	Provides the ability to restore an individual server after a hard drive failure or to use backup to move a server from one machine to another. Provides the option to use roll-forward logging so you can restore a server to the state it was in at the moment before it went down, so it is in the synchronization state expected by other servers in a replica ring. Has the ability to back up files related to eDirectory on an individual server. You can also create your own list of related files to include with the backup.
Roll-forward logging for an individual server	Not designed to provide this.	Lets you keep a record of transactions in the database since the last backup, so you can restore a server to the state it was in at the moment before it went down. In a multiple-server environment, this allows you to restore a server to the synchronization state that the other servers expect. Roll-forward logging is off by default. For more information, see “Using Roll-Forward Logs” on page 380 .

Overview of How the Backup eMTool Does a Restore

Before restoring, you need to collect all your backup files by following the instructions in [“Preparing for a Restore” on page 384](#). When you direct the Backup eMTool to begin the restore through iManager or the eMBox Client, the process is done by the Backup eMTool as follows:

1. The DS Agent is closed.
2. The active DIB (Data Information Base) set is switched from the DIB set named NDS to a new DIB set named RST.

(The existing NDS database is left on the server; if the restore verification fails it will once again become the active DIB set.)
3. The restore is performed, restoring to the DIB set named RST.
4. The DIB set is disabled.

The login disabled attribute is set on the pseudo server, preventing the DS Agent from being able to open using this DIB set.
5. The roll-forward log settings are reset to the default.

This means that after a restore, roll-forward logging on the server is always set to off, and the location of the roll-forward logs is reset to the default.

(If you want roll-forward logging turned on for this server, you must plan to re-create your configuration for roll-forward logging after a restore, to make sure it is turned on and the logs are being saved in a fault-tolerant location. After turning on the roll-forward logs, you must also do a new full backup.)

6. Verification of the restored RST database is performed.

The server attempts to verify the consistency of the data that has been restored. It does this by contacting every server that it shares a replica with and comparing the transitive vectors.

The output from this verification process is printed in the log file.

If the transitive vector on the remote server is ahead of the local vector, then data is missing from the restore, and the verification fails.

Here is an example of the information that's recorded in the log file if verification fails for one of the replicas, showing the transitive vectors that were compared:

```
Server: \T=LONE_RANGER\O=novell\CN=CHIP
Replica: \T=LONE_RANGER\O=novell
Status: ERROR = -6034
      Local TV          Remote TV
s3D35F377 r02 e002    s3D35F3C4 r02 e002
s3D35F370 r01 e001    s3D35F370 r01 e001
s3D35F363 r03 e001    s3D35F363 r03 e001
s3D35F31E r04 e004    s3D35F372 r04 e002
s3D35F2EE r05 e001    s3D35F2EE r05 e001
s3D35F365 r06 e003    s3D35F365 r06 e003
```

For more information, see [“Transitive Vectors and the Restore Verification Process” on page 378](#).

7. If verification is successful, RST is renamed to NDS and the login disabled attribute is cleared so it becomes the active eDirectory database on the server. If verification fails, the RST DIB is not renamed, and the active DIB set is set back to NDS.

If verification fails, see [“Recovering the Database If Restore Verification Fails” on page 411](#) for how to recover the server. (It's possible to force the RST database to be activated and unlocked using [advanced restore options](#), but this is not recommended unless suggested by Novell Support.)

Format of the Backup File Header

The backup files contain a header that you can read to learn important information such as

- ◆ The filename of the backup file when it was created.

This is helpful if the filename has been changed since the backup was created.

- ◆ The current roll-forward log at the time of this backup.

If this is the last backup in the set you are restoring from (such as the last incremental backup in a set of one full backup and three incremental backups), this helps you because it indicates the first roll-forward log that you need for a complete restore.

- ◆ The replicas this server held.

This is helpful if you did not have the placement of your replicas documented. If you experienced a disaster in which many servers were lost, the list of replicas shown in the backup file header might help you decide which servers to restore first.

- ◆ The names of the files that were included in the backup as specified in a user include file.
- ◆ The number of files in the backup set for that backup.

The header of the backup file for each individual backup is in XML format. Immediately following the header is the backup data from the database in binary code. (Because of the inclusion of binary data at the end of the file, parsing the file would give errors, but the XML header complies with XML standards.) In cases where the backup spanned more than one file, the header information is included in each file in the set.

WARNING: When opening a backup file, just view the header—make sure you don't try to save or modify the file, or it might become truncated. Most applications can't save the binary data correctly.

The following is the DTD for the XML header. (The DTD is included as part of the header in the backup file as well, for your reference.)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE backup [
<!ELEMENT backup (file|replica)*>
<!ELEMENT file (#PCDATA)>
<!ELEMENT replica EMPTY>
<!ATTLIST backup version CDATA #REQUIRED
  backup_type (full|incremental) #REQUIRED
  idtag CDATA #REQUIRED
  time CDATA #REQUIRED
  srvname CDATA #REQUIRED
  dsversion CDATA #REQUIRED
  compression CDATA "none"
  os CDATA #REQUIRED
  current_log CDATA #REQUIRED
  number_of_files CDATA #IMPLIED
  backup_file CDATA #REQUIRED
  incremental_file_ID CDATA #IMPLIED
  next_inc_file_ID CDATA #IMPLIED>
<!ATTLIST file size CDATA #REQUIRED
  name CDATA #REQUIRED
  encoding CDATA "base64"
  type (user|nici) #REQUIRED>
<!ATTLIST replica partition_DN CDATA #REQUIRED
  modification_time CDATA #REQUIRED
  replica_type (MASTER|SECONDARY|READONLY|SUBREF|
  SPARSE_WRITE|SPARSE_READ|Unknown) #REQUIRED
  replica_state (ON|NEW_REPLICA|DYING_REPLICA|LOCKED|
  CRT_0|CRT_1|TRANSITION_ON|DEAD_REPLICA|
  BEGIN_ADD|MASTER_START|MASTER_DONE|
  FEDERATED|SS_0|SS_1|JS_0|JS_1|MS_0|MS_1|
  Unknown) #REQUIRED>
]>
```

The following table explains the attributes in the DTD.

Attribute	Explanation
backup version	Version of the Backup tool.
backup backup_type	Type of backup being performed, either full or incremental. (A cold backup is a full backup.)
backup idtag	A GUID based on the time of backup. This helps in identifying the backup, even if the filename of the backup file is changed.

Attribute	Explanation
backup time	Date and time the backup was started.
backup srvname	Distinguished name of the server being backed up.
backup dsversion	eDirectory version running on the server.
backup compression	Whether the Backup eMTool has used compression on the backup data. This only applies to the backup data; the header itself will never be compressed.
backup os	Operating system the backup was performed on. We recommend that you restore only to the same operating system.
backup current_log	First roll-forward log that is required when restoring this backup. This helps you collect the correct set of files for a restore.
backup number_of_files	Number of files in the backup set. This value appears only in the first backup file.
backup backup_file	Filename of the current backup. If the backup spans multiple files, then the header for each file will show the filename including a number appended to show its order in the set. For an example of the filenames in a set of backup files, see <i>-s file_size</i> .
backup incremental_file_ID	If this is an incremental backup, this attribute shows the ID of the incremental file.
backup next_inc_file_ID	The ID that the next incremental backup will have when it is created. This helps you collect the correct set of files for a restore.
file size	Size of the data between the <file> tags for this file.
file name	Name and location of the file when it was backed up.
file encoding	The encoding algorithm used on the file.
file type	Indicates whether the file is a NICI file or a user included file.
replica partition_DN	Distinguished name of the partition. This is helpful if you did not have the placement of your replicas documented. If you experienced a disaster in which many server were lost, the list of replicas shown in the backup file header might help you decide which servers to restore first.
replica modification_time	Transitive vector for this replica at the time of the backup.
replica replica_type	Type of replica, such as master or read-only.
replica_state	State of the replica at the time of the backup, such as On or New Replica.

The following is an example of a backup file header from a Windows NT server:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE backup [
<!ELEMENT backup (file|replica)*>
<!ELEMENT file (#PCDATA)>
<!ELEMENT replica EMPTY>
<!ATTLIST backup version CDATA #REQUIRED
  backup_type (full|incremental) #REQUIRED
  idtag CDATA #REQUIRED
  time CDATA #REQUIRED
  srvname CDATA #REQUIRED
  dsversion CDATA #REQUIRED
  compression CDATA "none"
  os CDATA #REQUIRED
  current_log CDATA #REQUIRED
  number_of_files CDATA #IMPLIED
  backup_file CDATA #REQUIRED
  incremental_file_ID CDATA #IMPLIED
  next_inc_file_ID CDATA #IMPLIED>
<!ATTLIST file size CDATA #REQUIRED
  name CDATA #REQUIRED
  encoding CDATA "base64"
  type (user|nici) #REQUIRED>
<!ATTLIST replica partition_DN CDATA #REQUIRED
  modification_time CDATA #REQUIRED
  replica_type (MASTER|SECONDARY|READONLY|SUBREF|
  SPARSE_WRITE|SPARSE_READ|Unknown) #REQUIRED
  replica_state (ON|NEW_REPLICA|DYING_REPLICA|LOCKED|
  CRT_0|CRT_1|TRANSITION_ON|DEAD_REPLICA|
  BEGIN_ADD|MASTER_START|MASTER_DONE|
  FEDERATED|SS_0|SS_1|JS_0|JS_1|MS_0|MS_1|
  Unknown) #REQUIRED>
]>

<backup version="2" backup_type="full" idtag="3D611DA2" time="2002-8-
19'T10:32:35" srvname="\T=MY_TREE\O=novell\CN=DSUTIL-DELL-NDS"
dsversion="1041081" compression="none" os="windows"
current_log="00000003.log" next_inc_file_ID="2" number_of_files="0000001"
backup_file="c:\backup\header.bak">

<replica partition_DN="\T=MY_TREE" modification_time="s3D611D95_r1_e2"
replica_type="MASTER" replica_state="ON" />

<replica partition_DN="\T=MY_TREE\O=part1"
modification_time="s3D611D95_r1_e2" replica_type="MASTER" replica_state="ON"
/>

<replica partition_DN="\T=MY_TREE\O=part2"
modification_time="s3D611D95_r1_e2" replica_type="MASTER" replica_state="ON"
/>

<replica partition_DN="\T=MY_TREE\O=part3"
modification_time="s3D611D96_r1_e2" replica_type="MASTER" replica_state="ON"
/>

<file size="190" name="C:\WINNT\system32\novell\nici\bhawkins\XARCHIVE.001"
encoding="base64" type="nici">the data is included here</file>

<file size="4228" name="C:\WINNT\system32\novell\nici\bhawkins\XMGRCFG.KS2"
encoding="base64" type="nici">the data is included here</file>

<file size="168" name="C:\WINNT\system32\novell\nici\bhawkins\XMGRCFG.KS3"
encoding="base64" type="nici">the data is included here</file>

```

```

<file size="aaac" name="C:\WINNT\system32\novell\nici\nicintacl.exe"
encoding="base64" type="nici">the data is included here</file>

<file size="150" name="C:\WINNT\system32\novell\nici\NICISDI.KEY"
encoding="base64" type="nici">the data is included here
</file>

<file size="4228" name="C:\WINNT\system32\novell\nici\system\Xmgrcfg.ks2"
encoding="base64" type="nici">the data is included here
</file>

<file size="168" name="C:\WINNT\system32\novell\nici\system\Xmgrcfg.ks3"
encoding="base64" type="nici">the data is included here
</file>

<file size="1414" name="C:\WINNT\system32\novell\nici\xmgrcfg.wks"
encoding="base64" type="nici">the data is included here
</file>

</backup>

```

After the header, the binary data for the backup of the database is included in the backup file.

Format of the Backup Log File

The eDirectory Backup eMTool keeps a log that shows a high-level view of Backup eMTool activity, containing information about previous backups. The log file contains a history of all backups, records backup start time and end time, and contains information about possible errors during the backup process. This file is appended with each backup. It is also placed in a location you specify.

It is useful for reviewing whether unattended backups were successful. The success or failure and the error code are displayed on the last line.

The Backup eMTool log file also gives the ID of backups that have been done, which helps you gather the correct set of full and incremental backup files for a restore. The first four lines are duplicates of information in the header of the backup file.

Also recorded in the log file are other files that were included in the backup of the database, such as the files you specified in an include file.

For a restore, it will record the included files that were restored.

The following are two examples of log file entries:

```

|=====DSBackup Log: Backup=====|
Backup type: Full
Log file name: sys:/backup/backup.log
Backup started: 2002-6-21'T19:53:5GMT
Backup file name: sys:/backup/backup.bak
Server name: \T=VIRTUALNW_TREE\O=novell\CN=VIRTUALNW
Current Roll Forward Log: 00000001.log
DS Version: 1041072
Backup ID: 3D138421
Starting database backup...
Database backup finished
Completion time 00:00:03
Backup completed successfully

```



```
|=====DSBackup Log: Restore=====|
Log file name: sys:/save/doc.log
Restore started: 2002-7-19'T19:1:34GMT
Restore file name: sys:/backup/backup.bak
Starting database restore...
Restoring file sys:/backup/backup.bak
Database restore finished
Completion time 00:00:15
Restore completed successfully
```

Using DSMASTER Servers as Part of Disaster Recovery Planning

If you have a multiple-server environment and want to plan for recovery after a disaster in which all your servers are lost, you can use DSMASTER servers as part of the plan for your tree.

The Backup eMTool is used to back up each server separately; it is server-centric, not tree-centric. However, if you create DSMASTER servers, you can use Backup eMTool functionality specifically to back up your whole tree structure. An example of this strategy is outlined in [“Scenario: Losing All Servers in a Multiple-Server Environment” on page 419](#).

When restoring after a disaster, one of the main concerns is how to avoid restoring replicas of the same partition that are inconsistent with each other. If you lose roll-forward logs for your servers as part of a disaster, you won't be able to restore all your servers to the same moment in time. Without the roll-forward logs, the replicas you have in your backups are inconsistent with each other and would cause problems if they were all restored and brought into the tree together. (The restore verification process is designed to help prevent these problems; by default a restored eDirectory database will not open after the restore if it is inconsistent with the other replicas.)

You can use DSMASTER servers to help you prepare for this issue, by creating a master copy of your tree that you could use as a starting point.

To use DSMASTER servers to help prepare for a disaster:

- ◆ Plan your replicas so that you have one server that contains a replica of every partition in your tree, so a copy of the whole tree is in the eDirectory database on one server (or, if your tree is large, you can use a couple of key servers). This kind of server is often called a DSMASTER server. The replicas on the DSMASTER server should be master or read/write replicas.

NOTE: If a couple of key DSMASTER servers are used instead of just one, keep in mind that ideally each DSMASTER server should have a unique set of replicas of partitions. There should be no overlap between them, to avoid inconsistencies between the replicas when restoring after a disaster.

If your servers were lost in a disaster, you would not have access to the most recent roll-forward logs for restoring because roll-forward logs are saved locally on the server, so all the DSMASTER servers probably could not be restored to the same moment in time. If the same replica were held on two DSMASTER servers, the two copies would probably not be identical and would cause inconsistencies in the tree. So, for disaster recovery planning it's best to not have the same partition replicated on more than one DSMASTER server.

For general information on replicas, see [“Replicas” on page 49](#).

- ◆ Back up these DSMASTER servers regularly to create a backup copy of your tree. You might want to take extra precautions for storing the backups of DSMASTER servers as part of your disaster recovery plan.

If your tree is designed this way, in the event of a disaster you could get your tree structure up and running again quickly by restoring just that one server (or small group of key servers) and making sure the replicas it holds are designated as the master replicas.

After your tree structure is responding again, you could then move to the task of restoring other servers that were lost, using just the full and incremental backup files. Because you don't have the roll-forward logs, the verification of the restore process will fail for these other servers. To bring them back into the tree, you would remove them from the replica ring, change all their replica information to external references using DSRRepair, and then re-add the replicas to the servers using replication from the copy on the DSMASTER server. These steps are documented in [“Recovering the Database If Restore Verification Fails” on page 411](#).

If a disaster occurs in which you lose many servers but not all, the issues with replicas will probably be complex, and you should contact Novell Support.

Transitive Vectors and the Restore Verification Process

A transitive vector is a time stamp for a replica. It is made up of a representation of the number of seconds since a common specific point in history (January 1, 1970), the replica number, and the current event number. Here's an example:

```
s3D35F377 r02 e002
```

In the context of backup and restore, it's important because the transitive vector is used to verify that the server restored is in sync with the replica rings it participates in.

Servers that hold replicas of the same partition communicate with each other to keep the replicas synchronized. Each time a server communicates with another server in the replica ring, it keeps a record of the transitive vector the other server had when they communicated. These transitive vectors allow the servers in a replica ring to know what information needs to be sent to each replica in the ring to keep all the replicas synchronized. When a server goes down, it stops communicating, and the other servers don't send updates or change the transitive vector they have recorded for that server until the server starts communicating again.

When you restore eDirectory on a server, the restore verification process compares the transitive vector of the server being restored to the other servers in the replica ring. This is done to make sure that the replicas being restored are in the same state that the other servers expect.

If the transitive vector on the remote server is ahead of the local vector, then data is missing from the restore, and the verification fails. (For example, data might be missing because you did not turn on continuous roll-forward logging before the last full or incremental backup, you did not include the roll-forward logs in the restore, or the set of roll-forward logs you provided for the restore was not complete.)

By default the restored eDirectory database is not opened if it is inconsistent with the other replicas.

For an example of the log file entry when transitive vectors don't match, see [“Overview of How the Backup eMTool Does a Restore” on page 371](#).

For a description of compatibility issues that could cause the restore verification to fail, see [“Restore Verification Is Backward Compatible Only with eDirectory 8.5 or Later” on page 378](#).

For information on what to do if the restore verification fails, see [“Recovering the Database If Restore Verification Fails” on page 411](#).

Restore Verification Is Backward Compatible Only with eDirectory 8.5 or Later

The restore verification process is backward compatible only with eDirectory 8.5 or later. If the server you are restoring participates in a replica ring with a server running an earlier version than

eDirectory 8.5, the restore log will show a -666 error (incompatible DS version) for that replica. This does not indicate whether the replicas are out of sync; it merely indicates that the restore verification was unable to compare the transitive vectors because the version of eDirectory was earlier than 8.5.

By default the database will not open because the restore verification was not completely successful. However, you can use your best judgement in this case; if the only error was from an 8.5 server, and the other servers verified successfully, it might be safe to open the database, using the override restore option in the eMBox Client.

Another option might be to remove the server with the older version from the replica ring, and try the restore again.

For more information about the restore process and transitive vectors, see [“Overview of How the Backup eMTool Does a Restore” on page 371](#) and [“Transitive Vectors and the Restore Verification Process” on page 378](#).

Preserving Rights When Restoring File System Data on NetWare

On NetWare only, restoring file system rights (also called trustee assignments) is dependent on the object that is the trustee being present in eDirectory. Because of this relationship, you need to use caution when restoring eDirectory and file system data on NetWare, to preserve file system rights.

If you restore eDirectory *before* restoring file system data, file system rights should be preserved when file system data is restored. However, you should be aware of the issue. Test for potential problems and take preventive action if necessary.

Why a Restore Could Potentially Affect File System Rights

As part of your preparation to restore eDirectory, you need to do a new installation of eDirectory creating a new temporary tree, either on a new storage device to replace a failed one that held volume sys:, or on a new machine if you are migrating a server from one machine to another.

A brand-new installation of eDirectory will not contain the objects that trustee rights have been assigned for. (Of course, the objects will be restored when eDirectory is restored.)

When file system data is restored, the file system restore looks for the trustee objects in eDirectory. If an object which is a trustee does not exist in the eDirectory database (such as in a new installation before eDirectory has been restored), it's possible that rights assignments for that object might be removed from the file system.

How to Address the Issue If Necessary

You can address the potential issues with restores and file system rights/trustee assignments in a few different ways:

- ◆ Most importantly, restore eDirectory before restoring the file system.

You can do a new installation and restore eDirectory without taking any special measures, and after eDirectory is restored, you can plan to do a file system restore for any files you need the file system rights/trustee assignments to be recovered for.

- ◆ As part of your backup strategy, you can use `trustbar.nlm` to back up and restore file system rights/trustee assignments, or other third-party software that lets you do the same thing. This way, you can restore trustee assignments to the file system if necessary, after eDirectory is restored.

You could schedule backups of the file system rights/trustee assignments at intervals similar to the schedule you use to back up eDirectory and the file system.

NOTE: You can schedule the backup of file system rights using third-party scheduling software or [cron.nlm](http://support.novell.com/servlet/tidfinder/2939440) (<http://support.novell.com/servlet/tidfinder/2939440>), available from the Novell Support Web site.

- ◆ You can reconfigure your storage system to reduce the probability of failures that require restoring eDirectory and file system data. For example, by using a RAID system or another configuration, you can reduce the chances of loss of data if an individual storage device fails. If you have a redundant sys: volume and suffer a device failure, it's more likely that a new installation of eDirectory and a file system restore would not be necessary.
- ◆ If you restore the file system data before eDirectory for some reason, and you lose rights, you can do the file system restore again after restoring eDirectory.
- ◆ You can make sure that no volumes except sys: are mounted until eDirectory is restored, such as in a case where a storage device failure affects the sys: volume but other storage devices on the server are still functioning.

One way to ensure that volumes will not be mounted is to disconnect the storage devices from the server before the new installation of NetWare and eDirectory, and then reconnect them after the eDirectory restore is complete.

After eDirectory is restored, if necessary you could do a file system restore of sys: to recover rights on the sys: volume.

Using Roll-Forward Logs

Roll-forward logging is similar to journaling on other database products. The roll-forward logs (RFLs) are a record of all changes to the database.

The advantage of using roll-forward logging is that the roll-forward logs give you a history of changes since the last full or incremental backup, so you can restore eDirectory to the state it was in at the moment before a failure. Without roll-forward logs, you can restore eDirectory only to the point of the last full or incremental backup.

eDirectory creates a record of transactions in a log file before committing them to the database. By default, the log file for these records is reused over and over (consuming only a small amount of disk space), and the history of changes to the eDirectory database is not being saved.

When you turn on continuous roll-forward logging, the history of changes is saved in a set of consecutive roll-forward log files. Roll-forward logging does not reduce server performance; it simply saves the log file entries that eDirectory is already creating.

You must turn on roll-forward logging for servers that participate in a replica ring. If you don't, when you try to restore from your backup files you will get errors and the database will not open. The restore by default won't open a database that shares replicas with other servers unless it is restored back to the state it was in at the moment before it went down. (If you don't have roll-forward logs, you must follow a separate procedure to try to recover, described in [“Recovering the Database If Restore Verification Fails” on page 411](#).)

Roll-forward logging is off by default. You must turn it on if you want to use it on a server. Roll-forward logging is also turned off and the settings returned to default when you restore a server, so after a restore you must turn it on again, re-create your configuration, and create a new full backup. (The new full backup is necessary so that you are prepared for any failures that might occur before the next unattended full backup is scheduled to take place.)

In a single-server environment, roll-forward logging is not required, but you can use it if you want to be able to restore eDirectory to the moment before it went down instead of just to the last backup.

Make sure you monitor disk space when roll-forward logging is on. For more information, see [“Backing Up and Removing Roll-Forward Logs” on page 383](#).

In this section:

- ♦ [“Issues to Be Aware of When Turning On Roll-Forward Logging” on page 381](#)
- ♦ [“Location of the Roll-Forward Logs” on page 382](#)
- ♦ [“Backing Up and Removing Roll-Forward Logs” on page 383](#)
- ♦ [“Cautionary Note: Removing eDirectory Also Removes the Roll-Forward Logs” on page 384](#)

You can turn on and configure roll-forward logging using either iManager or the eMBox Client. See [“Configuring Roll-Forward Logs with iManager” on page 389](#) or [“Configuring Roll-Forward Logs with the eMBox Client” on page 398](#).

Issues to Be Aware of When Turning On Roll-Forward Logging

If you decide to use continuous roll-forward logging, you must be aware of the following issues:

- ♦ **Turn on roll-forward logging before a backup is done** if you want to be able to use this feature for restoring the database.
- ♦ **For fault tolerance, make sure that the roll-forward logs are placed on a different storage device than eDirectory.** For security, you should also restrict user rights to the logs. For more information, see [“Location of the Roll-Forward Logs” on page 382](#).
- ♦ **Document the location of the roll-forward logs.** For more information, see [“Location of the Roll-Forward Logs” on page 382](#).
- ♦ **Monitor the available disk space where the logs are located.** For more information, see [“Backing Up and Removing Roll-Forward Logs” on page 383](#).
- ♦ **If the logs are turned off or lost, turn them back on, then do a new full backup** to ensure that you can make a full recovery. This is necessary in these cases:
 - ♦ After a restore. Roll-forward logging is turned off and the settings are reset to the default as part of the restore process.
 - ♦ If you lose the directory containing the roll-forward logs because of a storage device failure or other failure.
 - ♦ If roll-forward logs are unintentionally turned off.
- ♦ **If you turn on logging of stream files, the roll-forward logs use up disk space more quickly.** When logging of stream files (such as login scripts) is turned on, the whole stream file is copied into the roll-forward log every time there is a change. You can slow the growth of the log files by turning off roll-forward logging of stream files and, instead, back them up only when you do an incremental or full backup.
- ♦ **The slowest part of restoring the database is replaying the roll-forward logs.** Roll-forward logs grow larger based on how many changes are made to the tree structure and whether stream files (such as login scripts) are being logged.

If your database changes often, you might need to consider more frequent eDirectory backups so that fewer changes need to be replayed from roll-forward logs during a restore.

- ♦ **Don't change the name of a roll-forward log file.** If the filename is different than when the log was created, the log file can't be used in a restore.
- ♦ **Keep in mind that removing eDirectory also removes all the roll-forward logs.** If you want to be able to use the logs for restoring in the future, before removing eDirectory you must first copy the roll-forward logs to another location.
- ♦ **If a restore is necessary, make sure you re-create the roll-forward logs configuration on the server after the restore is complete** to make sure they are turned on and are placed in a fault-tolerant location. After turning on the roll-forward logs, you must also do a new full backup.

This step is necessary because during a restore, the configuration for roll-forward logging is set back to the default, which means that roll-forward logging is turned off and the location is set back to the default. The new full backup is necessary so that you are prepared for any failures that might occur before the next unattended full backup is scheduled to take place.

Location of the Roll-Forward Logs

If you turn on roll-forward logging, you should change the location of the roll-forward log directory to a different storage device than eDirectory.

Here are the important issues to consider when choosing the location:

- ♦ **Don't leave them in the default location—make sure you put them on a different storage device than eDirectory.** This way, if eDirectory is lost because of a storage device failure, you can still access the roll-forward logs to restore eDirectory.

For example, on NetWare the default location is `sys:_netware\nds.rfl\`. However, if you turn on roll-forward logging, you should not use the default location. The logs should not be placed on volume `sys:` because that is the same volume where the eDirectory database is located.

If you only have one storage device on your server, the roll-forward logs can't provide fault tolerance for eDirectory in case of a storage device failure. In this case, you probably should not use the roll-forward logs.

You can change the location of the roll-forward logs using Backup Configuration in iManager or `setconfig` in the eMBox Client. The roll-forward logs directory must be local to the server.

- ♦ **Document the location.** Document where the roll-forward logs are placed so that you can find them when you need to restore the database on a server. It's important to do this while the server is healthy, before any failures happen.

To find out the location when the server is healthy, you can look it up in iManager in Backup Configuration, or in the eMBox Client using the `backup getconfig` option. But if the server has a failure that affects eDirectory (such as hardware failure), you won't be able to look up the location of the roll-forward logs.

If the server has already had a failure and you are trying to restore it, keep in mind that any new installations of eDirectory will show the default location of the roll-forward logs. So, if you have just reinstalled eDirectory as the first step of a restore process, eDirectory will not show the correct location of the roll-forward logs on the server before it went down. You will need to refer to your documentation to find out where they are.

The settings for the roll-forward logs are also recorded in the `_ndsdb.ini` file, but that file is on the same disk partition/volume as eDirectory, so if you were to lose the storage device where eDirectory was located, you couldn't use the `_ndsdb.ini` file to look up the location.

- ◆ **Restrict rights to where the roll-forward logs are located.** This is a security issue. The information is not easily readable, but the logs could be decoded to reveal sensitive data.
- ◆ **Monitor the amount of free disk space to make sure there is enough.** See [“Backing Up and Removing Roll-Forward Logs” on page 383](#).
- ◆ **A good strategy is to set up a disk partition/volume solely for the roll-forward logs.** This way, disk space and security privileges can be easily monitored.
- ◆ **The last directory in the path is created by eDirectory.** It is based on the name of the current eDirectory database.

For example, if the location you specified was d:\Novell\NDS\DIBFiles and your eDirectory database was currently named NDS, the location of the roll-forward logs would be d:\Novell\NDS\DIBFiles\nds.rfl. If you renamed the database from NDS to ND1, the roll-forward log directory would be changed to d:\Novell\NDS\DIBFiles\nd1.rfl.

When you change the location, the new directory is created immediately, but a roll-forward log is not created there until a transaction takes place in the database.

- ◆ **When restoring, all the necessary roll-forward logs must be in the same directory.** For more information, see [“Preparing for a Restore” on page 384](#).

Backing Up and Removing Roll-Forward Logs

If left unchecked, roll-forward logs can fill up the disk partition/volume where they are placed. If roll-forward logs cannot be created because no more disk space is available, eDirectory stops responding on that server. We recommend that you periodically back up the log files and remove unused logs from the server to free up disk space.

To identify, back up, and remove roll-forward logs that are safe to remove:

- 1** Make a note of the name of the last unused roll-forward log.

You can find out the name of the last unused roll-forward log in the following ways:

- ◆ In iManager, click eDirectory Maintenance > Backup Configuration and read the filename displayed.
- ◆ In the eMBox Client, enter the getconfig backup command. See [“Configuring Roll-Forward Logs with the eMBox Client” on page 398](#) for instructions.

The last unused roll-forward log is the most recent roll-forward log that the database has completed and is no longer using to record transactions. It’s called the last unused roll-forward log because the database has finished writing to it and has begun a new log file, so it does not need to have this one open any more. (The current roll-forward log in which the database is recording transactions is in use and is still needed by the database.)

- 2** Do a file system backup of the roll-forward logs, to put them all safely on tape.
- 3** Remove the roll-forward logs that are older than the last unused roll-forward log.

WARNING: Keep in mind that you must be cautious when removing roll-forward logs from the server. Compare carefully with your tape backup to make sure you have a backup copy of everything you delete.

The last unused roll-forward log indicates which file the database has just completed and closed. It does not indicate whether it’s safe to remove that file from the server. You must make sure that you remove only files that you have a tape backup for.

If you need to retrieve any of the roll-forward logs from tape for use in a restore because you have placed some of them on tape backup, keep in mind the following issues:

- ♦ As with any roll-forward logs used for a restore, log files retrieved from file system backup tapes must be placed in the same folder as the other roll-forward logs, local to the server being restored.
- ♦ You must compare time stamps for any files that are duplicated on the tape and on the server. Use the latest one, the one on the server, if the time stamps are not the same. For example, the roll-forward log file that was in use by the database during the time of the file system backup will be incomplete on the tape; the latest and complete version of that file will be on the server.

Cautionary Note: Removing eDirectory Also Removes the Roll-Forward Logs

If you remove eDirectory from your server, the roll-forward log directory and all the logs in it are also removed. If you want to be able to use the logs for restoring the server in the future, before removing eDirectory you must first copy the roll-forward logs to another location.

Preparing for a Restore

The most important part of restoring the eDirectory database is making sure it is complete. Before restoring an eDirectory database to a server, ensure the prerequisites have been met as described in [“Prerequisites for Restoring” on page 384](#). If you are not sure how to gather the right backup files, see [“Locating the Right Backup Files for a Restore” on page 385](#).

Prerequisites for Restoring

- All servers that share a replica with the server to be restored are up and communicating. This allows the restore verification process to check with servers that participate in the same replica ring.
- You have gathered all the backup files you need:
 - ♦ The full backup and subsequent incremental backup files are copied to one directory on the server to be restored.
 - ♦ All roll-forward logs since the last backup are in one directory on the server to be restored.

If this server participates in a replica ring, you must make sure all the roll-forward logs created since the last backup are in one directory on the server, with the same filenames they had when they were created.

See [“Locating the Right Backup Files for a Restore” on page 385](#).

NOTE: If you do not have backup files for the server, use XBrowse to query eDirectory to help you recover server information. You must do this before you remove the Server object or any associated objects from the tree. XBrowse and additional information is available from the [Novell Support Web site, Solution 2960653 \(http://support.novell.com/servlet/tidfinder/2960653\)](#).

- You have installed eDirectory, in a new temporary tree.

You bring up the server in a new tree at first because you will create the server with the same name it had before the failure, and you don't want to cause confusion in the original tree by putting the newly installed server in the tree before the restore has re-created the server's complete identity. Completing the restore process for the database will put the server back into its original tree.

- (Conditional) If you are using roll-forward logging on this server, plan to re-create your configuration for roll-forward logging after the restore, to make sure it is turned on and the logs are being saved in a fault-tolerant location. After turning on the roll-forward logs, you must also do a new full backup.

The restore process turns off roll-forward logging and resets the configuration for roll-forward logging back to the default.

The new full backup is necessary so that you are prepared for any failures that might occur before the next unattended full backup is scheduled to take place.

- ❑ (Conditional) If any applications or objects need to find this server by its IP address, use the same IP address for the restored server.
- ❑ (NetWare only) Make sure the name of the server you are restoring to is the same as the name of the failed server. If the names are not the same, you might encounter errors, such as Volume objects not being correct after the restore.

To change the name of the NetWare server you are restoring to, change the name in the `autoexec.ncf` file and restart the server.

- ❑ (NetWare only) Be aware of the issues involved with preserving file system rights when restoring file system data and eDirectory. You should restore eDirectory before restoring the file system data. You also might need to take additional steps, as explained in [“Preserving Rights When Restoring File System Data on NetWare” on page 379](#).

During the restore process, the eDirectory Backup eMTool first restores the full backup. After this is complete, the Backup eMTool prompts you to enter the filenames of the incremental backup files. It provides you with the ID of the next file. After all incremental files are restored, the Backup eMTool moves on to the roll-forward logs. (See also [“Overview of How the Backup eMTool Does a Restore” on page 371](#).)

After you have gathered all the files, perform the restore using either iManager or the eMBox Client. See [“Restoring from Backup Files with the eMBox Client” on page 400](#) or [“Restoring from Backup Files with iManager” on page 390](#).

Locating the Right Backup Files for a Restore

- 1 From your file system backup tape, copy the eDirectory full backup files to one directory on the server.

You can check the Backup eMTool log file if you want to confirm the ID of the last full backup.

- 2 From your file system backup tape, also copy each of the subsequent incremental backup files to the a directory on the server.

To confirm that you have the right incremental backup files, look in the header of the full backup file. It contains the ID of the next incremental backup file, shown in the `next_inc_file_ID` attribute. The `next_inc_file_ID` is the same as the ID noted in the header of the incremental backup file in the `incremental_file_number` attribute. (For a description of the header, see [“Format of the Backup File Header” on page 372](#).)

WARNING: When opening a backup file, just view the header—make sure you don't try to save or modify the file, or it might become truncated. Most applications can't save the binary data correctly.

Each incremental backup file will also contain the ID for the next incremental backup file.

You can also look for the incremental backup ID in the Backup eMTool log file.

The IDs are important because your backup files might have had the same filenames when they were created (for example, if you used the same batch file for unattended incremental backups so the backup filename specified was always the same), and you might have to change the filenames so you can place all the backups in the same directory. The ID in the header lets you find the correct files even if you have changed the filenames.

- 3 (Conditional) If you are using roll-forward logging on this server, make sure the roll-forward logs created since the last backup are in one directory on the server, with the same filenames they had when they were created.

If this server participates in a replica ring, you must restore using all the roll-forward logs. If you don't include all the roll-forward logs, the restore verification process will not be successful because the transitive vectors will not match when compared to the other replicas in the ring. By default the restored eDirectory database will not open after the restore if it is inconsistent with the other replicas.

Identify the first roll-forward log you need by opening the last backup file in a text editor and reading the `current_log` attribute in the header. You will need to collect this one and all the subsequent roll-forward logs.

WARNING: When opening a backup file, just view the header—make sure you don't try to save or modify the file, or it might become truncated. Most applications can't save the binary data correctly.

The roll-forward logs you need might not all be in the same location at the time you want to use them for a restore, so you need to make sure you have collected a complete set and placed them all in the same directory. The roll-forward logs might be in multiple locations for the following reasons:

- ◆ You have changed the location of the roll-forward logs directory since the last full or incremental backup.
- ◆ You have backed them up to tape using file system backup and then have removed them from the server, to save disk space.

If you need to retrieve any of the roll-forward logs from tape backup, make sure you have the most current set. You must compare time stamps for any files that are duplicated on the tape and on the server. The roll-forward log file that was in use by the database during the time of the file system backup will be incomplete on the tape; the latest and complete version of that file will be on the server.

- ◆ You have changed the name of the eDirectory database since the last backup (such as from NDS to ND1). This changes the last directory name in the path to the roll-forward logs.

For example, if the location you specified was `d:\novell\nds\dibfiles\`, and the name of your eDirectory database was NDS, the location of the roll-forward logs would be `d:\novell\nds\dibfiles\nds.rfl\`. If you renamed the database from NDS to ND1, the roll-forward log directory would change to `d:\novell\nds\dibfiles\nd1.rfl\`.

IMPORTANT: You must ensure that you provide all the necessary roll-forward logs. The Backup eMTool cannot tell whether your set of roll-forward logs is complete. It will open and use the roll-forward logs in order. When it cannot find the next roll-forward log in the directory you specified, it ends the restore process. If you have not provided all the necessary roll-forward logs, the restore will be incomplete.

Using Novell iManager for Backup and Restore

The Backup, Backup Configuration, and Restore tasks in Novell iManager give you access to most of the features of the eDirectory Backup eMTool, and iManager lets you perform tasks on your servers in a browser even if you are outside the firewall. For more information about Novell iManager, see the *Novell iManager 2.0.x Administration Guide* (<http://www.novell.com/documentation/lg/imanager20/index.html>).

The tasks that are not available in iManager are cold backup (a full backup with the database closed), unattended backup, and advanced restore options. These tasks must be done using the eMBox Client, as described in “Using the eMBox Client for Backup and Restore” on page 393.

Before performing backup and restore tasks, review [“Checklist for Backing Up eDirectory” on page 366](#) for an overview of the issues involved in planning an effective eDirectory backup strategy.

In this section:

- ◆ [“Backing Up Manually with iManager” on page 387](#)
- ◆ [“Configuring Roll-Forward Logs with iManager” on page 389](#)
- ◆ [“Restoring from Backup Files with iManager” on page 390](#)

Backing Up Manually with iManager

Use Backup in iManager in a browser to back up data from an eDirectory database to one or more files on the server where the backup is being performed. You can do a full or incremental backup.

The backup files contain information necessary to restore eDirectory to the state it was in at the time of the backup. The results of the backup process are written to the log file you specify.

Backups performed using iManager are hot continuous backups, meaning that the eDirectory database is open and accessible during the process, and you still get a complete backup that is a snapshot of the moment when the backup began.

Keep in mind that to do a cold backup (a backup with the database closed) or an unattended backup you must use the eMBox Client. See [“Backing Up Manually with the eMBox Client” on page 393](#) and [“Doing Unattended Backups, Using a Batch File with the eMBox Client” on page 396](#).

Before performing backup and restore tasks, review [“Checklist for Backing Up eDirectory” on page 366](#) for an overview of the issues involved in planning an effective eDirectory backup strategy.

Prerequisites

- Decide which additional files you want to back up along with eDirectory and create an include file if necessary.

If you want to include other files, such as the autoexec.ncf file, you must put the paths and filenames in an include file. Separate the paths and filenames with a semicolon and don't include hard returns or spaces. (For example, sys:\system\autoexec.ncf;sys:\etc\hosts;)

- Plan to do a file system backup shortly after doing the eDirectory backup, if you need to place the eDirectory backup files safely on tape. (The Backup eMTool only places them on the server.)

TIP: To make it easier to move the backup files to another storage device, you can specify the maximum size of eDirectory backup files. You can also use a third-party file compression tool on the files after they are created. They compress approximately 80%.

- If you are planning to use roll-forward logs for this server, make sure they are turned on before a backup is made.

You must turn on roll-forward logging for servers that participate in a replica ring. If you don't, when you try to restore from your backup files you will get errors and the database will not open.

For more information on roll-forward logs, see [“Using Roll-Forward Logs” on page 380](#). For how to turn them on, see [“Configuring Roll-Forward Logs with iManager” on page 389](#).


- For multiple-server trees, you should upgrade all the servers that share replicas with this server to eDirectory 8.5 or later.

For more information, see “[Restore Verification Is Backward Compatible Only with eDirectory 8.5 or Later](#)” on page 378.

Procedure

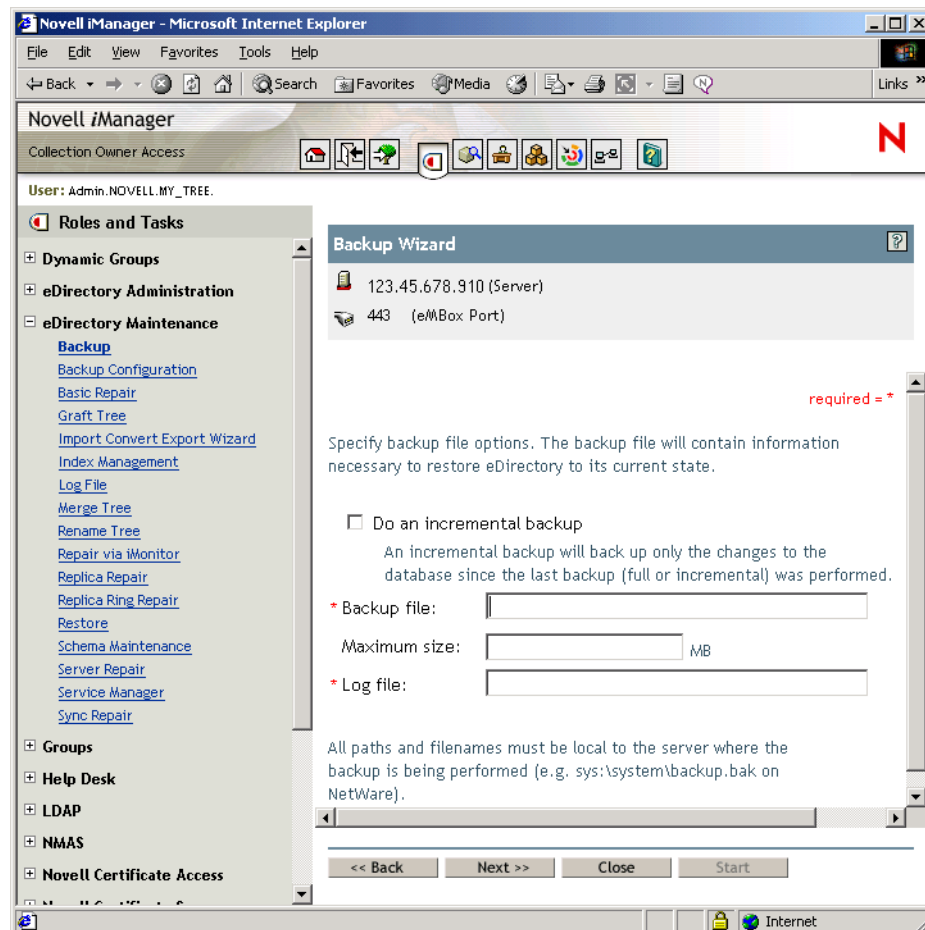
To back up the eDirectory database on a server, using iManager:

TIP: A description of the options available in iManager is provided in the online help.

- 1 Click the Roles and Tasks button .
- 2 Click eDirectory Maintenance > Backup.
- 3 Specify the server that will perform the backup, then click Next.
- 4 Specify a username, password, and context for the server where you want to perform the backup, then click Next.
- 5 Specify backup file options, then click Next.

To back up only the changes made to the database since the last backup was performed, click Do an Incremental Backup.

The following is an example of the screen.



- 6 Specify additional files to back up.
If no additional files are specified, only the eDirectory database is backed up.

- 7 Follow the online instructions to complete the backup.
- 8 Make sure you do a file system backup shortly after the eDirectory backup is created, to put the eDirectory backup files safely on tape. (The Backup eMTool only places them on the server.)

Configuring Roll-Forward Logs with iManager

Use Backup Configuration in a browser to change the settings for roll-forward logs. You can do the following tasks:


- ◆ Turn roll-forward logging on or off

You must turn on roll-forward logging for servers that participate in a replica ring. If you don't, when you try to restore from your backup files you will get errors and the database will not open.

- ◆ Change the roll-forward logs directory
- ◆ Set the minimum and maximum roll-forward log size
- ◆ Determine the current and last unused roll-forward log
- ◆ Turn stream file logging on or off for the roll-forward logs

For more information about roll-forward logs, see [“Using Roll-Forward Logs” on page 380](#).

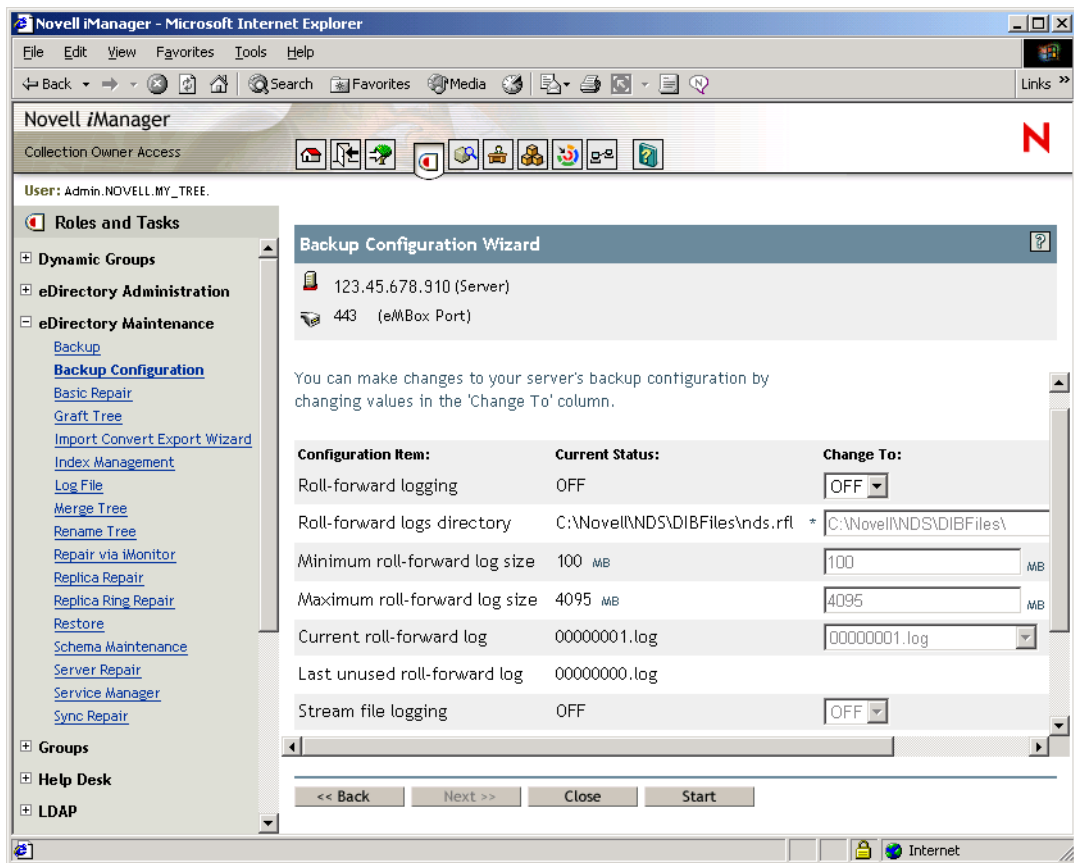
TIP: A description of the options available in iManager is provided in the online help.

- 1 Click the Roles and Tasks button .
- 2 Click eDirectory Maintenance > Backup Configuration.
- 3 Specify the server that will change configuration, then click Next.
- 4 Specify a username, password, and context for the server where you want to change configuration, then click Next.
- 5 Make the changes you want to the server's backup configuration.

WARNING: If you turn on roll-forward logging, don't use the default location. For fault tolerance, put the directory on a different disk partition/volume and storage device than eDirectory. The roll-forward logs directory must be on the server where the backup configuration is being changed.

IMPORTANT: If you turn on roll-forward logging, you must monitor disk space on the volume where you place the roll-forward logs. If left unchecked, the log file directory will grow until it fills up the disk partition/volume. If roll-forward logs cannot be created because no more disk space is available, eDirectory stops responding on that server. We recommend you periodically back up and remove unused roll-forward logs from your server. See [“Backing Up and Removing Roll-Forward Logs” on page 383](#).

The following is an example of the screen.



- 6 Follow the online instructions to complete the operation.

Restoring from Backup Files with iManager

Use Restore in a browser to restore an eDirectory database from data stored in backup files. The results of the restore process are written to the log file you specify.

For a description of the restore process, see [“Overview of How the Backup eMTool Does a Restore” on page 371](#).

Keep in mind that for advanced restore options you must use the eMBox Client, as described in [“Using the eMBox Client for Backup and Restore” on page 393](#).

Prerequisites

- Gather all the backup files you need for a restore and place them in a directory on the server you are restoring to.

See [“Preparing for a Restore” on page 384](#) and [“Locating the Right Backup Files for a Restore” on page 385](#).

- Make sure eDirectory is already installed on the server you are restoring to and is up and running.


For example, if the restore is necessary because of a failed storage device, you need to do a new installation of eDirectory on the new storage device. If you are restoring a failed server onto a brand new machine, or simply moving a server from one machine to another, you need to install both the operating system and eDirectory on the new machine.

- ❑ Review the description of the restore process in [“Overview of How the Backup eMTool Does a Restore”](#) on page 371.
- ❑ (NetWare only) Be aware of the issues involved with preserving file system rights when restoring file system data and eDirectory. You should restore eDirectory before restoring the file system data. You also might need to take additional steps, as explained in [“Preserving Rights When Restoring File System Data on NetWare”](#) on page 379.

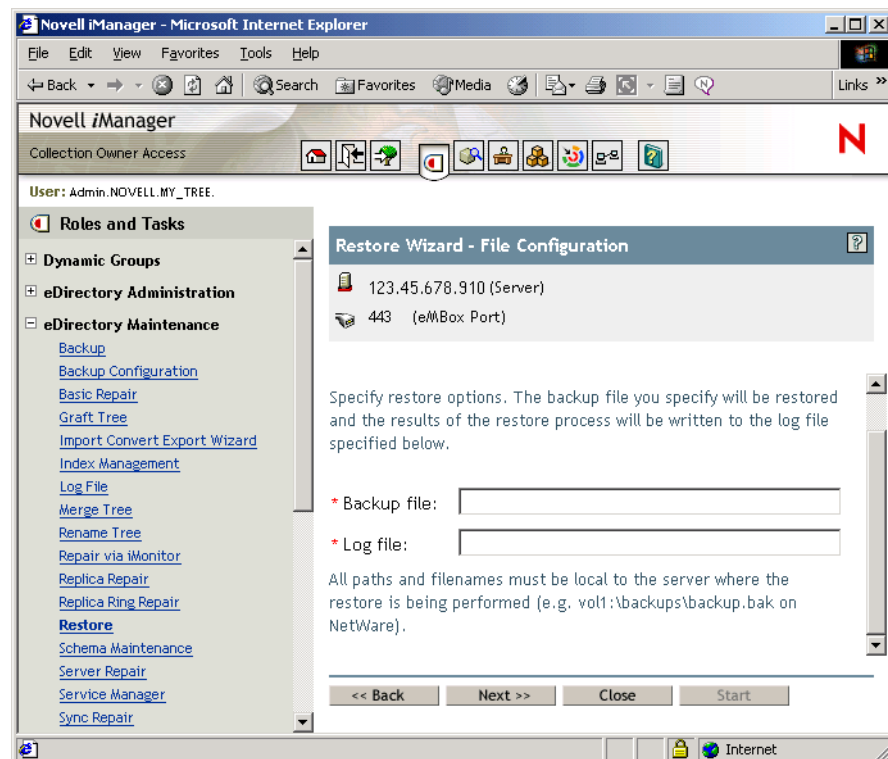
Procedure

TIP: A description of the options available in iManager is provided in the online help.

To restore the eDirectory database on a server, using iManager:

- 1 Make sure you have gathered the backup files you need, as described in [“Preparing for a Restore”](#) on page 384.
- 2 Click the Roles and Tasks button .
- 3 Click eDirectory Maintenance > Restore.
- 4 Specify the server that will perform the restore, then click Next.
- 5 Specify a username, password, and context for the server where you want to perform the restore, then click Next.
- 6 Specify the name of the backup and log files you want to use, then click Next.

The following is an example of the screen.



- 7 Specify additional restore options, then click Next.

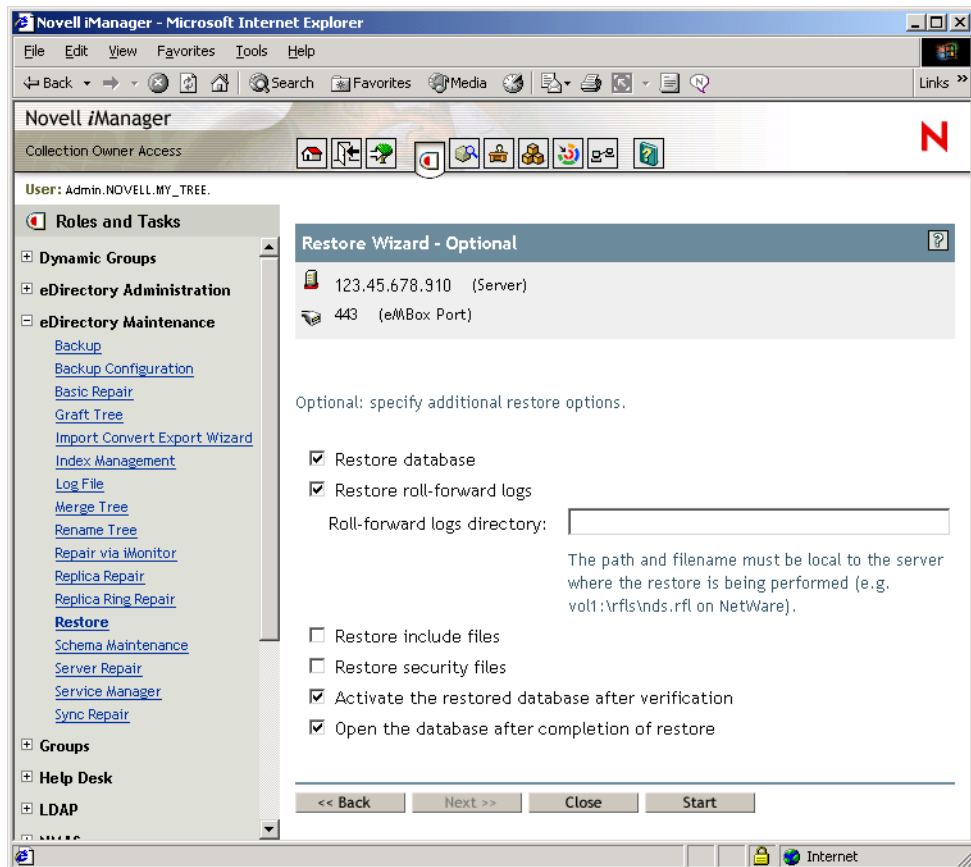
In most cases you should at least check the check boxes for

- ♦ Restore database

- ◆ Activate the restored database after verification
- ◆ Open the database after completion of restore

If you are restoring roll-forward logs, make sure you include the full path to the logs, including the directory that is automatically created by eDirectory, usually named \nds.rfl. (For more information about this directory, see [“Location of the Roll-Forward Logs” on page 382.](#))

The following is an example of the screen.



- 8 Follow the online instructions to complete the restore.

If the restore verification fails, see [“Recovering the Database If Restore Verification Fails” on page 411.](#)

NOTE: If the server you are restoring shares a replica with a server running an earlier version than eDirectory 8.5, the restore log will show a -666 error (incompatible DS version) for that replica. For more information on this situation and what you might be able to do, see [“Restore Verification Is Backward Compatible Only with eDirectory 8.5 or Later” on page 378.](#)

- 9 Make sure the server is responding as usual.
- 10 (Conditional) If you are using roll-forward logging on this server, you must re-create your configuration for roll-forward logging to make sure it is turned on and the logs are being saved in a fault-tolerant location. After turning on the roll-forward logs, you must also do a new full backup.

This step is necessary because during a restore, the configuration for roll-forward logging is set back to the default, which means that roll-forward logging is turned off and the location is

set back to the default. The new full backup is necessary so that you are prepared for any failures that might occur before the next unattended full backup is scheduled to take place.

For more information about roll-forward logs and their location, see [“Using Roll-Forward Logs” on page 380](#).

Your restore should now be complete. If you use roll-forward logging, you have prepared for any failures in the future by turning on roll-forward logging again after the restore and creating a new full backup as a baseline.

Using the eMBox Client for Backup and Restore

The eMBox Client is a command line Java client that gives you access to eMBox tools such as the eDirectory Backup eMTool. You can back up, restore, and configure roll-forward logging for multiple servers from a single machine if you have access behind the firewall.

Because the eMBox Client can be run in batch mode, you can use it to do unattended backups using the eDirectory Backup eMTool.

The eMBoxClient.jar file is installed on your server as part of eDirectory. You can also copy the file and run it on any machine with Sun JVM 1.3.1. For more information, see [“The eDirectory Management Toolbox” on page 465](#) and [“Running the eMBox Client on a Workstation” on page 467](#).

Before performing backup and restore tasks, review [“Checklist for Backing Up eDirectory” on page 366](#) for an overview of the issues involved in planning an effective eDirectory backup strategy.

In this section:

- ◆ [“Backing Up Manually with the eMBox Client” on page 393](#)
- ◆ [“Doing Unattended Backups, Using a Batch File with the eMBox Client” on page 396](#)
- ◆ [“Configuring Roll-Forward Logs with the eMBox Client” on page 398](#)
- ◆ [“Restoring from Backup Files with the eMBox Client” on page 400](#)
- ◆ [“Backup and Restore Command Line Options” on page 403](#)

Backing Up Manually with the eMBox Client

Use the eMBox Client to back up data from an eDirectory database to a file you specify on the server where the backup is being performed. This backup file or set of files contains information necessary to restore eDirectory to the state it was in at the time of the backup. The results of the backup process are written to the log file you specify.

Before performing backup and restore tasks, review [“Checklist for Backing Up eDirectory” on page 366](#) for an overview of the issues involved in planning an effective eDirectory backup strategy.

Using the eMBox Client, you can do tasks such as the following:

- ◆ Do a full or incremental backup while the database is open (hot continuous backup)

Hot continuous backup means that the eDirectory database is open and accessible during the process, and you still get a complete backup that is a snapshot of the moment when the backup began.

- ◆ Do a cold backup (the database is closed and a full backup is created)

This option is helpful when upgrading hardware or moving a server to a new machine with the same operating system (as described in [“Upgrading Hardware or Replacing a Server”](#) on page 447).
- ◆ Set the database to stay closed and locked after a backup
- ◆ Set the maximum backup file size

To do these tasks unattended, see [“Doing Unattended Backups, Using a Batch File with the eMBox Client”](#) on page 396.

Prerequisites

- Make sure the eMBoxClient.jar file is on the machine you want to initiate the backup from.

The file is installed on your server as part of eDirectory. You can copy it from there and run it on any machine with Sun JVM 1.1.3. You can run backups for multiple servers from a single machine if you have access behind the firewall. For more information, see [“Using the eMBox Command Line Client”](#) on page 465.
- If you are planning to use roll-forward logs for this server, make sure they are turned on before a backup is made.

You must turn on roll-forward logging for servers that participate in a replica ring. If you don't, when you try to restore from your backup files you will get errors and the database will not open.

For more information on roll-forward logs, see [“Using Roll-Forward Logs”](#) on page 380. For how to turn them on, see [“Configuring Roll-Forward Logs with the eMBox Client”](#) on page 398.
- Decide which additional files you want to back up along with eDirectory, and create an include file if necessary.

If you want to include other files, such as the autoexec.ncf file, you must put the paths and filenames in an include file. Separate the paths and filenames with a semicolon and don't include hard returns or spaces. (For example, sys:\system\autoexec.ncf;sys:\etc\hosts;)
- Plan to do a file system backup shortly after doing the eDirectory backup, to place the eDirectory backup files safely on tape. (The Backup eMTool only places them on the server.)

TIP: To make it easier to move the backup files to another storage device, you can specify the maximum size of eDirectory backup files as part of the backup command (use -s and a number for size in bytes). You can also use a third-party file compression tool on the files after they are created. They compress approximately 80%.
- Review the description of the command line options in [“Backup and Restore Command Line Options”](#) on page 403.
- For multiple-server trees, you should upgrade all the servers that share replicas with this server to eDirectory 8.5 or later.

For more information, see [“Restore Verification Is Backward Compatible Only with eDirectory 8.5 or Later”](#) on page 378.

Procedure

To back up the eDirectory database on a server using the eMBox Client:

- 1 Run the eMBox Client in interactive mode.
 - ◆ NetWare and UNIX: At the command line, enter `edirutil -i`.

- ♦ Windows: Run
`drive\novell\nds\edirutil.exe -i`

The edirutil file gives you a shortcut to running the eMBox Client. It points to the Java executable and the default location where the eMBox Client is installed with eDirectory, and for NetWare, it includes the necessary `-ns` option. (You can also enter the information manually, as described in [“Setting Up the Path and Classpath for eMBox Client” on page 467.](#))

When the eMBox Client opens, the eMBox Client prompt appears: eMBox Client>

- 2** Log in to the server you want to back up by entering

```
login -s server_name_or_IP_address -p port_number  
-u username.context -w password
```

For example, on Windows enter

```
login -s 151.155.111.1 -p 8009 -u admin.mycompany -w mypassword
```

If you get an error saying that a secure connection cannot be established, make sure your machine has the JSSE files listed in [“Establishing a Secure Connection with the eMBox Client” on page 473.](#)

For help finding out which port number to use, see [“Finding Out eDirectory Port Numbers” on page 473.](#)

The eMBox Client indicates whether the login is successful.

- 3** Enter the backup command at the eMBox Client prompt, following this general pattern:

```
backup -b -f backup_filename_and_path -l backup_log_filename_and_path -u  
include_file_filename_and_path -t -w
```

A space must be between each switch. The order of the switches is not important.

For example, on Windows enter

```
backup -b -f c:\backups\8_20_2001.bak -l c:\backups\backup.log -u  
c:\backups\myincludefile.txt -t -w
```

This example command would result in a full backup (`-b`) with the backup file placed at `c:\backups\8_20_2001.bak` and the log file for the process placed at `c:\backups\backup.log`. This command specifies that other files should be backed up along with the database:

- ♦ The files listed in an include file (`-u c:\backups\myincludefile.txt`) that was created beforehand by the administrator.
- ♦ Stream files (`-t`)

This example command specifies that the backup file should be overwritten (`-w`), so if a file of the same name existed, the Backup eMTool would replace it.

The eMBox Client indicates whether the backup is successful.

- 4** Log out from the server by entering the following command:

```
logout
```

- 5** Exit the eMBox Client by entering the following command:

```
exit
```

- 6** Make sure you do a file system backup shortly after the eDirectory backup is created, to put the eDirectory backup files safely on tape. (The Backup eMTool only places them on the server.)

Doing Unattended Backups, Using a Batch File with the eMBox Client

Use a batch file to do unattended backups of eDirectory through the eMBox Client. For example, you might want to do a full backup of eDirectory on your servers weekly and an incremental backup nightly.

You can run the eMBox Client in batch mode using a system batch file, an eMBox Client internal batch file, or a combination of both. For more information, see [“Running the eMBox Command Line Client in Batch Mode” on page 470](#).

This procedure describes using a system batch file.

Prerequisites

- Consult the documentation for your operating system or third-party scheduling software for instructions on how to run batch files unattended.

NOTE: On NetWare, you can use third-party scheduling software, or `cron.nlm` (<http://support.novell.com/servlet/tidfinder/2939440>), available from the Novell Support Web site.

- Make sure the eMBoxClient.jar file is on the machine you want to initiate the backup from.

The file is installed on your server as part of eDirectory. You can copy it from there and run it on any machine with Sun JVM 1.3.1. You can run backups for multiple servers from a single machine if you have access behind the firewall. For more information, see [“Using the eMBox Command Line Client” on page 465](#).

- If you are planning to use roll-forward logs for this server, make sure they are turned on before a backup is made.

You must turn on roll-forward logging for servers that participate in a replica ring. If you don't, when you try to restore from your backup files you will get errors and the database will not open.

For more information on roll-forward logs, see [“Using Roll-Forward Logs” on page 380](#). For how to turn them on, see [“Configuring Roll-Forward Logs with the eMBox Client” on page 398](#).

- Decide which additional files you want to back up along with eDirectory and create an include file if necessary.

If you want to include other files, such as the `autoexec.ncf` file, you must put the paths and filenames in an include file. Separate the paths and filenames with a semicolon and don't include hard returns or spaces. (For example, `sys:\system\autoexec.ncf;sys:\etc\hosts;`)

- Schedule file system backups shortly after eDirectory backups, to place the eDirectory backup files safely on tape. (The Backup eMTool only places them on the server.)

TIP: To make it easier to move the backup files to another storage device, you can specify the maximum size of eDirectory backup files. You can also use a third-party file compression tool on the files after they are created. They compress approximately 80%.

- Review the description of the command line options in [“Backup and Restore Command Line Options” on page 403](#).

Procedure

- 1 Create a system batch file to back up the servers, following these general patterns, with one line per server.

On Windows and UNIX, this is the general pattern:

```
java -cp path/eMBoxClient.jar embox -s server_name -p port_number -u
username.context -w password -t backup.backup -b -f
backup_filename_and_path -l backup_log_filename_and_path -u
include_file_filename_and_path -t -w
```

On NetWare, you would follow the same general pattern, but with the addition of `-nsac`, which should not be used on the other platforms:

```
java -nsac -cp path/eMBoxClient.jar embox -s server_name -p port_number -
u username.context -w password -t backup.backup -b -f
backup_filename_and_path -l backup_log_filename_and_path -u
include_file_filename_and_path -t -w
```

For examples and more explanation, see [“Examples of System Batch Files for Unattended Backups” on page 397](#).

For nightly incremental backups, you could use the same file you use for full backups, but change the `-b` switch to `-i` to do an incremental backup instead of a full backup. It’s also probably a good idea to use a different backup filename for incremental backups than for the full backup.

For help finding out which port number to use, see [“Finding Out eDirectory Port Numbers” on page 473](#). If you want to use a secure connection, see [“Establishing a Secure Connection with the eMBox Client” on page 473](#). For information on using an eMBox Client internal batch file as well, see [“Running the eMBox Command Line Client in Batch Mode” on page 470](#).

- 2** Run the batch files unattended, according to the instructions in your operating system or third-party documentation.
- 3** Make sure you schedule file system backups shortly after eDirectory backups, to place the eDirectory backup files safely on tape.

The Backup eMTool only places them on the server.

- 4** Periodically check the results recorded in the log file you specified, to make sure the unattended backups are successful.

Examples of System Batch Files for Unattended Backups

Below are the following two examples:

- ♦ [“Example Batch File for NetWare” on page 397](#)
- ♦ [“Example Batch File for Windows” on page 398](#)

Example Batch File for NetWare

```
java -nsac -cp sys:\system\embox\eMBoxClient.jar embox -s 10.10.1.200 -p 8008
-u admin.mycontainer -w mypassword -n -t backup.backup -b -f
sys:\system\backup\backup.bak -l sys:\system\backup\backup.log -u
sys:\system\backup\includefile.txt -t -w
```

In this example batch file, the following options are shown.

- ♦ On NetWare only, include `-nsac` after the java command. (Don’t use `-nsac` on any other platform.)

WARNING: On a NetWare server only, to avoid an abend you must include `-ns`.

The `-ns` option opens a new screen.

The `ac` option automatically closes the screen when the batch file task is complete. If you don't include `ac` in NetWare batch files, each time the backup batch file is run unattended a screen will be left open on the server.

- ◆ A full backup is specified (`-b`).
- ◆ An include file is specified (`-u`). This is optional. You can use an include file if you want to back up other files of your choice. The include file must be created beforehand.
- ◆ Stream files (`-t`) are also backed up.
- ◆ The option to overwrite a backup file of the same name is specified (`-w`).

IMPORTANT: If a backup file of the same name exists (this is likely if you use the same batch file regularly), it's important to use the `-w` option to overwrite the existing backup file to make sure your backup is successful.

In batch mode, if `-w` is not specified and a file of the same name exists, the default behavior is to not overwrite the file, so a backup is not created. (In interactive mode, if `-w` is not specified, the eMBox Client will ask you whether you want to overwrite the file.)

If you are making a file system backup shortly after each full or incremental backup of eDirectory, your previous backup files should have been copied from the server to file system backup tapes, so it should be safe to use this option to overwrite the existing backup file.

- ◆ A nonsecure port is used in this example (`-p 8008`), so a nonsecure connection is specified (`-n`).

Example Batch File for Windows

```
java -cp c:\novell\nds\embox\emBoxClient.jar embox -s myserver -p 8008 -u
admin.myorg -w mypassword -n -t backup.backup -b -f c:\backup\backup.bak -u
c:\backup\includes\includefile.txt -l c:\backup\backup.log -t -w
```

In this example batch file, the following options are shown.

- ◆ A full backup is specified (`-b`).
- ◆ An include file is specified (`-u`). This is optional. You can use an include file if you want to back up other files of your choice. The include file must be created beforehand.
- ◆ Stream files (`-t`) are also backed up.
- ◆ The option to overwrite a backup file of the same name is specified (`-w`).

IMPORTANT: If a backup file of the same name exists (this is likely if you use the same batch file regularly), it's important to use the `-w` option to overwrite the existing backup file to make sure your backup is successful.

In batch mode, if `-w` is not specified and a file of the same name exists, the default behavior is to not overwrite the file, so a backup will not be created. (In interactive mode, if `-w` is not specified, the eMBox Client will ask you whether you want to overwrite the file.)

If you are making a file system backup shortly after each full or incremental backup of eDirectory, your previous backup files should have been copied from the server to file system backup tapes, so it should be safe to use this option to overwrite the existing backup file.

- ◆ A nonsecure port is used in this example (`-p 8008`), so a nonsecure connection is specified (`-n`).

NOTE: The `-ns` or `ac` options shown in NetWare batch file examples are to be used only on the NetWare platform. Don't use them for Windows or UNIX.

Configuring Roll-Forward Logs with the eMBox Client

Use the eMBox Client to change the settings for roll-forward logs. You can do the following tasks:

- ◆ Find out the current settings
- ◆ Turn roll-forward logging on or off

You must turn on roll-forward logging for servers that participate in a replica ring. If you don't, when you try to restore from your backup files you will get errors and the database will not open.
- ◆ Change the roll-forward logs directory
- ◆ Set the minimum and maximum roll-forward log size
- ◆ Find out the current and last unused roll-forward log
- ◆ Turn stream file logging on or off for the roll-forward logs

For information about roll-forward logging, see [“Using Roll-Forward Logs” on page 380](#).

Prerequisites

- Make sure the eMBoxClient.jar file is on the machine you want to initiate the configuration changes from.

The file is installed on your server as part of eDirectory. You can copy it from there and run it on any machine with Sun JVM 1.3.1. You can run backups for multiple servers from a single machine if you have access behind the firewall. For more information, see [“Using the eMBox Command Line Client” on page 465](#).

- Review the description of the command line options in [“Backup and Restore Command Line Options” on page 403](#).

Procedure

- 1** Run the eMBox Client in interactive mode:

- ◆ NetWare and UNIX: At the command line, enter `edirutil -i`.
- ◆ Windows: Run


```
drive\novell\nds\edirutil.exe -i.
```

The edirutil file gives you a shortcut to running the eMBox Client. It points to the Java executable and the default location where the eMBox Client is installed with eDirectory, and for NetWare, it includes the necessary -ns option. (You can also enter the options manually, as described in [“Running the eMBox Client on a Workstation” on page 467](#).)

When the eMBox Client opens, the eMBox Client prompt appears: eMBox Client>

- 2** Log in to the server you want to configure roll-forward logging on by entering

```
login -s server_name_or_IP_address -p port_number
-u username.context -w password
```

For example, on Windows enter

```
login -s 151.155.111.1 -p 8009 -u admin.mycompany -w mypassword
```

If you get an error saying that a secure connection cannot be established, make sure your machine has the JSSE files listed in [“Establishing a Secure Connection with the eMBox Client” on page 473](#).

For help finding out which port number to use, see [“Finding Out eDirectory Port Numbers” on page 473](#).

The eMBox Client indicates whether the login is successful.

- 3** (Optional) Find out the current settings by entering

```
getconfig
```

No switches are necessary.

The following is an example of the information you receive:

```
Roll forward log status OFF
Stream file logging status OFF
Current roll forward log directory voll:/rfl/nds.rfl
Minimum roll forward log size (bytes) 104857600
Maximum roll forward log size (bytes) 4294705152
Last roll forward log not used 00000000.log
Current roll forward log 00000001.log
*** END ***
```

- 4** Change the settings using the setconfig command, following this general pattern:

```
setconfig [-L|-l] [-T|-t] -r path_to_roll-forward_logs -n minimum_file_size -m
maximum_file_size
```

A space must be between each switch. The order of the switches is not important.

For example, on NetWare enter

```
setconfig -L -r rflvolume:\logs
```

This example turns on roll-forward logging (-L switch) and specifies that the roll-forward logs are placed in rflvolume:\logs. (Ideally, you would have a separate disk partition/volume dedicated to roll-forward logs to make it easier to monitor disk space and rights.) This example does not include the option to turn on logging of stream files.

WARNING: If you turn on roll-forward logging, don't use the default location. For fault tolerance, put the directory on a different disk partition/volume and storage device than eDirectory. The roll-forward logs directory must be on the server where the backup configuration is being changed.

IMPORTANT: If you turn on roll-forward logging, you must monitor disk space on the volume where you place the roll-forward logs. If left unchecked, the log file directory will grow until it fills up the disk partition/volume. If roll-forward logs cannot be created because no more disk space is available, eDirectory stops responding on that server. We recommend you periodically back up and remove unused roll-forward logs from your server. See ["Backing Up and Removing Roll-Forward Logs" on page 383](#).

- 5** Log out from the server by entering the following command:

```
logout
```

- 6** Exit the eMBox Client by entering the following command:

```
exit
```

Restoring from Backup Files with the eMBox Client

Use the eMBox Client to restore an eDirectory database from data stored in backup files you created manually or with a batch file. The results of the restore process are written to the log file you specify.

The eMBox Client also lets you use advanced restore options not available in iManager. They are described in ["Backup and Restore Command Line Options" on page 403](#), under **restore** and **restadv**.

Prerequisites

- Make sure the eMBoxClient.jar file is on the machine you want to initiate the restore from.

The file is installed on your server as part of eDirectory. You can copy it from there and run it on any machine with Sun JVM 1.3.1. You can do restores for multiple servers from a single machine if you have access behind the firewall. For more information, see [“Using the eMBox Command Line Client” on page 465](#).

- ❑ Gather all the backup files you need for a restore and place them in a directory on the server you are restoring to.

See [“Preparing for a Restore” on page 384](#) and [“Locating the Right Backup Files for a Restore” on page 385](#).

- ❑ Make sure eDirectory is installed and running on the server you are restoring to.

For example, if the restore is necessary because of a failed storage device, you need to do a new installation of eDirectory on the new storage device. If you are restoring a failed server onto a brand new machine, or simply moving a server from one machine to another, you need to install both the operating system and eDirectory on the new machine.

- ❑ Review the description of the command line options in [“Backup and Restore Command Line Options” on page 403](#).
- ❑ Review the description of the restore process in [“Overview of How the Backup eMTool Does a Restore” on page 371](#).
- ❑ (NetWare only) Be aware of the issues involved with preserving file system rights when restoring file system data and eDirectory. You should restore eDirectory before restoring the file system data. You also might need to take additional steps, as explained in [“Preserving Rights When Restoring File System Data on NetWare” on page 379](#).

Procedure

To restore an eDirectory database on a server using the eMBox Client:

- 1** Make sure you have gathered the backup files you need, as described in [“Preparing for a Restore” on page 384](#).

- 2** Run the eMBox Client in interactive mode:

- ◆ NetWare and UNIX: At the command line, enter `edirutil -i`.
- ◆ Windows: Run
`drive\novell\nds\edirutil.exe -i`

The edirutil file gives you a shortcut to running the eMBox Client. It points to the Java executable and the default location where the eMBox Client is installed with eDirectory, and for NetWare, it includes the necessary `-ns` option. (You can also enter the information manually, as described in [“Running the eMBox Client on a Workstation” on page 467](#).)

When the eMBox Client opens, the eMBox Client prompt appears: eMBox Client>

- 3** Log in to the server you want to restore by entering

```
login -s server_name_or_IP_address -p port_number
-u username.context -w password
```

For example, on Windows enter

```
login -s 151.155.111.1 -p 8009 -u admin.mycompany -w mypassword
```

If you get an error saying that a secure connection cannot be established, make sure your machine has the JSSE files listed in [“Establishing a Secure Connection with the eMBox Client” on page 473](#).

For help finding out which port number to use, see [“Finding Out eDirectory Port Numbers” on page 473](#).

The eMBox Client indicates whether the login is successful.

- 4 Enter the restore command at the eMBox Client prompt, following this general pattern:

```
restore -r -a -o -f full_backup_path_and_filename  
-d roll-forward_log_location -l restore_log_path_and_filename
```

A space must be between each switch. The order of the switches is not important. Make sure you use the `-r` switch to restore the eDirectory database itself; otherwise only the other kinds of files will be restored. If you want the database to be active and open when the restore is complete, make sure you specify `-a` and `-o`.

If you are restoring roll-forward logs, make sure you include the full path to the logs, including the directory that is automatically created by eDirectory, usually named `\nds.rfl`. (For more information about this directory, see [“Location of the Roll-Forward Logs” on page 382](#).)

For example:

```
restore -r -a -o -f sys:/backup/nds.bak -d voll:/rfl_dir/nds.rfl -l sys:/backups/backup.log
```

This example command specifies that the database itself should be restored (`-r`), and it should be activated (`-a`) and opened (`-o`) after the restore verification is successfully completed. The `-f` switch indicates where the full backup file is, `-d` the roll-forward logs, and `-l` the log file in which to record the results of the restore.

The eMBox Client will restore the full backup, then prompt you for the incremental backup files.

- 5 (Conditional) If you are restoring incremental backup files, provide the path and filename for each one when the eMBox Client prompts you for the next incremental file.

It will tell you the ID of the next file, which you can find in the incremental backup file header.

The eMBox Client indicates whether the restore was successful.

- 6 (Conditional) If the restore was not successful, check the log file to see the errors.

If the restore verification fails, see [“Recovering the Database If Restore Verification Fails” on page 411](#).

NOTE: If the server you are restoring shares a replica with a server running an earlier version than eDirectory 8.5, the restore log will show a -666 error (incompatible DS version) for that replica. For more information on this situation and what you might be able to do, see [“Restore Verification Is Backward Compatible Only with eDirectory 8.5 or Later” on page 378](#).

- 7 Log out from the server by entering the following command:

```
logout
```

- 8 Exit the eMBox Client by entering the following command:

```
exit
```

- 9 Make sure the server is responding as usual.

- 10 (Conditional) If you are using roll-forward logging on this server, you must re-create your configuration for roll-forward logging to make sure it is turned on and the logs are being saved in a fault-tolerant location. After turning on the roll-forward logs, you must also do a new full backup.

This step is necessary because during a restore, the configuration for roll-forward logging is set back to the default, which means that roll-forward logging is turned off and the location is

set back to the default. The new full backup is necessary so that you are prepared for any failures that might occur before the next unattended full backup is scheduled to take place.

For more information about roll-forward logs and their location, see [“Using Roll-Forward Logs” on page 380](#).

Your restore should now be complete. If you use roll-forward logging, you have prepared for any failures in the future by turning on roll-forward logging again after the restore and creating a new full backup as a baseline.

Backup and Restore Command Line Options

The eDirectory Backup eMTool command line options are divided into six functions: **backup**, **restore**, **restadv**, **getconfig**, **setconfig**, and **cancel**.

The switches can be placed in any order in the command after the name of the function. They must be separated by a space.

Option and Switches	Description
backup	Perform a backup of the database and associated files.
-f <i>file_name</i>	(Mandatory) Backup filename and path Specifies the filename and location of the backup file you want the Backup eMTool to create. This file must be on the server you are backing up. For example, <code>backup -f vol1:\backup\ndsbak.bak</code> will back up the database to <code>vol1:\backup\ndsbak.bak</code> .
-l <i>file_name</i>	(Mandatory) Log filename and path Specifies the log file to record the results of the backup operation.
-b	(Optional) Perform a full backup Performs a full backup of the eDirectory database. This option is the default behavior. If neither <code>-i</code> nor <code>-c</code> is specified, a full backup is performed.
-i	(Optional) Perform an incremental backup Performs an incremental backup of the eDirectory database. This will back up any changes made to the database since the last full or incremental backup.
-t	(Optional) Back up stream files Includes the stream files when backing up the eDirectory database.

Option and Switches	Description
<code>-u file_name</code>	<p>(Optional) User includes filename and path</p> <p>Specifies an include file that lists additional files to back up. You can create this configuration file to include other files in the backup that might be important when restoring the server's eDirectory database.</p> <p>In the include file, list the full path of each file you want backed up, followed by a semicolon (;). For example, if an administrator wanted to include the <code>autoexec.ncf</code> and <code>hosts</code> file in the backup for a NetWare server, the text in the user include file would be the following:</p> <pre>sys:\system\autoexec.ncf;sys:\etc\hosts;</pre> <p>Don't include any spaces or hard returns in the list of files.</p> <p>To confirm that these files are being backed up, check the backup log or look at the header of the backup file. (See "Format of the Backup Log File" on page 376 and "Format of the Backup File Header" on page 372.)</p> <p>WARNING: When opening a backup file, just view the header—make sure you don't try to save or modify the file, or it might become truncated. Most applications can't save the binary data correctly.</p>
<code>-s file_size</code>	<p>(Optional) Backup file size limit (bytes)</p> <p>Specifies the maximum size (in bytes) of the backup file. You might want to use this option if you are concerned about file size because of the media you are using to store the backup files after they are created.</p> <p>If the maximum size is reached, a new backup file is created with the same name as the first with a five-digit hex extension added to denote what file it is. This extension increments with each new file.</p> <p>For example, you could set the maximum size of the backup files to 1 MB using the following switches as part of your command: <code>backup -f vol1:/backup/mydib.bak -s 1000000</code>. If the database is 3.5 MB, this is the resulting set of backup files:</p> <pre>vol1:/backup/mydib.bak, size is 1 MB vol1:/backup/mydib.bak.00001, size is 1 MB vol1:/backup/mydib.bak.00002, size is 1 MB vol1:/backup/mydib.bak.00003, size is .5 MB</pre> <p>The smallest possible size is about 500 KB. The first file could be larger, depending on how many files are being included with the backup.</p> <p>The first file contains an attribute under the backup tag called <code>number_of_files</code>. This is the total number of files in the backup set. For the above example, this number would be 4. Also, the header of each backup file contains an attribute called <code>backup_file</code>. This is the original name of the file. (For more information, see "Format of the Backup File Header" on page 372.)</p> <p>When restoring a set of backup files like the set in the example above, the command would be</p> <pre>restore -f vol1:/backup/mydib.bak -l log_file_path_and_filename</pre> <p>The Backup eMTool identifies that there are multiple files and looks for them in the same directory as the first, but with the above name mutations.</p> <p>TIP: The backup files can also be made much smaller using a third-party file compression tool. They compress approximately 80%.</p>

Option and Switches	Description
-w	<p>(Optional) Overwrite existing backup file of same name</p> <p>Overwrites the backup file specified with the -f switch if a file of the same name already exists. If this option is not used and a file of the same name already exists, in interactive mode the Backup eMTool will ask you whether to overwrite or not. In batch mode, if a file of the same name exists and -w is not specified, the default behavior is to not overwrite the file, so a backup will not be created.</p> <p>If you are making a file system backup shortly after each full or incremental backup of eDirectory, your previous backup files should have been copied from the server to file system backup tapes, so it should be safe to use this option to overwrite the existing backup file.</p> <p>IMPORTANT: Use this option in your batch files for unattended backups. If a backup file of the same name exists (this is likely if you use the same batch file regularly), it's important to use the -w option to overwrite the existing backup file to make sure your backup is successful.</p> <p>In batch mode, if -w is not specified and a file of the same name exists, the default behavior is to not overwrite the file, so a backup will not be created. (In interactive mode, if -w is not specified, the eMBox Client will ask you whether you want to overwrite the file.)</p>
-c	<p>(Optional) Perform a cold backup</p> <p>Performs a full backup of the database, but closes the database before the backup. After the backup has completed, the database reopens unless the -o or -o and -d switches are used.</p>
-o	<p>(Optional) Leave database closed after cold backup</p> <p>Can be used only if the -c switch is also used. Leaves the database closed after a cold backup. This option is helpful when upgrading hardware or moving a server to a new machine with the same operating system (as described in "Upgrading Hardware or Replacing a Server" on page 447).</p>
-d	<p>(Optional) Disable DS agent after a cold backup</p> <p>Can be used only if both the -c and -o switches are also used. Disables the DS agent after a cold backup. This option is helpful when upgrading hardware or moving a server to a new machine with the same operating system (as described in "Upgrading Hardware or Replacing a Server" on page 447).</p> <p>The DS agent is disabled by setting the login disabled attribute on the pseudo server. This results in a -663 error when eDirectory starts.</p>
restore	Perform a restore of the database and associated files.
-f <i>file_name</i>	<p>(Mandatory) Backup filename and path</p> <p>Specifies which full backup to restore from. This file must be located on the server being restored. For example, restore -f vol1:/backup/ndsbak.bak will restore from the file vol1:/backup/ndsbak.bak.</p> <p>If the backup was made up of more than one file, all the files in the set must be copied into the same directory on the server.</p>
-l <i>file_name</i>	<p>(Mandatory) Log filename and path</p> <p>Specifies the log file to record the results of the restore operation.</p>

Option and Switches	Description
-r	<p>(Optional) Restore DIB set</p> <p>Specifies that the eDirectory database should be restored.</p> <p>WARNING: If you omit this option, the eDirectory database itself will not be restored. The only files that will be restored are other kinds of files you specify.</p>
-d <i>dir_name</i>	<p>(Optional) Roll-forward log directory</p> <p>Specifies the directory where the roll-forward logs are located. This must be the entire path and must be on the server being restored. All the roll-forward logs must be in the directory specified and they must have the same filenames as they did at the time of creation.</p> <p>After the database is restored, the changes recorded in these logs are replayed against the database to bring it up to date. If the -d switch is not used, the Backup eMTool does not replay any logs against the database, even if roll-forward logging was turned on at the time of the backup.</p> <p>To determine the first required roll-forward log, open the last backup file being restored in a text editor and read the <code>current_log</code> attribute of the backup tag. The last backup file being restored is either the full backup file specified by the -f option or the last incremental backup file that is to be applied during the restore. (For more information about the attributes listed in the header, see “Format of the Backup File Header” on page 372.)</p> <p>WARNING: When opening a backup file, just view the header—make sure you don't try to save or modify the file, or it might become truncated. Most applications can't save the binary data correctly.</p>
-u	<p>(Optional) Restore user included files</p> <p>Restores the user files that were included with the backup of the database.</p> <p>As part of the backup, you can create a text file containing a list of files that you want backed up along with the database, and specify that file as the user includes file. These files will not be available to restore unless they were included in the backup.</p>
-a	<p>(Optional) Activate DIB after verifying</p> <p>Renames the database from RST to NDS after the restore verification completes successfully. (For an overview of the process, see “Overview of How the Backup eMTool Does a Restore” on page 371.)</p>
-o	<p>(Optional) Open database when finished</p> <p>Directs the Backup eMTool to open the database when the operation is complete. If the restore verification is successful, it opens the restored database. If the restore verification fails, this option opens the database that was on the machine before the restore was performed. (For an overview of the process, see “Overview of How the Backup eMTool Does a Restore” on page 371.)</p>

Option and Switches	Description
-n	<p>(Optional) Do not verify database after restore</p> <p>Directs the Backup eMTool to restore the database without verifying. The transitive vector of this server will not be compared with the one expected by other servers in the replica rings it participates in. (For information about transitive vectors, see “Transitive Vectors and the Restore Verification Process” on page 378). The database is not renamed from RST to NDS unless another option is used to do so.</p> <p>IMPORTANT: We do not recommend using this option unless suggested by Novell Support.</p>
-v	<p>(Optional) Override restore</p> <p>Renames the database from RST to NDS without trying to verify.</p> <p>IMPORTANT: We do not recommend using this option unless suggested by Novell Support.</p>
-k	<p>(Optional) Remove lockout on database</p> <p>Removes the lockout on the NDS database.</p>
restadv	Advanced restore options. (NOTE: The DS agent will be closed for all advanced restore options.)
-l <i>file_name</i>	<p>(Mandatory) Log filename and path</p> <p>Specifies the log file to record the results of the restore operation.</p>
-o	<p>(Optional) Open database when finished</p> <p>Directs the Backup eMTool to open the database when the operation is complete. If the restore verification is successful, it opens the restored database. If the restore verification fails, this option opens the database that was on the machine before the restore was performed. (For an overview of the process, see “Overview of How the Backup eMTool Does a Restore” on page 371.)</p>
-n	<p>(Optional) Try to verify a previously failed restore</p> <p>Tries to verify a previously restored RST database.</p>
-m	<p>(Optional) Remove restored DIB files</p> <p>Removes the RST database if it is present.</p>
-v	<p>(Optional) Override restore</p> <p>Renames the database from RST to NDS without trying to verify.</p> <p>IMPORTANT: We do not recommend using this option unless suggested by Novell Support.</p>
-k	<p>(Optional) Remove lockout on database</p> <p>Removes the lockout on the NDS database.</p>
getconfig	Retrieves the current roll-forward log configuration.

Option and Switches	Description
	<p>No options are needed.</p> <p>Displays the current settings. For example, on a server with roll-forward logging turned off, the getconfig command would return information like the following:</p> <pre data-bbox="634 369 1377 594"> Roll forward log status OFF Stream file logging status OFF Current roll forward log directory voll:/rfl/nds.rfl Minimum roll forward log size (bytes) 104857600 Maximum roll forward log size (bytes) 4294705152 Last roll forward log not used 00000000.log Current roll forward log 00000001.log *** END *** </pre>
setconfig	Sets the roll-forward log configuration.
-L	<p>(Optional) Start keeping roll-forward logs</p> <p>Turns on roll-forward logging. (Default=Off) Using continuous roll-forward logging lets you restore a server to the state it was in at the moment before it went down, instead of just to the last full or incremental backup.</p> <p>You must use roll-forward logging for servers that participate in replica rings, so that you can restore a server back to the synchronization state that the other servers expect.</p> <p>Administrative intervention is required after the roll-forward logs have been turned on. If left unchecked, the roll-forward logs continue to grow until they fill up the disk partition/volume. If roll-forward logs cannot be created because no more disk space is available, eDirectory stops responding on that server. Periodically, it is necessary to back up and delete unused logs. See “Backing Up and Removing Roll-Forward Logs” on page 383.</p> <p>For more information, see “Using Roll-Forward Logs” on page 380.</p>
-I	<p>(Optional) Stop keeping roll-forward logs</p> <p>Turns off roll-forward logging. (Default=off.) The database reuses the current roll-forward log instead of saving a consecutive set of logs. If the roll-forward logs are turned off, you can restore eDirectory only to the point of the last full or incremental backup.</p> <p>If the logs are turned off unintentionally, you need to turn them back on and then do a new backup of the database to ensure that you can make a full recovery.</p> <p>For more information, see “Using Roll-Forward Logs” on page 380.</p>
-T	<p>(Optional) Start logging of stream files</p> <p>(Only applicable if the roll-forward logs are turned on.) Copies the entire stream file into the roll-forward log if a stream file is modified. Stream files are additional information files that are related to the database, such as login scripts.</p> <p>Roll-forward logs will fill disk space faster when stream files are being logged. Make sure you monitor disk space on the disk partition/volume where roll-forward logs are placed. If roll-forward logs cannot be created because no more disk space is available, eDirectory stops responding on that server.</p>

Option and Switches	Description
-t	<p>(Optional) Stop logging of stream files</p> <p>Stops copying the entire stream file into the roll-forward log if a stream file is modified. If roll-forward logging of stream files is turned off, you can use the backup options to back up stream files during full and incremental backup. Backing them up this way might be sufficient if your stream files don't change often.</p> <p>Turning off logging of stream files can help slow the growth of roll-forward logs.</p>
-r <i>dir_name</i>	<p>(Optional) Set roll-forward log directory</p> <p>Changes the directory where the roll-forward logs are placed. For example, if the command used was <code>setconfig -r vol2:\rfl</code>, a directory is created under <code>vol2:\rfl</code> and the roll-forward logs are placed in it.</p> <p>This directory name is based on the name of the current eDirectory database. For typical installs this is NDS, so the final directory name would be <code>vol2:\rfl\nds.rfl\</code>. If you renamed the eDirectory database from NDS to ND1, the roll-forward log directory would be changed to <code>vol2:\rfl\nd1.rfl\</code>.</p> <p>You can find out the current location by entering the <code>getconfig</code> command.</p> <p>When you change the location, the new directory is created immediately, but a roll-forward log is not created there until a transaction takes place in the database.</p> <p>IMPORTANT: The backup tool has no way of tracking the changes to the roll-forward log directory. When restoring the database, you must collect all roll-forward logs and place them in one directory on the server.</p> <p>For more information, see "Using Roll-Forward Logs" on page 380.</p>
-n <i>file_size</i>	<p>(Optional) Set minimum roll-forward log size</p> <p>Sets the minimum size of the roll-forward log files (in bytes). When the minimum size is reached, the database starts a new roll-forward log after the current transaction is finished.</p>
-m <i>file_size</i>	<p>(Optional) Set maximum roll-forward log size</p> <p>Sets the maximum size for the roll-forward log files (in bytes). If this limit is reached and a transaction is in progress, the transaction is continued over into the next file. This setting must always be larger than the minimum size.</p>
-s	<p>(Optional) Start a new roll-forward log</p> <p>Starts a new roll-forward log at the end of the current transaction. The new file is created at the beginning of the next transaction.</p>
cancel	<p>Cancels any running backup or restore operation. No options are needed.</p>

Using DSBK.NLM on NetWare

Dsbk is a thin command line parser that performs the same operations as the Backup eMTool, but lets you initiate a backup from the server console without having to log in first or set up Role-Based Services (see [Chapter 17, "The eDirectory Management Toolbox," on page 465](#)). It runs as an NLM on the server, using the same command line options as the Backup eMTool. This utility can also be used in scripting backups using NCF files on the server.

IMPORTANT: Dsbk will not restore incremental backups. You can only use it to restore full backups.

After a dsbk operation has completed, the results of the operation are written to a file (dsbk.err) that you can programmatically open and view the results. The first four bytes of this file contain error codes if any are generated during the operation. If there are no errors, the first four bytes of this file will contain zeros.

To use dsbk.nlm:

- 1** Download and install **eDirectory 8.7.3 IR3** (<http://support.novell.com/cgi-bin/search/searchtid.cgi?/2969860.htm>).
- 2** Make sure that dsbk.nlm is in the sys:\system directory.
Dsbk must be located in the same directory as backupcr.nlm, the core library that contains all backup and restore functionality. This library has no user interface; it is loaded and linked dynamically by the dsbk utility.
- 3** At the server console, run the following command with any of the options listed in **“Backup and Restore Command Line Options”** on page 403:

```
load dsbk
```

Changes to Server-Specific Information Backup (NetWare Only)

In many NetWare installations, administrators have been creating backups of server-specific information. With the release of eDirectory 8.6, the structure of the eDirectory schema was changed. Further changes were included with the release of eDirectory 8.7. However, server-specific information backups created by filesystem TSA or third-party backup tools were not supported by the changes. Instead, the database changes were supported in a new “hot backup” facility provided by the Backup eMTool in Novell iManager or by the eMBox client. Support for backup of server-specific information using filesystem TSA was not included at that time. In eDirectory 8.7.3, this is now supported using the hot backup functionality. As in previous versions, filesystem TSA calls the dsbacker.nlm to create the backup, but now dsbacker.nlm calls the backupcr.nlm, which creates a backup using the Backup eMtool functionality.

Effective backups can be created and restored using the following recommendations for various NetWare and eDirectory versions.

eDirectory version	NetWare version	Backup/Restore Method Recommendations
8.6 or earlier	Any version	To restore a backup of server-specific information (SSI) using filesystem TSA: <ul style="list-style-type: none"> ◆ Do not delete the volume or server objects associated with the downed server. ◆ Call Novell Support for detailed instructions.
8.7	5.1 & 6.0	Back up and restore only using the Backup eMtool.
8.7.1 or later	5.1	Back up and restore only using the Backup eMtool.

eDirectory version	NetWare version	Backup/Restore Method Recommendations
8.7.1 or later	6.0 with SP3 (Required for eDirectory 8.7.1)	Use either the Backup eMtool, filesystem TSA, or third-party tools. Restoration is done using the Backup eMTool.

The main differences in server-specific information in NetWare 6.0 with eDirectory 8.7.1 are as follows:

- ♦ **Bigger File Size:** The former method of SSI backup contained only a small portion of the database. Now, because the backup file contains all the information about all directory objects on the server, it is much bigger. It will be roughly the same size as the database.
- ♦ **User-Defined File Location:** In former versions of server-specific information backup, only one file, servedata.nds, was created in the sys:system directory. Because the file was smaller, it was not critical where the data was placed before copying off to tape. With eDirectory 8.7.3 you can use filesystem TSA to create a full backups of the database. Three files are involved. For one of these, ssiback.bak, the file location is user defined.

File	Description	Location
ssiback.bak	This backup file is the same as the full "hot backup" created with the Backup eMTool. See "About the eDirectory Backup eMTool" on page 369 .	User defined. The default is sys:system. Because of file size, we recommend it be relocated onto a volume other than sys:.
ssiback.ini	A text file containing the path where the ssiback.bak file is located. Default backup file location is sys:system. For example: vol1:/backups/ssibackup.bak.	sys:system
ssiback.log	A high-level view containing information about previous backups. The log file contains a history of all backups, records backup start time and end time, and contains information about possible errors during the backup process.	sys:system

- ♦ **Restore Using Backup eMTool:** The server-specific information can only be restored using the Backup eMTool.

Recovering the Database If Restore Verification Fails

The restore process includes a verification step, which compares the eDirectory database on the server being restored to other servers in the replica ring by comparing the transitive vectors. (For more information on the restore process, see ["Overview of How the Backup eMTool Does a Restore" on page 371](#) and ["Transitive Vectors and the Restore Verification Process" on page 378](#).)

If the transitive vectors do not match, the verification fails. This usually indicates that data is missing from the files you used for the restore. For example, data might be missing for the following reasons:

- ◆ You did not turn on roll-forward logging before the last backup was performed.
- ◆ You did not include the roll-forward logs in the restore.
- ◆ The set of roll-forward logs you provided for the restore was not complete.

NOTE: Another issue that causes the restore verification to fail is participating in a replica ring with a server running a version of eDirectory that is earlier than 8.5. For more information on this situation and what you might be able to do, see ["Restore Verification Is Backward Compatible Only with eDirectory 8.5 or Later" on page 378](#).

By default the restored eDirectory database will not open after the restore if it is inconsistent with the other replicas.

If you have all the backup files and roll-forward logs necessary for a complete restore but forgot to provide all of them during the process, you can simply run the restore again with a complete set of files. If the restore is complete on a second try, the verification can succeed and the restored database will open.

If you do not have all the backup files and roll-forward logs necessary to make the restore complete so that verification will be successful, you must follow the instructions in this section to recover the server. Here is an outline of what you can recover if verification fails:

- ◆ You can still recover the server's identity and file system rights.
- ◆ You cannot recover any replicas on this server from backup, but the server can still be used for the replicas it contained after you follow the recovery procedure in this section. You must remove the server from the replica ring and use advanced Restore options and the DSRepair Tool to bring the server to a state where it can be put back in the replica ring. Then you can re-add the desired replicas to it.
- ◆ Unfortunately, if this server had the sole copy of any partition of the database (there were no other replicas of the partition), the partition cannot be recovered.

Use the instructions in this section after verification fails to recover the server's identity and file system rights, and to remove and re-add it to the replica ring. When you have followed these steps and the replication process is complete, the server should function as it did before the failure (with the exception of any partitions that were not replicated and, therefore, can't be recovered).

First, complete ["Cleaning Up the Replica Ring" on page 412](#). Then continue with ["Repair the Failed Server and Readd Replicas to the Server" on page 414](#).

Cleaning Up the Replica Ring

This procedure explains how to

- ◆ **Reassign master replicas.** If the failed server holds a master replica of any partition, you must use DSRepair to designate a new master replica on a different server in the replica list.
- ◆ **Remove replica list references to the failed server.** Each server participating in replica rings that included the failed server must be told that the failed server is no longer available.

Prerequisites

- eDirectory is installed on the machine where you are trying to restore the failed server.
- A restore was attempted, and the restore verification failed.

- ❑ The NDS database is open and running, and the database named RST is still on the machine (left there by the restore process).
- ❑ You know which replicated partitions were stored on the failed server. The replicas this server held are listed in the header of the backup file.

Procedure

To clean up the replica ring:

- 1** At the console of one of the servers that shared a replica with the failed server, load DSRepair with the switch that lets you access the advanced options.
 - ◆ NetWare and Windows: Use the -a switch.
 - ◆ UNIX: Use the -Ad switch.

For more information on how to run DSRepair with advanced options using the -a or -Ad switches, see “[Advanced DSRepair Options](#)” on page 221.

WARNING: If you use DSRepair with -a or -Ad, some of the advanced options can cause damage to your tree. For more information on these options, refer to the [Novell Support Web site, Solution 2938493 \(http://support.novell.com/servlet/tidfinder/2938493\)](http://support.novell.com/servlet/tidfinder/2938493).

- 2** Select Replica and Partition Operations.
- 3** Select the partition you want to edit, so you can remove the failed server from the replica ring of that partition.
- 4** Select View Replica Ring to see a list of servers that have replicas of the partition.
- 5** (Conditional) If the failed server held the master replica, select another server to hold the master by selecting Designate This Server As the New Master Replica.

The replica ring now has a new master replica. All replicas participating in the ring are notified that there is a new master.

- 6** Wait for the master replica to be established. Make sure the other servers in the ring acknowledge the change before proceeding.
- 7** Go back to View Replica Ring. Select the name of the failed server, then select Remove This Server from the Replica Ring.

If you have not loaded DSRepair with -a or -Ad (depending on the platform) for advanced options, you will not see this option in the list.

WARNING: Make sure you do not do this if the failed server is designated as the master replica. You can see this information in the list of servers in the ring. If it is the master, designate a different server as the master as noted in [Step 5](#). Then, come back to this step and remove the failed server from the replica ring.

- 8** Log in as Admin.
- 9** After reading the explanation message, enter your agreement to continue.
- 10** Exit DSRepair.

All servers participating in that replica ring are notified.
- 11** Repeat this procedure on one server for each replica ring that the failed server participated in.

To finish preparing the failed server to get new copies of the replicas, continue with the next procedure, “[Repair the Failed Server and Readd Replicas to the Server](#)” on page 414.

Repair the Failed Server and Readd Replicas to the Server

This procedure lets you change the replica information on the server to external references, so that the server does not consider itself to be part of a replica ring. After you remove the replicas from the server in this way, you can unlock the database.

After removing the replicas, you complete the procedure by readding the replicas to the server. This way, the server receives a new, up-to-date copy of each replica. When each replica has been readded, the server should function as it did before the failure.

To remove replicas using DSRepair, and re-add them using replication:

- 1** Make sure you have completed [“Cleaning Up the Replica Ring” on page 412](#).
- 2** Override the restore on the server using the advanced restore option in the eMBox Client.

2a Run the eMBox Client in interactive mode:

- ◆ NetWare and UNIX: At the command line, enter `edirutil -i`.
- ◆ Windows: Run `drive\novell\nds\edirutil.exe -i`

The edirutil file gives you a shortcut to running the eMBox Client. It points to the Java executable and the default location where the eMBox Client is installed with eDirectory, and for NetWare, it includes the necessary -ns option. (You can also enter the information manually, as described in [“Running the eMBox Client on a Workstation” on page 467](#).)

When the eMBox Client opens, the eMBox Client prompt appears: eMBox Client>

2b Log in to the server you want to restore by entering

```
login -s server_name_or_IP_address -p port_number -u
username.context -w password
```

For example, on Windows enter

```
login -s 151.155.111.1 -p 8008 -u admin.mycompany -w mypassword
```

If you get an error that says a secure connection cannot be established, check your machine for the JSSE files listed in [“Establishing a Secure Connection with the eMBox Client” on page 473](#).

For help finding out which port number to use, see [“Finding Out eDirectory Port Numbers” on page 473](#).

The eMBox Client indicates whether the login is successful.

2c Specify the advanced restore option to override the restore, then specify a log filename:

```
restadv -v -l logfilename
```

This advanced restore option will rename the RST database (the database that was restored but failed the verification) to NDS, but keep the database locked.

- 3** At the server console, change all the replica information on the server into external references using advanced options in DSRepair.
 - ◆ NetWare: Enter `dsrepair -xk2 -rd`
 - ◆ Windows: Click Start > Settings > Control Panel > Novell eDirectory Services. Select dsrepair.dlm. In the Startup Parameters field, type `-xk2 -rd`. Click Start.
 - ◆ UNIX: Enter `ndsrepair -R -Ad -xk2`

The `-rd` or `-R` switch repairs the local database and the replica.


WARNING: If used incorrectly, DSRepair advanced options can cause damage to your tree. For more information on these options, refer to the [Novell Support Web site, Solution 2938493 \(http://support.novell.com/servlet/tidfinder/2938493\)](http://support.novell.com/servlet/tidfinder/2938493).

- 4** When the repair is finished, remove the lockout and open the database using the following advanced restore options in the eMBox Client:

```
restadv -o -k -l logfilename
```

The `-o` opens the database and the `-k` removes the lockout.

- 5** Use iManager to add the server back into the replica ring:

5a In Novell iManager, click the Roles and Tasks button .

5b Click Partition and Replicas > Replica View.

5c Specify the name and context of the partition you want to replicate, then click OK.

5d Click Add Replica.

5e Next to the Server Name field, click the Browse button , then select the server you just restored.

5f Select the type of replica you want, click OK, then click Done.

5g Repeat these steps for each replica ring that the server was participating in.

- 6** Wait for the replication process to complete.

The replication process is complete when the state of the replicas changes from New to On. You can check the state in iManager. See [“Viewing Information about a Replica” on page 124](#) for more information.

- 7** (Conditional) If you want to use roll-forward logging on this server, you must re-create your configuration for roll-forward logging to make sure it is turned on and the logs are being saved in a fault-tolerant location. After turning on the roll-forward logs, you must also do a new full backup.

This step is necessary because during a restore, the configuration for roll-forward logging is set back to the default, which means that roll-forward logging is turned off and the location is set back to the default. The new full backup is necessary so that you are prepared for any failures that might occur before the next unattended full backup is scheduled to take place.

For more information about roll-forward logs and their location, see [“Using Roll-Forward Logs” on page 380](#).

Scenarios for Backup and Restore

- ♦ [“Scenario: Losing a Hard Drive Containing eDirectory in a Single-Server NetWork” on page 416](#)
- ♦ [“Scenario: Losing a Hard Drive Containing eDirectory in a Multiserver Environment” on page 416](#)
- ♦ [“Scenario: Losing an Entire Server in a Multiple-Server Environment” on page 419](#)
- ♦ [“Scenario: Losing Some Servers in a Multiple-Server Environment” on page 419](#)
- ♦ [“Scenario: Losing All Servers in a Multiple-Server Environment” on page 419](#)

Scenario: Losing a Hard Drive Containing eDirectory in a Single-Server Network

Indira is the administrator for a single-server network at Stationery Supply, Inc. Indira can't rely on replication for fault tolerance, because her environment has only one server. The new Backup eMTool functionality in eDirectory 8.7.3 provides a simple solution for Indira to back up and restore eDirectory. It's server-centric and it's fast.

After upgrading her Windows NT server from eDirectory 8.6.2 to eDirectory 8.7.3, Indira sets up unattended backups for her server using batch files to run the Backup eMTool.

Indira wants to do a full backup of eDirectory every Sunday night, and an incremental backup every weeknight. She sets the unattended backups to run shortly before her full and incremental file system backups each night, so her tape backups contain the eDirectory backup files as well as the file system data. She has contracted with a remote data storage company to send the tape backups offsite.

Every Monday morning, Indira checks the backup log to make sure the full backup was successful. She also checks the logs occasionally during the week to make sure the incremental backups were successful.

Indira decides not to turn on roll-forward logs for the following reasons:

- ◆ She does not have a separate storage device on her server, so turning on roll-forward logs would not provide any additional backup of eDirectory. If there were a storage device failure, the logs would be lost along with eDirectory, so there is no point in creating them.
- ◆ The tree does not change very much, and she is satisfied with being able to restore only up to last night's backup. She doesn't need to be able to restore eDirectory to the moment before a failure.
- ◆ Because the server does not participate in a replica ring with other servers, roll-forward logs are not required for the restore verification process to be successful.

Stationery Supply, Inc. decides to reorganize the staff, so Indira does a manual backup before and after making significant changes to the tree. Her strategy is to make a new backup of changes during the middle of a weekday when necessary, instead of running roll-forward logs all the time.

To make sure her backup strategy is ready to go when she needs it, Indira tests it occasionally. She doesn't have the budget to purchase a second server for testing, so she makes arrangements with a test lab in her town. Using a server like hers in the test lab, she installs her operating system and tries to approximate the environment of her eDirectory database. She restores her backups and checks to make sure eDirectory is restored as she expects.

One Wednesday morning, the hard drive containing eDirectory on the server has a failure. Indira obtains a new hard drive and the backup files from the full backup on Sunday evening, the incremental backup on Monday evening, and the incremental backup on Tuesday evening. She installs the new hard drive and installs eDirectory on it. Then she restores the full and incremental backups. Any changes to the tree that were made on Wednesday morning before the hard drive failure are lost because Indira was not running roll-forward logs on the server. But Indira is satisfied with restoring only to last night's backup; she doesn't feel that running roll-forward logs would be worth the administrative overhead.

Scenario: Losing a Hard Drive Containing eDirectory in a Multiserver Environment

Jorge at Outdoor Recreation, Inc. has 10 servers running eDirectory. He does full backups every Sunday night and incremental backups nightly, running the eDirectory backup shortly before the file system backup to tape.

All of the servers are participating in replica rings. Jorge uses roll-forward logging for all the servers. On each of his servers, he has placed the roll-forward logs on a different storage device than eDirectory. He monitors the free space and rights on those storage devices to make sure the roll-forward logs don't fill up the storage device. Occasionally he backs up the roll-forward logs to tape and removes all except the one in use by eDirectory, to free up space.

The administrative overhead of turning on continuous roll-forward logging is worth it to Jorge, because it gives him the up-to-the moment backup required for servers that participate in replica rings. This way, if he needs to restore a server, the restored server will match the synchronization state that other servers in the replica ring expect.

In his test lab, Jorge periodically tests his backup files to make sure his backup strategy will meet his goals.

One Thursday at 2:00 p.m., the Linux server named Inventory_DB1 has a hard drive failure on the drive containing eDirectory.

Jorge needs to gather the last full backup and the incremental backups since then, which will restore the database up to the point of last night's incremental backup at 1:00 a.m. The roll-forward logs have been recording the changes to the database since last night's backup, so Jorge will include them in the restore to bring the database back to the state it was in just before the hard drive failure.

Jorge takes the following steps:

1. He gets a replacement hard drive for the server.
2. He gets the tape of the full backup for the server from the previous Sunday night.

The batch file he uses to run full backups every Sunday night places the backup file in /adminfiles/backup/backupfull.bk.

He had specified a file size limit of 200 MB in the backup configuration settings, so there are two backup files:

backupfull.bk.00001 (250 MB)
backupfull.bk.00002 (32 MB)

3. He also gets the tapes containing the incremental backups for Monday, Tuesday, and Wednesday nights.

The batch file he uses to run incremental backups every weeknight places the backup file in /adminfiles/backup/backupincr.bk.

Because he runs the same batch file every weeknight for the incremental backups of eDirectory, they all have the same filename. He needs to give them new names when he copies them back onto the server, because they all must be placed in the same directory during the restore.

4. Jorge installs the replacement hard drive.

In this case, the Linux operating system for the server was not on the hard drive that failed, so he does not need to install Linux.

5. Jorge restores the file system from tape backup for the disk partitions that were affected.
6. Jorge reinstalls eDirectory, putting the server into a new temporary tree (the restore puts it back into the original tree again later).
7. Jorge creates an /adminfiles/restore directory on the server, to hold the files to be restored.
8. He copies the full backup (the set of two files) into that directory.

9. He copies the incremental backups for Monday, Tuesday, and Wednesday nights into the directory.

Each of them is named backupincr.bk, so when he copies them into the directory he changes the filenames to

backupincr.mon.bk

backupincr.tues.bk

backupincr.wed.bk

NOTE: Full and incremental backups aren't required to be in the same directory together, but all the incremental backups must be in the same directory.

10. He uses iManager to restore eDirectory:
 - a. He goes into iManager and clicks eDirectory Maintenance Utilities > Restore.
 - b. He logs in to the server, using the context of the new temporary tree.
 - c. In the Restore Wizard - File Configuration screen, he does the following:
 - Enters /adminfiles/restore for the location where he placed the backup files.
 - Enters /adminfiles/restore/restore.log for the location where the restore log should be created.
 - d. In the Restore Wizard - Optional screen, he does the following:
 - Checks Restore Database.
 - Checks Restore Roll-Forward Logs.
 - Enters the location of the roll-forward logs.
 - (This is the separate location that he created specifically to hold the roll-forward logs. Because he placed them on a different hard drive than eDirectory, the hard drive failure did not affect them and they are still available.)
 - Checks Restore Security Files
 - Checks Activate the Restored Database after Verification.
 - Checks Open the Database after Completion of Restore.
 - Wants eDirectory to open if the restore verification is successful.
11. He starts the restore and enters the filenames of the incremental backup files when prompted.
12. The restore verification is successful, so the database opens, back in its original tree.

The restore verification was successful because roll-forward logs were running on the server when the hard drive failed, and Jorge included the logs in the restore.
13. Jorge re-creates the roll-forward logs configuration on the server after the restore is complete, then he creates a new full backup.

The settings are reset to the default during a restore, which means roll-forward logging is turned off, so he has to turn it back on. The new full backup is necessary so that he is prepared for any failures that might occur before the next unattended full backup is scheduled to take place.

Jorge checks the way the server is running, and it appears to be normal.

Scenario: Losing an Entire Server in a Multiple-Server Environment

Bob is the administrator for 15 servers at GK Designs Company. He does full backups every Saturday night and incremental backups nightly, running the eDirectory backup shortly before the file system backup to tape.

All of the servers are participating in replica rings. Bob uses roll-forward logging for all the servers.

An electrical fire destroys one of the servers in a branch across town. Fortunately, all but one of the partitions held by this server are also replicated on other servers. Bob had turned on roll-forward logs on that server, but they were lost along with all the other server data, so he can't restore the eDirectory database on that server to the state it was in just before the server went down.

However, he is able re-create the server's eDirectory identity by restoring with the existing backup files. Because Bob can't include the roll-forward logs in the restore, the server does not match the synchronization state that the other servers expect (see [“Transitive Vectors and the Restore Verification Process” on page 378](#)), so the restore verification process is not successful. This means that by default the eDirectory database is not opened after the restore.

Bob addresses the situation by removing this server from the replica rings, using DSRepair to change all the outdated replica information on the server to external references, and then re-adding a new copy of each partition to this server using replication from the other servers that hold the up-to-date replicas. (These steps are described in [“Recovering the Database If Restore Verification Fails” on page 411](#).)

The one partition on this server that Bob had not replicated was a container that held network printing objects for the branch office location, such as a fax/printer and a wide-format color printer. This partition information can't be recovered by the method noted above because no other server has a replica. Bob must re-create the objects in that partition, and this time he chooses to replicate them on other servers for better fault tolerance in the future.

Bob also re-creates the roll-forward log configuration after the server is back on line (because the restore turns it off and resets the settings to the default), and creates a new full backup as a baseline.

Scenario: Losing Some Servers in a Multiple-Server Environment

Joe administers 20 servers across three locations. At one location, a pipe bursts and water destroys 5 out of 8 servers.

Joe has eDirectory backups for all the servers. However, all the servers participate in replica rings, and he is concerned about bringing them back into the tree without the roll-forward logs, which were also lost. He is not sure which servers to restore eDirectory on first or how to address inconsistencies between replicas. Because of the complex issues involved, he calls Novell Support for help in deciding how to restore.

Scenario: Losing All Servers in a Multiple-Server Environment

Delores and her team at Human Resources Consulting, Inc. administer 50 servers at one location.

For fault tolerance during normal business circumstances, they have created three replicas of each partition of their tree, so that if one server is down, the objects in the partitions it holds are still available from another server. They have also planned for recovery of individual servers by backing up all their servers regularly with the Backup eMTool, turning on roll-forward logging, and storing the backup tapes at a remote location.

For disaster recovery planning, Delores and her team have also designated two of their servers as DSMASTER servers. They use two servers because their tree is large enough that more than one DSMASTER server is needed to hold a replica of every partition. Every partition in the tree is replicated on one of the two DSMASTER servers. Neither of the two DSMASTER servers hold replicas of the same partition, so there is no overlap between them. This design is an important part of their disaster recovery plan.

In their test lab, Delores and her team periodically test the backups to make sure their backup strategy will meet their goals.

One night the Human Resources Consulting, Inc. building is damaged by a hurricane, and all the servers in the data center are destroyed.

After this disaster, Delores and her team first restore the two DSMASTER servers, which hold replicas of every partition. They use the last full backup and the subsequent incremental backups, but can't include roll-forward logs in the restore because they were lost when the servers were destroyed. Delores and her team planned the DSMASTER servers so that they don't share replicas. Because the two DSMASTER servers do not share replicas, the restore verification process is successful for both servers even though the roll-forward logs are not part of the restore. After the DSMASTER servers are restored, all the objects in the tree for Human Resources Consulting, Inc. are now available again.

The DSMASTER servers are important because Delores and her team can use them to re-create the tree without inconsistencies after a disaster.

They were using roll-forward logs so they could restore a server to the state it was in at the moment before it went down, bringing it back to the synchronization state expected by other servers in the replica ring. This allows the server to resume communication where it left off, and receive any updates it needs from the other replicas to keep the whole replica ring in sync.

However, in this disaster situation, Delores and her team do not have the roll-forward logs. Without the roll-forward logs, only one server in a replica ring can be restored without errors—the first one they restore. For the rest of the servers, the restore verification process will fail because the synchronization states don't match what the other servers expect (see [“Transitive Vectors and the Restore Verification Process” on page 378](#)). If the restore verification fails, the restore process will not activate the restored eDirectory database.

Delores and her team anticipated this, and they have planned for it. They use the two DSMASTER servers as a starting point, which gives them only one replica of each partition. Those servers can be restored without verification errors, and then the replicas they hold can be used as masters to be copied onto all the other servers.

After restoring the DSMASTER servers, restoring the rest of the servers requires some extra steps. Delores and her team must restore each of the remaining servers by doing the following:

- ◆ Making sure that the replicas on the DSMASTER servers are designated as master replicas.
- ◆ Removing all the servers except the DSMASTER servers from the replica rings.
- ◆ Restoring the full and incremental backups for each of the other servers.

Delores and her team know that the restore verification process will fail for the rest of the servers, because they could not use roll-forward logs in the restore for any of the servers. This leaves them with a restored database that is not activated.

- ◆ Activating the restored database, but keeping it locked, using advanced restore options
- ◆ Using DSREPAIR to change all the replica information to external references.

- ◆ Unlocking the restored database.

At this point the server has the same identity it did before but it will not try to synchronize replica information. Instead, it is prepared to receive a new copy of the replicas it held before.

For NetWare servers, Delores and her team make sure that the file system restore takes place after eDirectory is restored.

- ◆ Adding the replicas back on to each server by replicating them from the copy on the DSMASTER server.

Delores and her team have a pretty good idea which replicas were held by each server, but they can read the header of the backup files for each server to see a list of the replicas that were on the server at the time of the last backup.

- ◆ Re-creating the roll-forward log configuration after the servers are back on line (since the restore turns it off and resets the settings to the default), and creating a new full backup as a baseline to prepare for any other failures that might happen before the next unattended full backup is scheduled.

(These steps are explained in more detail in [“Recovering the Database If Restore Verification Fails” on page 411.](#))

Delores and her team have a lot of work to do, but they can get the tree itself up relatively quickly, and they can expect to recover the eDirectory identity for all of their servers.

Backing Up and Restoring NICI

Novell International Cryptography Infrastructure (NICI) stores keys and user data in the file system and in system and user specific directories and files. These directories and files are protected by setting the proper permissions on them using the mechanism provided by the operating system. This is done by the NICI installation program.

Uninstalling NICI from the system does not remove the system or user directories and files. Therefore, the only reason to restore these files to a previous state is to recover from a catastrophic system failure or a human error. It is important to understand that overwriting an existing set of NICI user directories and files might break an existing application.

Backing up and restoring NICI requires two things:

1. Backing up and restoring directories and files.
2. Backing up and restoring specific user rights on those directories and files.

The exact sequence of events required is depends on the platform you are using.

The critical issue with backup and restore is to maintain the exact permissions on the directories and files. NICI's operation and the security it provides depend on these permissions being set properly.

Typical commercial backup software should preserve permissions on the NICI system and user directories and files. Check your backup software to see if it does the job before doing a custom backup of NICI.

Care should be taken to back up the existing NICI directory structure and its contents, if any, before doing a restore. Losing the machine key is unrecoverable. Because the user data and keys could be encrypted using the machine key, losing it would result in a permanent loss of user data.

Doing a restore of just NICI will require knowledge on your part to determine which files must be restored. During restoration, it is important that the correct access rights be restored for the correct owner. On UNIX and Windows systems, the name of the user specific directory reflects the ID of the owner, but on both systems, the owner ID might change between the time of the backup and the time of the restore. For security reasons, the operator must know which account is being restored and determine that the directory name and access rights are assigned accordingly. The mere existence of a user account on the system with the same ID as the one that was backed up does not mean that the current account is the actual owner of the information being restored.

For more information, see [TID10098087, How to Backup NICI 2.7.x and 2.6.x \(http://support.novell.com/cgi-bin/search/searchtid.cgi?/10098087.htm\)](http://support.novell.com/cgi-bin/search/searchtid.cgi?/10098087.htm) and [TID10096647, How to Backup the eDirectory Database and Associated Security Services Files \(http://support.novell.com/cgi-bin/search/searchtid.cgi?/10096647.htm\)](http://support.novell.com/cgi-bin/search/searchtid.cgi?/10096647.htm) in the Novell Knowledgebase.

UNIX

In NICI 2.6.5 and earlier, the `/var/novell/nici` directory contains all the system and user directories and files. In NICI 2.7.0 and later, `/var/novell/nici` is a symbolic link to the `/var/opt/novell/nici` directory that contains the files.

To determine the version of NICI you are using, see the `/etc/nici.cfg` file.

Performing a Backup

The following files and directories should be backed up. Make sure you preserve the rights on all the directories and files.

For NICI Versions Earlier Than 2.7.0

File/Directory Name	Type of File and Special Instructions
<code>/etc/nici.cfg</code>	Configuration file.
<code>/usr/lib/libccs2.so</code>	Symbolic link to the actual library in <code>/usr/lib/</code> .
<code>/usr/lib/libccs2.so.*</code>	The NICI library (the version of the library completes the name).
<code>/var/novell/nici</code>	This directory contains all the system keys, user directories and files/keys, and the programs used to initialize NICI.

For NICI 2.7.0 and Later

File/Directory Name	Type of File and Special Instructions
<code>/etc/nici.cfg</code>	Symbolic link to the <code>/etc/opt/novell/nici.cfg</code> config file.
<code>/etc/opt/novell/nici.cfg</code>	Configuration file.
<code>/usr/lib/libccs2.so</code>	Symbolic link to the actual library in <code>/opt/novell/lib/</code> .
<code>/opt/novell/lib/libccs2.so.*</code>	The NICI library (the version of the library completes the name).
<code>/var/novell/nici</code>	Symbolic link to the <code>/var/opt/novell/nici</code> directory.

File/Directory Name	Type of File and Special Instructions
/var/opt/novell/nici	This directory contains all the system keys, user directories and files/keys, and the programs used to initialize NICI.

Restoring NICI

To restore the NICI configuration files, first determine whether NICI is already installed on the machine by searching for the /etc/nici.cfg file or link.

- 1** If NICI is already installed on the system, take a backup of the existing set up as outlined above.
- 2** Uninstall NICI and remove the /var/novell/nici or /var/opt/novell/nici directory structure. This is to make sure that the existing system keys do not conflict with the restored set.
- 3** Restore the whole structure from the backup store (depending on the version of NICI), remembering to restore the access rights.

We recommend that you follow the above steps, but knowledgeable operator can choose to restore individual files or directories, possibly changing the names of the files or directories and assigning new access rights. This can be done if the nicifk and xmgrcfg.wks files haven't changed from those on the backup store.

The following guidelines for each file/directory are recommended when restoring when NICI is already installed on the box:

File Name	Procedure
xmgrcfg.nif	Can be restored over an existing file.
xarchive.000	Can be restored over an existing file.
User specific directories and files	<p>Take care that the userid in the backup is the same as the user on the box. If the user directory already exists, determine if the user wants to keep the current files or restore them to a previous state. Normally, user configuration files should be restored as a group rather than individually. Be sure to restore the user files under the correct user's correct userid and to restore the rights on the user directory and contents.</p> <p>For example, if BOB had userid 1000 at the time of the backup but now has userid 5000, the files in the backed up directory 1000 should be restored to directory 5000, or BOB's UID must be changed back to 1000.</p> <p>The restore process must not just blindly restore the user directories without input from the operator. In either case, a backup of the existing NICI user directory needs to be done.</p>

NetWare

Before NICI 2.x, the configuration files were kept in sys:_NetWare and different procedures apply. These instructions are valid only for NICI 2.x or later.

Performing a Backup

Back up the `sys:\system\NICI` directory and any subdirectories and access rights. There is only one user on NetWare so the complication of backing up and restoring the user directories as on UNIX and Windows does not exist.

Restoring NICI

If NICI is not installed, restore the `sys:\system\NICI` directory and its contents.

If NICI is installed (as indicated by the presence of the `sys:\system\NICI\nici.cfg` file), take a backup of the existing setup and remove NICI. Copy the whole backup structure from the backup store to restore.

Selective restoration can be done only if the `nicifk` file hasn't changed from the one on the backup store. If it hasn't changed, restore whatever files in the `sys:\system\NICI` directory you want. Generally, the files should be restored as a group, but a knowledgeable operator can choose to restore only certain files or subdirectories.

Windows

Configuration information is kept in the system registry under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Novell\NICI.
```

A second key will identify the version of NICI currently installed. For example:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Novell\NICI (Shared) U.S./Worldwide (128 bit)
```

Performing a Backup

- 1 Backup any registry information under `HKEY_LOCAL_MACHINE\SOFTWARE\Novell\NICI*`
NICI* indicates all registry keys which begin with NICI. There might be more than one.
- 2 Back up the directory, including subdirectories, identified by `HKEY_LOCAL_MACHINE\SOFTWARE\Novell\NICI\ConfigDirectory`.
As with the UNIX systems, remember the access rights on that directory and all subdirectories. See [“Performing a Backup” on page 422](#) for more information.

If commercial software is used to do the back up, make sure the backup program itself runs as a system process. This will ensure that the program will be able to access all the directories and subdirectories.

Restoring NICI

- 1 If NICI is not installed, restore all the registry information first.
or
If NICI is installed, remove NICI and overwrite the registry information from the backup store.
- 2 Restore the files and directories within `HKEY_LOCAL_MACHINE\SOFTWARE\Novell\NICI\ConfigDirectory` as selected by the operator.

As in the UNIX case, we recommend restoring all the files as a group. But a knowledgeable operator can choose to restore individual entries. This can be done only if the `nicifk` and `xmgrcfg.wks` files did not change from the one on the backup store. In that case, be sure to adjust the access rights based on the new owner of the user configuration directories. The individual directories are named after the owner but access rights are controlled by the SID. Just because a subdirectory is named BOB does not automatically mean that the current user BOB is the correct owner of the information being restored.

Special Case for Windows

It is possible to configure the registry value `HKEY_LOCAL_MACHINE\SOFTWARE\Novell\NICI\UserDirectoryRoot` to indicate that the user configuration files be placed in the user's personal configuration directory. In that case, be prepared to back up and restore the user information independently as part of normal backup and restore operations. If NICI has been configured in that manner, you should know about it and be prepared to do individual backups.

This special case for the Windows user directory is enabled by creating the registry value `EnableUserProfileDirectory` rather than just pointing the directory path there. If Windows is configured to automatically create and delete user accounts, the directory might be automatically deleted when the user profile directory is enabled. In that case, backup and restore is only necessary for those specific users who are permanent. The default path will be user the `Application Data\Novell\Nici` directory branch of the user's directory in Documents and Settings.

15

Maintaining Novell eDirectory

For Novell® eDirectory™ to perform optimally, you need to maintain the directory through routine health check procedures and upgrading or replacing hardware when necessary.

This chapter covers the following maintenance topics:

Performance

- ◆ [“Improving eDirectory Performance” on page 427](#)
- ◆ [“Improving eDirectory Performance on Linux, Solaris, AIX, and HP-UX Systems” on page 435](#)

Health Checks

- ◆ [“Keeping eDirectory Healthy” on page 444](#)
- ◆ [“Resources for Monitoring” on page 447](#)

Hardware Replacements

- ◆ [“Upgrading Hardware or Replacing a Server” on page 447](#)

eDirectory Recovery

- ◆ [“Restoring eDirectory after a Hardware Failure” on page 454](#)

Improving eDirectory Performance

The most significant setting that affects eDirectory performance is the cache. In earlier versions of NDS®, you could specify a block cache limit to regulate the amount of memory that the directory used for the cache. The default was 8 MB RAM for cache.

With eDirectory 8.5 or later, you can specify a block cache limit and an entry cache limit. The block cache, available in earlier versions of NDS, caches only physical blocks from the database. The entry cache, a feature introduced in eDirectory 8.5, caches logical entries from the database. The caching of entries reduces the processing time required to instantiate entries in memory from the block cache.

Although there is some redundancy between the two caches, each cache is designed to boost performance for different operations. Block cache is most useful for update operations. Entry cache is most useful for operations that browse the eDirectory tree by reading through entries, such as name resolution.

Both block and entry caches are useful in improving query performance. Block cache speeds up index searching. Entry cache speeds up the retrieval of entries referenced from an index.

The defaults for eDirectory 8.7.3 are listed below:

- ◆ If the server you are installing eDirectory on does not have a replica, the default is a hard memory limit of 16 MB, with 8 MB for block cache and 8 MB for entry cache.
For more information, refer to [“Understanding the Hard Memory Limit” on page 429](#).
- ◆ If the server contains a replica, the default is a dynamically adjusting limit of 51% of available memory, with a minimum threshold of 8 MB and a maximum threshold of keeping 24 MB available.
For more information, refer to [“Understanding the Dynamically Adjusting Limit” on page 428](#).

Distributing Memory between Entry and Block Caches

With an entry cache and a block cache, the total available memory for caching is shared between the two caches. The default is an equal division. To maintain the amount of block cache available in earlier versions of NDS 8, you need to double the total cache size for eDirectory. If you use the cache to boost LDIF-import performance, for example, you can either double the total cache size or change the default cache settings. To change the default cache settings, refer to [“Configuring Dynamically Adjusting and Hard Memory Limits” on page 429](#).

The more blocks and entries that can be cached, the better the overall performance will be. The ideal is to cache the entire database in both the entry and block caches, although this is not possible for extremely large databases. Generally, you should try to get as close to a 1:1 ratio of block cache to DIB Set as possible. For entry cache, you should try to get close to a 1:2 or 1:4 ratio. For the best performance, exceed these ratios.

Using the Default Cache Settings

eDirectory provides two methods for controlling cache memory consumption: a dynamically adjusting limit and a hard memory limit. You can use either method, but you cannot use them at the same time because they are mutually exclusive. The last method used always replaces any prior settings.

Understanding the Dynamically Adjusting Limit

The dynamically adjusting limit causes eDirectory to periodically adjust its memory consumption in response to the ebb and flow of memory consumption by other processes. You specify the limit as a percentage of available physical memory. Using this percentage, eDirectory recalculates a new memory limit at fixed intervals. The new memory limit is the percentage of physical memory available at the time.

Along with the percentage, you can set a maximum and minimum threshold. The threshold is the number of bytes that eDirectory will adjust to. It can be set as either the number of bytes to use or the number of bytes to leave available. The minimum threshold default is 16 MB. The maximum threshold default is 4 GB.

If the minimum and maximum threshold limits are not compatible, the minimum threshold limit is followed. For example, you could specify the following settings:

Minimum threshold:	8 MB
Percentage of available physical memory to use:	75
Maximum threshold:	Keep 10 MB available

When eDirectory adjusts its cache limit, there is 16 MB of available physical memory. eDirectory calculates a new limit of 12 MB. eDirectory checks to see whether the new limit falls within the range of minimum and maximum thresholds. In this example, the maximum threshold requires that 10 MB must remain available, so eDirectory sets the limit to 6 MB. However, the minimum threshold is 8 MB, so eDirectory sets the final limit to 8 MB.

With the dynamically adjusting limit, you also specify the interval length. The default interval is 15 seconds. The shorter the interval, the more the memory consumption is based on current conditions. However, shorter intervals are not necessarily better because the percentage recalculation will create more memory allocation and freeing.

Understanding the Hard Memory Limit

The hard memory limit is the method that earlier versions of eDirectory use to regulate memory consumption. You set a hard memory limit in one of the following ways:

- ◆ Fixed number of bytes
- ◆ Percentage of physical memory

The percentage of physical memory at the interval becomes a fixed number of bytes.

- ◆ Percentage of available physical memory

The percentage of available physical memory at the interval becomes a fixed number of bytes.

Cleaning Up the Cache


NDS 8 creates multiple versions of blocks and entries in its cache for transaction integrity purposes. Earlier versions of NDS 8 did not remove these blocks and entries when they were no longer needed. In eDirectory 8.7.3, a background process periodically browses the cache and cleans out older versions. This helps minimize cache memory consumption. The default browsing interval is 15 seconds.

Configuring Dynamically Adjusting and Hard Memory Limits

You can configure dynamically adjusting and hard memory limits in either of the following methods:

- ◆ [“Using Novell iMonitor” on page 429](#)
- ◆ [“Using the _ndsdb.ini File” on page 431](#)

Using Novell iMonitor

- 1** Click Agent Configuration .
- 2** Click Database Cache, then view the following information:

Database Cache Information	Description
Maximum Size	The maximum size (in KB) that the specified cache is allowed to grow to.
Current Size	The current size (in KB) of the specified cache.
Items Cached	The number of items in the specified cache.

Database Cache Information	Description
Old Versions Cached	The number of old versions in the specified cache. Old versions of cache items are kept to maintain the consistency of read transactions in the database. In other words, if one thread is in a read transaction and another is in a write transaction, old versions of blocks modified by the writer are maintained on behalf of the reader. This is done so that the reader's results are guaranteed to produce a consistent view during the life of its transaction even though modifications are taking place during that time.
Old Versions Size	The size (in KB) of the old version items cached.
Hits	The number of times an item was successfully accessed from the specified cache.
Hit Looks	The number of items looked at in the cache before an item was successfully accessed from the specified cache. The hit-look-to-hit ratio is a measure of cache lookup efficiency. Normally, the ratio should be close to 1:1.
Faults	The number of times an item was not found in the specified cache and had to be obtained in a lower level cache or from the disk.
Fault Looks	The number of items looked at in the cache before it was determined that the desired item was not in the specified cache. The fault-look-to-fault ratio is a measure of cache lookup efficiency. Normally, the ratio should be close to 1:1.

3 Choose from the following options:

Option	Description
Dynamic Adjust	Allows the eDirectory database to dynamically adjust the amount of system memory to be used for the cache based on the amount it thinks it needs and the parameters specified below.
Cache Adjust Percentage	The percentage of available memory allowed to be used for the record and block caches combined.
Cache Size Constraints	While dynamically adjusting, follow the specified constraints. Namely, use no less than the specified amount of memory for the cache and no more than the total amount of available memory minus the specified amount.
Hard Limit	The exact amount of system memory to be use for the cache.
Cache Maximum Size	The size (in KB) of the record and block caches combined.

Option	Description
Block Cache Percentage	The percentage of the system memory available for caching that should be allocated to the block cache. The remaining percentage will be allocated to the record cache.
Cache Adjust Interval	This interval applies only when Dynamic Adjust is set. It controls how often the cache size is adjusted, based on the specified percentage and constraints.
Cache Cleanup Interval	Controls how often unused old versions are removed from the cache.
Cache Settings Permanent	When this option is selected, any changes submitted through iMonitor will be made permanent, overwriting any previously saved settings or system defaults.

4 Click Submit.

Using the `_ndsdb.ini` File

1 Open `_ndsdb.ini` in a text editor.

In NetWare®, this file is in `sys:\netware`. In Windows NT and Windows 2000, this file is generally in `\Novell\NDS\DIBfiles`.

2 Add the applicable syntax to the file:

Command	Variable Explanation	Definition
<code>cache=cache_bytes</code>	Fixed number of bytes you want used.	Sets a hard memory limit. For example, to set a hard limit of 8 MB, enter <code>cache=8000000</code>

Command	Variable Explanation	Definition
<code>cache=cache_options</code>	<p>Multiple options can be specified in any order, separated by commas.</p> <ul style="list-style-type: none"> ♦ DYN Sets a dynamically adjusting limit. ♦ HARD Sets a hard memory limit. ♦ %:percentage Percentage of available or physical memory to use. ♦ AVAIL or TOTAL Percentage of available or total physical memory for hard memory limit only. ♦ MIN:number_of_bytes Minimum number of bytes. ♦ MAX:number_of_bytes Maximum number of bytes. ♦ LEAVE:number_of_bytes Minimum number of bytes to leave. 	<p>Sets a hard memory or dynamically adjusting limit.</p> <p>For example, to set a dynamically adjusting limit of 75% of available memory and a minimum of 16 MB, enter</p> <p><code>cache=DYN, %:75,MIN:16000000</code></p> <p>Or to set a hard limit of 75% of total physical memory and a minimum of 16 MB, enter</p> <p><code>cache=HARD,%:75,MIN: 16000000</code></p>

- 3** (Optional) To specify the dynamic adjusting limit interval, add the following line:

`cacheadjustinterval=number_of_seconds`

- 4** (Optional) To specify the interval for cleaning up older versions of entries and blocks, add the following line:

`cachecleanupinterval=number_of_seconds`

- 5** (Optional) To change the percentage split between block cache and entry cache, add the following line:

`blockcachepersent=percent`

The variable *percent* should be between 0 and 100. The percentage you specify is the percentage of cache memory used for the block cache. The remaining percentage is used for the entry cache. We do not recommend setting the percentage to 0.

- 6** Restart the eDirectory server for the changes to take effect.

Configuring Limits Using DSTrace

If you are using eDirectory for NetWare, you can configure the dynamically adjusting and hard memory limits in DSTrace. You do not need to restart the server for the changes to take effect.

- 1** (Optional) To set a fixed hard limit, enter the following at the server console:

`SET DSTTRACE=!MBamount_of_RAM_to_use_in_bytes`

For example, to set a hard limit of 8 MB, you would enter


```
SET DSTRACE=!MB8388608
```

- 2** (Optional) To set a calculated hard limit, enter the following at the server console. Include only the options you want to specify.

```
SET DSTRACE=!MHARD,AVAIL OR
TOTAL,%:percent,MIN:number_of_bytes,MAX:number_of_
bytes,LEAVE:number_of_bytes_to_leave,NOSAVE
```

For example, to set a hard limit of 75% of total physical memory and minimum of 16 MB, and to specify not to save these options to the startup file, you would enter

```
SET DSTRACE=!MHARD,%:75,MIN:16777216,NOSAVE
```

- 3** (Optional) To set a dynamically adjusting limit, enter the following at the server console:

```
SET DSTRACE=!MDYN,%:percent,MIN:number_of_bytes,MAX:
number_of_bytes,LEAVE:number_of_bytes_to_leave,
NOSAVE
```

For example, to set a dynamic limit of 75% of available memory and a minimum of 8 MB, you would enter

```
SET DSTRACE=!MDYN,%:75,MIN:8388608
```

Tuning LDAP for eDirectory

For information on basic LDAP server hardware and software configuration, tuning parameters, and tips on directory organization, see [How to Configure and Optimize eDirectory LDAP Servers \(http://developer.novell.com/research/appnotes/2000/septembe/04/a000904.htm\)](http://developer.novell.com/research/appnotes/2000/septembe/04/a000904.htm).

Managing the Memory

eDirectory uses memory for the database cache and for directory usage. These are separate allocated memory pools. The directory engine uses memory from available memory pools in the operating system as needed. The database uses a cache pool that is defined by parameters detailed below. Usually, the more database cache given to eDirectory, the better the performance. However, because eDirectory uses available system memory for its buffers, if clients are performing queries that require large data sets to be returned, the size of the database cache might need to be decreased to have enough system memory for the directory to handle building the query responses.

The database engine uses the database cache to hold the most recently accessed blocks. This cache is initially defined with a fixed size of 16 MB. The size of this cache can be changed from the command line in shipping versions of eDirectory. The following example command will set the eDirectory database cache to 80 million bytes:

```
set dstrace=!mb 80000000
```

You can also define a file named `_ndsdb.ini` in the `sys:_netware` directory on a NetWare server, or in the directory containing the eDirectory database files on the Windows (normally `install directory\nds\dbfiles`) and UNIX environments (normally `\var\nds\dib`). This text file simply needs to contain a line such as the following:

```
cache=80000000
```

Don't add any white space around the equals (=) sign

The cache in eDirectory 8.7.3 can be initialized with a hard limit just as with earlier versions. In addition, the upper and lower limits can be set either as hard numbers or as a percentage of available memory. Dynamic allocation control parameters allow the cache size to grow or shrink

depending on use. If the proper configuration parameters are set, the database cache dynamically grows or shrinks based on other system resource needs.

Editing the `_ndsdb.ini` file can manually control database memory usage. The format for INI file commands is given below:

```
cache=cacheBytes # Set a hard memory limit
```

Alternative formats are shown in the following table:

Command	Description
<code>cache=<i>cache_options</i></code>	Sets a hard limit or dynamically adjusting limit. Multiple cache options can be specified in any order, separated by commas. All are optional. They are as follows:
DYN or HARD	Dynamic or hard limit.
AVAIL or TOTAL	These only apply if a hard limit was chosen. Omit these options for a dynamic limit.
<code>%:<i>percentage</i></code>	The percentage of available or total physical memory.
<code>MIN:<i>bytes</i></code>	The minimum number of bytes.
<code>MAX:<i>bytes</i></code>	The maximum number of bytes.
<code>LEAVE:<i>bytes</i></code>	The minimum number of bytes to leave for the OS.
<code>blockcachepersent=<i>percentage</i></code>	Splits the cache between the block and record cache.

If a hard limit is specified and the administrator wants to define the database cache to use a percentage of the memory, the administrator can select between a percentage of total memory or a percentage of available memory. Dynamic limits always refer to a percentage of available memory. The following command examples are all valid in the `_ndsdb.ini` file.

The following is an example dynamic limit of 75% available memory, a minimum of 16 million bytes, and 32 million bytes for the OS:

```
cache=DYN,%,75,MIN:16000000, LEAVE 32000000
```

The following is an example hard limit of 75% total physical memory, a minimum of 18 million bytes, and a maximum of 512 million bytes:

```
cache=HARD, TOTAL,%,75,MIN:18000000, MAX 512000000
```

The following is an example old style hard limit of 8 million bytes:

```
cache=8000000
```

The database cache is divided between block cache and record cache. Block cache holds data and index blocks that mirror the storage on the disk. Record cache holds in-memory representations of directory objects and attributes. If updating or adding to the directory, use the block cache setting. If performing mostly reads, use the record cache. It is possible to cause a thrashing condition in both caches if performing numerous sequential updates without allocating cache size properly. Unless specifically changed, the cache is allocated to be 50% block cache and 50% record cache. The `blockcachepersent` option can be included in the `_ndsdb.ini` file to specify the percentage of cache allocated to caching data and index blocks. (The default is 50%.) The remaining cache is used for entries.

For example, to designate 60% block cache and 40% record cache, enter the following:

```
blockcachepersent=60
```

Do not select 100% of the cache for either block or record cache and starve the other cache type. In general, do not allocate more than 75% of your cache memory to one or the other type.

Database cache settings can also be controlled using Novell iMonitor.

Although the cache size is dynamic depending on the amount of memory available, the DSTRACE command can still be used for custom environments.

Improving eDirectory Performance on Linux, Solaris, AIX, and HP-UX Systems

The following sections provide information about how you can improve the performance of eDirectory on UNIX systems:

- ♦ [“Fine-Tuning the eDirectory Server” on page 435](#)
- ♦ [“Optimizing eDirectory Cache” on page 436](#)
- ♦ [“Optimizing Bulkload Data” on page 439](#)
- ♦ [“Tuning the Solaris OS for Novell eDirectory” on page 440](#)

Fine-Tuning the eDirectory Server

Novell eDirectory on Linux and Solaris uses a dynamically adjusted thread pool to service client requests. The thread pool is self-adjusting and delivers optimum performance in most cases. However, you can avoid the delay caused by starting up threads when there is a sudden load on the server by setting the following parameters in the `/etc/nds.conf` file.

Parameter	Description and Recommended Settings
n4u.server.idle-threads	<p>Minimum number of threads (regardless of activity)</p> <p>The value of this parameter should be based on the average client load, in order to minimize the time required to produce new threads during normal client activity.</p>
n4u.server.max-threads	<p>Maximum number of threads</p> <p>The value of this parameter should be based on the maximum number of simultaneous clients that need to be serviced, along with the following recommendations:</p> <ul style="list-style-type: none"> ♦ eDirectory requires a minimum of 16 threads ♦ One thread for every 255 LDAP connection (Monitor Thread) ♦ One thread for every four concurrent clients (Worker Thread)
n4u.server.start-threads	<p>Number of threads that start when eDirectory starts</p> <p>The value of this parameter should be based on the average client load, in order to minimize the time required to produce new threads during normal client activity.</p>

Optimizing eDirectory Cache

Novell eDirectory uses persistent caching so that changes being made to a server are held in a vector. If the server crashes in the middle of changes, eDirectory will load faster and synchronize the changes in seconds when the server is brought back up. Novell eDirectory uses a rollback model with a log file to roll forward transactions in the event of a system failure.

eDirectory settings begin with 16MB of cache, 50% of which is allocated to block caching and the other 50% is allocated to record cache. After 15 minutes, eDirectory will modify its cache thresholds to initialize up to 51% of the available free memory for the cache, leaving at least 24 MB for the OS. This algorithm is used only if the host OS supports the call that enables you to determine the amount of free memory available.

You can optimize your eDirectory cache in the following ways:

- ◆ [“Using a Fixed Amount of RAM for UNIX Systems” on page 436](#)
- ◆ [“Setting Cache Parameters” on page 438](#)
- ◆ [“Optimizing Bulkload Data” on page 439](#)
- ◆ [“Optimizing LBURP Transaction Size” on page 439](#)

Using a Fixed Amount of RAM for UNIX Systems

Although the above algorithm works well for Windows and NetWare, it does not work as well for UNIX systems. On UNIX systems, the free available memory reported by the OS will be less than other operating systems because of the way the UNIX OS uses free memory for internal caching of file system blocks, frequently run programs, libraries, etc. In addition to this memory allocation, libraries on UNIX normally do not return the freed memory back to the OS.

For these reasons, we recommend allocating a fixed amount of RAM to the cache.

Fix the amount of RAM for UNIX systems by doing one of the following:

- ◆ [“Manually Creating a .ini File” on page 436](#)
- ◆ [“Using Novell iMonitor” on page 437](#)

Manually Creating a .ini File

- 1** Create a file called `_ndsdb.ini` in the same directory that the eDirectory database files (DIB set) are located (usually in `/var/nds/dib`).
- 2** Add the following parameters listed in to the `_ndsdb.ini` file:

Parameter	Description
<code>blockcachepcent=50</code>	Sets the percentage of cache that is allocated to caching database blocks.
<code>cacheadjustinterval=15</code>	Sets the time (in seconds) in which eDirectory will evaluate its utilization of free memory and adjust the overall cache size.
<code>cachecleanupinterval=15</code>	Sets the time (in seconds) in which eDirectory will write dirty cache blocks to disk.
<code>cache=16777216</code>	Sets the hard limit (in bytes).

Using Novell iMonitor

1 Click Agent Configuration .

2 Click Database Cache, then view the following information:

Database Cache Information	Description
Maximum Size	The maximum size (in KB) that the specified cache is allowed to grow to.
Current Size	The current size (in KB) of the specified cache.
Items Cached	The number of items in the specified cache.
Old Versions Cached	The number of old versions in the specified cache. Old versions of cache items are kept to maintain the consistency of read transactions in the database. In other words, if one thread is in a read transaction and another is in a write transaction, old versions of blocks modified by the writer are maintained on behalf of the reader. This is done so that the reader's results are guaranteed to produce a consistent view during the life of its transaction even though modifications are taking place during that time.
Old Versions Size	The size (in KB) of the old version items cached.
Hits	The number of times an item was successfully accessed from the specified cache.
Hit Looks	The number of items looked at in the cache before an item was successfully accessed from the specified cache. The hit-look-to-hit ratio is a measure of cache lookup efficiency. Normally, the ratio should be close to 1:1.
Faults	The number of times an item was not found in the specified cache and had to be obtained in a lower level cache or from the disk.
Fault Looks	The number of items looked at in the cache before it was determined that the desired item was not in the specified cache. The fault-look-to-fault ratio is a measure of cache lookup efficiency. Normally, the ratio should be close to 1:1.

3 Choose from the following options:

Option	Description
Dynamic Adjust	Allows the eDirectory database to dynamically adjust the amount of system memory to be used for the cache based on the amount it thinks it needs and the parameters specified below.
Cache Adjust Percentage	The percentage of available memory allowed to be used for the record and block caches combined.

Option	Description
Cache Size Constraints	While dynamically adjusting, follow the specified constraints. Namely, use no less than the specified amount of memory for the cache and no more than the total amount of available memory minus the specified amount.
Hard Limit	The exact amount of system memory to be use for the cache.
Cache Maximum Size	The size (in KB) of the record and block caches combined.
Block Cache Percentage	The percentage of the system memory available for caching that should be allocated to the block cache. The remaining percentage will be allocated to the record cache.
Cache Adjust Interval	This interval applies only when Dynamic Adjust is set. It controls how often the cache size is adjusted, based on the specified percentage and constraints.
Cache Cleanup Interval	Controls how often unused old versions are removed from the cache.
Cache Settings Permanent	When this option is selected, any changes submitted through iMonitor will be made permanent, overwriting any previously saved settings or system defaults.

4 Click Submit.

Setting Cache Parameters

By default, eDirectory uses dynamic cache. If you have sufficient RAM to increase the eDirectory cache size, you can increase the performance of eDirectory considerably for large databases by allocating more RAM to the eDirectory cache.

The parameters that are listed in the following table can be adjusted to enhance your eDirectory performance:

eDirectory Cache Parameter	Description
<code>blockcachepersent=<i>value</i></code>	Sets the percentage of cache that is allocated to caching database blocks. The default is 50.
<code>cachecleanupinterval=<i>value</i></code>	Sets the time (in seconds) in which eDirectory will write dirty cache blocks to disk. The default is 15.
<code>cacheadjustinterval=<i>value</i></code>	Sets the time (in seconds) in which eDirectory will evaluate its utilization of free memory and adjust the overall cache size. The default is 15.
<code>cache=<i>value</i></code>	Sets a hard limit (in bytes) of memory that eDirectory can use for caching.
<code>cache=leave:<i>value</i></code>	Specifies the minimum number of bytes to leave.
<code>min:<i>value</i></code>	Specifies the minimum cache size in bytes.
<code>max:<i>value</i></code>	Specifies the maximum cache size in bytes.

According to the algorithm, the default setting for Novell eDirectory is the following:

```
cache=dyn,%:51,min:16777216,max:0,leave:0
```

This indicates the following:

- ◆ The minimum cache size is 16 MB.
- ◆ There is no maximum limit.
- ◆ Dynamically, up to 51% of available memory will be used.
- ◆ 24 MB should be left for the OS.

eDirectory operates with a hard limit of 16 MB, so that all applications are started and the system is stabilized.

You can also configure eDirectory to use a percentage of the total memory. To do so, specify the cache as shown below:

```
cache=hard,total,%:percentage_of_total_memory_in_bytes
```

Optimizing Bulkload Data

Bulkload performance using the Import/Convert/Export (ICE) utility can be affected by a number of items. The most common performance hits come from poor Disk I/O management and the allocation of insufficient memory for the Novell eDirectory cache.

If eDirectory is essentially the only application, you can set the eDirectory cache up to 80% of the total memory. All allocated cache will eventually be used. eDirectory performance on highly volatile data is improved with more cache.

IMPORTANT: You should avoid setting the cache memory size above 40% of the total memory if the server is hosting services or applications other than eDirectory. The smallest tested cache size is 0 and the largest is 3 GB. Determining the proper cache size depends on the memory needs of other processes running on the same server, and on the amount of disk cache required. You should test a variety of cache sizes to find a good balance.

To optimize the bulkload performance, allocate a higher percentage of the eDirectory cache for block cache. We recommend setting a value of 80% for block cache. This can be reset after the operation is completed.

Using iMonitor is the quickest way to modify the blockcachepcentage parameter. To do so, follow the instructions in [“Using Novell iMonitor” on page 437](#).

Optimizing LBURP Transaction Size

The LBURP transaction size sets the number of records that will be sent from ICE to the LDAP server during a single transaction. Increasing this value can improve bulkload performance, assuming that you have adequate memory and that the increase does not cause I/O contention.

The default transaction size is 25, which is appropriate for small LDIF files (fewer than 100,000 operations) but not for a large number of records. The LBURP transaction size can be set anywhere between 1 and 1000.

Modifying the Transaction Size

To modify the transaction size, modify the required value for the `n4u.ldap.lburp.transize` parameter in `/etc/nds.conf`.

In ideal scenarios, a higher transaction size ensures faster performance. However, the transaction size must not be set to arbitrarily high values for the following reasons:

- ◆ A larger transaction size requires the server to allocate more memory to process the transaction. If the system is running low on memory, this can cause a slowdown due to swapping.
- ◆ Make sure that the LDIF file is free of errors and that any entries already existing in eDirectory have been commented out. Even if a single error exists in the transaction (including cases where the object to be added already exists in the directory), eDirectory will ignore the LBURP transaction setting and perform a commit after each operation to ensure data integrity.

See [“Debugging LDIF Files” on page 491](#) for more information.

- ◆ LBURP optimization works only for leaf objects. If the transaction contains both a container and its subordinate objects, eDirectory will treat this as an error. To avoid this, load container objects first from a separate LDIF file or enable the use of forward references.

See [“Enabling Forward References” on page 491](#) for more information.

Tuning the Solaris OS for Novell eDirectory

The following sections provide information about how to tune the Solaris kernel, network, and file system:

IMPORTANT: Before you begin, make sure that you have applied the recommended patches to the Solaris OS. For more information, see [“Installing or Upgrading Novell eDirectory on Solaris”](#) in the *Novell eDirectory 8.7.3 Installation Guide*.

- ◆ [“Tuning the Solaris Kernel” on page 440](#)
- ◆ [“Tuning the Solaris Network” on page 441](#)
- ◆ [“Fine-Tuning the Solaris File System” on page 441](#)

Tuning the Solaris Kernel

To optimize the performance of eDirectory on Solaris, set the following kernel variables in the `etc/system`:

Parameter	Description
set maxphys=1048576	Maximum number of bytes that can be transferred per SCSI transaction.
set md_maxphys=1048576	Maximum number of bytes that can be transferred per SCSI transaction if you are using disksuite, vol_maxio, or vxvm.
set ufs:ufs_LW=1/128_of_available_memory	Barrier for the number of outstanding bytes on a single file below which the condition variable on which other sleeping processes are toggled.
set ufs:ufs_HW=1/64_of_available_memory	Number of bytes outstanding on a single fail barrier value.

Parameter	Description
<code>ctcp:tcp_conn_hash_size=8192</code>	Number of connection hash entries that are allocated to quickly locate the kernel data structures that are associated with TCP connection. (This can be increased to 262144, based on the number of LDAP clients.)

Tuning the Solaris Network

You can enhance LDAP search performance using the Solaris `ndd` command. The following command syntax allows you to analyze and modify tunable parameters that affect networking operation and behavior:

```
ndd -set /dev/tcp variable_name variable_value
```

The recommended values for the variables are listed in the following table:

Parameter	Description
<code>tcp_conn_req_max_q: 1024</code>	The “q” stands for queue, which is the completed socket holding pen where sockets remain until the application issues accept.
<code>tcp_time_wait_interval: 60000</code>	Sets (in this case lowers) the time wait interval.
<code>tcp_xmit_hiwat: 64000</code> <code>tcp_xmit_lowat: 64000</code>	Adjusts the minimum and maximum TCP send window size.
<code>tcp_slow_start_initial: 2</code>	Adjusts the number of first transmission packets from 1 to 2.

Fine-Tuning the Solaris File System

Novell eDirectory performance on Solaris can be improved if the Solaris file system is adequately tuned, especially for bulk loading data into the directory. File system tuning for eDirectory is similar to tuning for a database. See the [Sunworld* Web site \(http://www.sunworld.com/sunworldonline\)](http://www.sunworld.com/sunworldonline) for more information on the Solaris file system.

Advanced Referral Costing

Server applications often communicate with other servers via a built-in client (Dclient), because a single server doesn't contain all the necessary eDirectory data for an application to operate. An example is NLDAP, when it is configured to chain requests.

When a server application requests data that the local server does not hold, the server locates another server that contains the requested data, and subsequently retrieves the data for the client. This process is called “tree walking”. It naturally takes longer for a server to fulfill a request through tree walking. Although best practice guidelines for eDirectory tree design minimize the need for tree walking, it is still sometimes necessary.

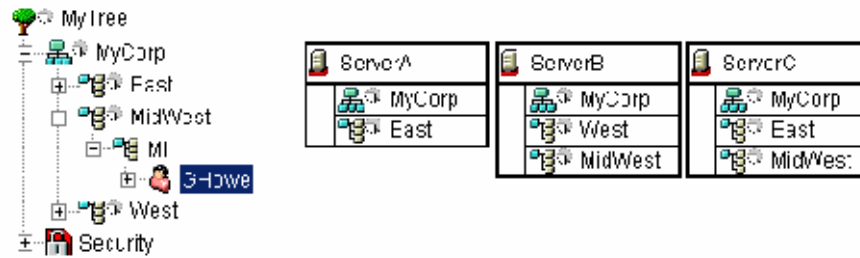
Figure 35 Advanced Referral Costing

Figure 35 illustrates an LDAP subtree search to Server A for `cn=GHowe`, starting at `O=MyCorp`. However, the `cn=GHowe` object is located in the `ou=MidWest` partition, which is not represented on Server A.

To locate a server that holds the data needed to fulfill the client request, Server A must either get the data from Server B or Server C. To do this, Server A must send the request to either Server B or C. Server A happens to choose Server B; the server chosen is not random, but the process is unpredictable. Server B is available on the network and accepts the request, but is unable to complete the request quickly, resulting in Server A waiting for Server B even though Server C could also provide the required data. Until Server B either fulfills the request or is no longer available on the network, the request from Server A must wait.

Improving Server-to-Server Connection

Advanced Referral Costing (ARC) provides an improved server-to-server connection. ARC helps eDirectory servers avoid servers that are responding at a transport level, and are slow or unresponsive to eDirectory requests. This feature must be enabled using `DSTrace`, `NDSTrace`, or `NDS iMonitor`.

Whenever a server must perform a tree walk, the local server maintains a quality-of-service table. Every time a Resolve Name request is sent to a remote server, the resulting responsiveness is used to update the table. As a server's response times get slower, its resulting cost increases and is recorded in the table. When it is necessary to select a connection for making a request, the table is checked and the fastest server is chosen. There are some services that exist only on specific servers in a tree, so slower servers aren't placed in a bad address cache, but their cost is adjusted according to their responsiveness.

ARC does not prevent all requests from being backed-up behind a slow server, but it does significantly limit the impact that a distressed server has on tree walking requests. ARC has been enhanced to adjust quickly to even minor slowdowns in response times, so server requests balance themselves across the fastest responding servers in the enterprise.

In addition, responsiveness is measured on a per-address level. The original costing algorithm evaluated each individual connection; however, the enhanced method evaluates connections based on the responsiveness of all connections to any given address. This allows one connection to benefit from the information obtained from another connection to the same address.

ARC can be deployed on a server-by-server basis, with each server benefiting when communicating with a slow server. The servers that are running ARC can then route new requests to other servers. For example, you can deploy ARC to key servers that are expected to make remote requests to other servers, such as LDAP servers that need to chain requests.

As requests are routed to the quickest servers, the net result is a load balancing effect that distributes the load across multiple servers. Although it is not true load balancing, it does have many benefits.

Advantages of Referral Costing

- ◆ It times/routes most Resolve Name requests to remote servers as they are made.
- ◆ It averages the Resolve Name request times in milliseconds on each address. This allows ARC to be more granular and adjust the cost of the referral more aggressively. It is also able to quickly detect a slow server, because timing is tracked in milliseconds instead of seconds.
- ◆ It tracks outstanding requests so quickly determine if a request is taking too long. It does not have to wait for the request to complete in order to know that the server is taking a long time .
- ◆ It tracks response time on a per-address basis. It is normal for a server to have numerous connections to the same address. By tracking per address instead of per connection, one connection can benefit from statistics gathered from the other connections.

NOTE: To account for LDAP requests, ARC also takes into account responsiveness of private connections.

Enabling Advanced Referral Costing

- DSTrace (Currently NetWare only)

Use DSTrace command to enable ARC on NetWare.

- NDSTrace (All Unix platforms)

Use NDSTrace command to enable ARC on all UNIX platforms.

Table 1 Enabling ARC on Unix Platforms

set NDSTRACE =!ARC	Displays the gv_ResolveTimesTable for debugging.
set NDSTRACE = !ARC0	Disables Advanced Referral Costing.
set NDSTRACE = !ARC1	Enables Advanced Referral Costing.
set NDSTRACE = !ARC2	Enables Advanced Referral Costing in debug mode and displays the resulting costs of each referral on the Resolve Name DSTrace flag anytime a costing decision is made.

NOTE: Currently, ARC is not supported on the Windows platform. However, if you you want ARC on the Windows, please contact your [Novell Support Provider \(http://www.novell.com/documentation/nwec/nwec_enu/nwec_novell_support_provider.html\)](http://www.novell.com/documentation/nwec/nwec_enu/nwec_novell_support_provider.html).

Monitoring Advanced Referral Costing

The new costing is very dynamic and changes very frequently. In order to watch it work, you can set the Advanced Referral Costing parameter to Debug mode.

NOTE: Ensure you reset ARC to non debug mode, i.e. set NDSTRACE = !ARC1 when you have finished monitoring. Overhead printing costs are not desirable when you don't need it.

In the DSTrace or NDSTrace, you now see the individual referral costs displayed if Advanced Referral Costing and +RSLV are turned on.

Sorted results from DCAdjustCostAndSort follow:

```
137.65.10.3 cost of 217
137.65.10.9 cost of 222
137.65.10.10 cost of 400
```

The numbers change quickly if a remote server is slow or overloaded. The ExRef server's costing adjusts dynamically every second, so to watch costs over time you should the trace to a log file.

Printing the ResolveTimes Table

Another way to observe Advanced Referral Costing in action is to print the ResolveTimes table.

To do this, use the following commands:

- ◆ set DSTRACE = +DBG
- ◆ set DSTRACE = !ARC

This prints the Resolve Times table and the current stored information for each server. It shows the transport address, the milliseconds since the address was last used, the last cost that was used in a referral decision, and the number of outstanding requests for that address.

A high number of outstanding requests is not necessarily a problem. It might simply mean that that server is used frequently.

Keeping eDirectory Healthy

The health of directory services is vital to any organization. Regular health checks using Novell iMonitor will keep your directory running smoothly and will make upgrades and troubleshooting much easier.

When to Perform Health Checks

In general, if your network doesn't change often (servers and partitions are added only every couple of months and only simple changes are made frequently), perform health checks once a month.

If your network is more dynamic (partitions or servers are added weekly or your organization is reorganizing), perform health checks weekly.

Adjust the frequency of health checks as your environment changes. Factors that influence the timing of your health checks include the following:

- ◆ Number of partitions and replicas
- ◆ Stability of replica holding servers
- ◆ Amount of information in an eDirectory partition
- ◆ Object size and complexity
- ◆ Number of errors in previous DSRepairs

When you perform a health check, iMonitor gathers information from all servers based on given rights. Be aware that running health check reports might generate network traffic and use disk space.

Health Check Overview

A complete health check includes checking the following:

- ◆ eDirectory version

Running different versions of NDS or eDirectory on the same version of NetWare can cause synchronization problems. If your version of NDS or eDirectory is outdated, download the latest software patch from [Novell Directory Services Patches and Files \(http://support.novell.com/filefinder/5069/index.html\)](http://support.novell.com/filefinder/5069/index.html).

- ◆ Time synchronization

All eDirectory servers must maintain accurate time. Time stamps are assigned to each object and property and they ensure the correct order for object and property updates. Using time stamps, eDirectory determines which replicas need to be synchronized.

- ◆ Synchronization tolerances

Time periods since a server has synched with inbound and outbound data changes, how much data is outstanding, etc.

- ◆ Background processes

Processes that perform a variety of tasks including replication of changes and maintenance of system information.

- ◆ External references
- ◆ Obituaries
- ◆ eDirectory Schema

Step-by-step instructions for completing these checks are given in the following section, [“Checking eDirectory Health Using iMonitor.”](#)

Checking eDirectory Health Using iMonitor


Depending on your preference, you can perform an eDirectory server health check by using either of two methods in iMonitor:

- ◆ [Using the Navigator Frame](#)
- ◆ [Using the Assistant Frame](#)

Using the Navigator Frame

- 1** Access iMonitor.

See [“Accessing iMonitor” on page 165](#).

- 2** In the Navigator frame, click the Reports icon .

- 3** In the Assistant frame, click the Report Config link.

A Runnable Report List appears in the Data frame.

- 4** Click the Configure Report icon  for your desired server information.

A Server Information Report appears in the Data frame. You will use this report to select the desired options for your report.

- 5** Check the Health Sub-Report check box.

- 6 To run the report at specified intervals, select the desired options in the Schedule Report section of the Data frame.

IMPORTANT: If you run a scheduled report, it will run as public and might not be able to gather as much information as it would if you ran it as an authenticated user.

- 7 Click Run Report to process the report.

Using the Assistant Frame

- 1 Access iMonitor.

See “Accessing iMonitor” on page 165.

- 2 In the Assistant frame, click Agent Health.


Health check information appears in the Data frame for the server that iMonitor is reading the information from (not necessarily the server that you are connected to).

Reviewing Report Information

After you have generated a report, the Data frame shows the report results. If you have servers that aren't healthy in your tree, the report is divided into three categories (grouping begins with servers that have the poorest health):

- ◆ Servers with warnings
- ◆ Servers that are suspect
- ◆ Servers that are OK

If none of your servers has warnings or is suspect, those categories are not shown.

For servers that are not healthy, you can click the Agent Health Sub-Report link  next to each server. Use the online context-sensitive help to resolve the issues. This can help you determine what each of the options means and why it is important, how to resolve any issues, how to adjust the ranges, and whether you want certain options to be included in the health check.

IMPORTANT: If you have a server reported with warnings, we strongly recommend that you resolve the issues with that server. Servers that are suspect should also be evaluated.

For More Information

The tools and techniques used to keep eDirectory healthy are documented in the Novell Certified Directory Engineer Course 991: Advanced eDirectory Tools and Diagnostics. In this course you learn how to

- ◆ Perform eDirectory health checks.
- ◆ Perform eDirectory operations properly.
- ◆ Properly diagnose, troubleshoot, and resolve eDirectory issues.
- ◆ Use eDirectory troubleshooting tools and utilities.

To learn more about this course, visit the [Novell Training Services Web site \(http://www.novell.com/training/index.html\)](http://www.novell.com/training/index.html).

Resources for Monitoring

The Novell DSTrace utility runs on NetWare, Windows NT, Linux, Solaris, AIX, and HP-UX. This tool helps you monitor the vast resources of eDirectory. For more information on DSTrace, see the following:

- ◆ [“Configuring Trace Settings” on page 175](#)
- ◆ [Looking Into the Directory Services Trace \(DSTrace\) Options \(http://developer.novell.com/research/sections/netmanage/dirprimer/2001/august/spv.htm\)](http://developer.novell.com/research/sections/netmanage/dirprimer/2001/august/spv.htm)
- ◆ [More on Using the DSTrace Command \(http://developer.novell.com/research/sections/netmanage/dirprimer/2001/septembe/p010901.htm\)](http://developer.novell.com/research/sections/netmanage/dirprimer/2001/septembe/p010901.htm)

You can also invest in third-party products that provide additional management solutions for your eDirectory environment. For more information, see the following Web sites:

- ◆ [BindView \(http://www.bindview.com\)](http://www.bindview.com)
- ◆ [Blue Lance \(http://www.bluelance.com\)](http://www.bluelance.com)
- ◆ [NetPro* \(http://www.netpro.com\)](http://www.netpro.com)

If you need to monitor or audit certain characteristics of eDirectory that our partners do not provide, Novell Consulting Services can help you use the Novell Event System for customized assessment and auditing.

Upgrading Hardware or Replacing a Server

This section provides information about transferring or safeguarding eDirectory on a specific server when you upgrade or replace hardware. It is based on information in [“Backing Up and Restoring Novell eDirectory” on page 365](#).

The Backup eDirectory Management Tool allows you to prepare eDirectory information on a server for

- ◆ [“Planned Hardware or Storage Device Upgrade without Replacing the Server” on page 447](#)
- ◆ [“Planned Replacement of a Server” on page 451](#)

Planned Hardware or Storage Device Upgrade without Replacing the Server

If you are planning to upgrade hardware such as a storage device or RAM, you prepare by doing a cold backup of eDirectory using the Backup eMTool, as well as a file system backup. This will let you safeguard the server’s eDirectory identity and file system data, which has the following benefits:

- ◆ If you are replacing storage devices, the backups let you transfer information from the old storage devices to the new.
- ◆ If you are replacing the storage device that includes the disk partition/volume containing eDirectory, having this backup information lets you use the restore process to re-create the eDirectory database on the new storage device.
- ◆ Doing a cold backup of eDirectory and keeping the database closed afterward means you can upgrade hardware and transfer the database without worrying that the database has changed since the backup.

- ◆ If anything goes wrong, you have backups you can use to recover.

For the eDirectory cold backup, you must use the options to lock and disable eDirectory on the server, preventing any data change after the backup is made. To other servers that communicate with this server, the server appears to be down. Any eDirectory information that is normally sent to the server is stored by other servers in the tree until they can communicate with the server again. The stored information is used to synchronize the server when you bring it back online.

NOTE: Because other servers in the eDirectory tree expect the server to come back online quickly, you should complete the upgrade promptly and open the eDirectory database on the server as soon as possible.

To perform a planned hardware upgrade:

- 1 If you are concerned that the upgrade might cause a problem for your server, you might want to prepare another machine to use if necessary.

See [“1. Preparing for a Server Replacement” on page 451](#).

- 2 Use an eMBox Client command like the following to do a cold backup of the eDirectory database and keep the database closed and locked when finished.

```
backup -f backup_filename_and_path
-l log_filename_and_path -t -c -o -d
```

See [“Backing Up Manually with the eMBox Client” on page 393](#) and [“Backup and Restore Command Line Options” on page 403](#) for more information about using the eMBox Client and the switches.

The eDirectory database is now locked. You must leave it locked so that no new data changes will be made on that server until you finish the procedure.

Complete the rest of the procedure promptly, to minimize the amount of time that the server is unavailable.

- 3 Back up the file system using your backup tool of choice. (For NetWare, you can use SMS™.) It's important to do this *after* backing up the database, so that the eDirectory backup files are saved to tape along with the rest of the file system.
- 4 Down the server and replace the hardware.
- 5 After replacing the hardware, proceed by following the instructions for the kind of hardware change you made:

If you...	Perform These General Steps
Did not make any changes to storage devices	Bring up the server and unlock the database.
Replaced storage devices, but the disk partition/volume containing eDirectory was not affected	<ol style="list-style-type: none"> 1. Bring up the server and eDirectory. 2. Restore the file system only for the disk partitions/volumes that were on the storage devices you changed. 3. Unlock the eDirectory database.

If you...	Perform These General Steps
Replaced the storage device that contained eDirectory on an operating system that is not NetWare	<ol style="list-style-type: none"> 1. Install the operating system if necessary. 2. Restore the file system on disk partitions that were affected by the storage device change. 3. Install eDirectory on the new storage device, in a new temporary tree. 4. Restore eDirectory from backup (which puts it back into the original tree), specifying the option to keep it closed and locked after the restore. Use a command like the following: <pre>restore -r -f <i>backup_filename_and_path</i> -i <i>log_filename_and_path</i></pre> <p>Add the -u option if you backed up files listed in an include file.</p> 5. Unlock the eDirectory database. 6. If you restored NIS security files, after completing the restore, restart the server to reinitialize the security system. 7. Check to see whether the server responds as usual. <p>Use ConsoleOne[®] to check the server and its synchronization. Make sure that login scripts and printing work correctly.</p> 8. If you were using roll-forward logging on this server, make sure you re-create the roll-forward logs configuration after the restore is complete. After turning on the roll-forward logs, you must also do a new full backup. <p>The settings are reset to the default after a restore, which means roll-forward logging is turned off. The new full backup is necessary so that you are prepared for any failures that might occur before the next unattended full backup is scheduled to take place.</p>

If you...	Perform These General Steps
Replaced the storage device that contained the sys: volume and eDirectory on NetWare	<p>You must be aware of the issues involved with preserving file system rights when restoring file system data on NetWare. You should restore eDirectory before restoring the file system. You might also need to take additional steps, as explained in “Preserving Rights When Restoring File System Data on NetWare” on page 379.</p> <ol style="list-style-type: none"> 1. Install NetWare and eDirectory on the new storage device, creating a new sys: volume in a new temporary tree. 2. Place the eDirectory backup files on that volume, copied from your tape backup. 3. Restore eDirectory from backup (which puts it back into the original tree), specifying the option to keep it closed and locked after the restore. Use a command like the following: <pre>restore -r -f <i>backup_filename_and_path</i> -l <i>log_filename_and_path</i></pre> <p>Add the -u option if you backed up files listed in an include file.</p> 4. Restore the file system for all the volumes affected by the storage device change. 5. Unlock the eDirectory database. 6. If you restored NCI security files, after completing the restore, restart the server to reinitialize the security system. 7. Check to see whether the server responds as usual. Use iMonitor to check the server and its synchronization. Make sure that login scripts and printing work correctly. 8. If you were using roll-forward logging on this server, make sure you re-create the roll-forward logs configuration after the restore is complete. After turning on the roll-forward logs, you must also do a new full backup. The settings are reset to the default after a restore, which means roll-forward logging is turned off. The new full backup is necessary so that you are prepared for any failures that might occur before the next unattended full backup is scheduled to take place.

If the server does not respond as usual, you might need to recover by doing one of the following:

- ◆ Re-create the hardware configuration you had before, because it was working before the change.
- ◆ Transfer this server’s identity to another machine using the file system and eDirectory backups you made. See [“Planned Replacement of a Server” on page 451](#).

Planned Replacement of a Server

The following instructions are designed for situations where a server is actually replaced by moving the server's eDirectory identity and file system data onto a different machine. For naming purposes in these instructions, the old server is referred to as Server A, and its replacement is referred to as Server B.

You prepare by doing a cold backup (a backup done while the database is closed) of eDirectory using the Backup eMTool, as well as a file system backup using your tool of choice. This backup information lets you use the restore process to re-create the server on the new machine.

For the eDirectory cold backup, you must use the options to lock and disable eDirectory on Server A, preventing any data change after the backup is made. To other servers that communicate with this server, the server appears to be down. Any eDirectory information that is normally sent to the server is stored by other servers in the tree until they can communicate with the server again. The stored information is used to synchronize the server when you bring it back online on the new machine, Server B.

NOTE: Because other servers in the eDirectory tree expect the server to come back online quickly, you should complete the change and restore eDirectory information on the server as soon as possible.

Follow these general steps to replace a server:

1. To reduce down time for Server A while you are replacing it, it's best to prepare Server B as much as possible before you begin the replacement, by installing the operating system, etc., as described in [“1. Preparing for a Server Replacement” on page 451](#).
2. Do the eDirectory and file system backups on Server A as described in [“2. Creating a Backup of eDirectory” on page 452](#).
3. Transfer the information to Server B as described in [“3. Restoring eDirectory Information for a Server Replacement” on page 452](#).

1. Preparing for a Server Replacement

Use the following checklists for Server A and Server B to determine whether you are ready to replace Server A. Preparing Server B before proceeding will reduce the time the server is down while you transfer from one machine to the other.

Preparation for Server A

- Make sure that Server A has the latest version of the operating system installed.
- Make sure the tree for Server A is healthy by running DSRepair on the server that holds the master of the Tree partition and by running time synchronization.
- Run DSRepair on the database of Server A. Ensure that Server A is synchronized completely.

Preparation for Server B

- Install the latest version of the operating system. This must be the same operating system as Server A.
- Install eDirectory, putting Server B in a new temporary tree.

(Restoring eDirectory during [“3. Restoring eDirectory Information for a Server Replacement” on page 452](#) will put Server B into the original tree that Server A was in.)

- ❑ (NetWare only) Be aware of the issues involved with preserving file system rights when restoring file system data as part of replacing a server. You should plan to restore eDirectory before restoring the file system. You might also need to take additional steps, as explained in [“Preserving Rights When Restoring File System Data on NetWare” on page 379.](#)

Continue with the steps in the next section, [“2. Creating a Backup of eDirectory.”](#)

2. Creating a Backup of eDirectory

You must create a backup of eDirectory prior to a server replacement. After completing [“1. Preparing for a Server Replacement” on page 451](#), use the eMBox Client to do a cold backup of the eDirectory database on Server A, using the advanced options to disable and lock the database after the backup.

To create a cold backup (a backup done while the database is closed) of eDirectory and keep the database closed afterward:

- 1** Make sure you have completed [“1. Preparing for a Server Replacement” on page 451](#).
- 2** Do a cold backup of the eDirectory database on Server A and keep the database closed and locked when finished, by using a backup command like the following in the eMBox Client with the `-c`, `-o`, and `-d` switches:

```
backup -f backup_filename_and_path  
-l log_filename_and_path -t -c -o -d
```

See [“Backing Up Manually with the eMBox Client” on page 393](#) and [“Backup and Restore Command Line Options” on page 403](#) for more information about using the eMBox Client and the switches.

Server A’s eDirectory database is now locked. You must leave it locked so that no new data changes will be made on that server until you bring it back into the tree by restoring onto Server B.

Complete the rest of the server upgrade or replacement procedure promptly, to minimize the amount of time that the server is unavailable.

- 3** Make a full backup of Server A’s file system. (For NetWare, you can use SMS.)
It’s important to do the file system backup *after* backing up the database, so that the eDirectory backup files are saved to tape along with the rest of the file system.

For complete information on using SMS, see the [Storage Management Services Administration Guide](#) (<http://www.novell.com/documentation/lg/nw65/smsadmin/data/hjc2z4tu.html>).

- 4** Lock the eDirectory database on Server A and unplug Server A from the network.
Continue with the steps in [“3. Restoring eDirectory Information for a Server Replacement” on page 452](#).

3. Restoring eDirectory Information for a Server Replacement

To transfer Server A’s eDirectory identity and file system to Server B:

- 1** Make sure you have completed [“1. Preparing for a Server Replacement” on page 451](#) and [“2. Creating a Backup of eDirectory” on page 452](#).
- 2** Make sure Server B is up and eDirectory is running.
- 3** Use restore to transfer Server A’s eDirectory identity and file system to Server B:

3a Copy the eDirectory cold backup files created for Server A to Server B.

The backup files can be made much smaller using a third-party file compression tool, because they compress well. This could help you copy the files faster.

3b Restore the eDirectory database from Server A onto Server B using the eDirectory backup files you copied. In the eMBox command line client, use a command like the following:

```
restore -r -f backup_filename_and_path
-l log_filename_and_path
```

Add the -u option if you backed up files listed in an include file. (See “[Restoring from Backup Files with the eMBox Client](#)” on page 400 and “[Backup and Restore Command Line Options](#)” on page 403 for more information about using the eMBox Client and the switches.)

No roll-forward logs need to be included in the restore, because you did a cold backup and kept the database closed afterward. No transactions have occurred in the database because it’s closed, so no roll-forward logs have been created since the backup.

IMPORTANT: On NetWare, it is especially important for you to restore eDirectory *before* the file system, so that trustee assignments and rights are preserved when the file system data is restored. For more information, see “[Preserving Rights When Restoring File System Data on NetWare](#)” on page 379.

3c Transfer Server A’s file system data onto Server B, from backup.

4 (NetWare only) Rename Server B using Server A’s IP address and server name in autoexec.ncf.

5 If you use NICE, restart the server to reinitialize NICE so it will use the restored NICE security files.

6 Unlock the eDirectory database.

7 After completing the restore, check to see whether Server B has successfully taken on Server A’s identity and is responding as usual. Use ConsoleOne to check the server and its synchronization. Make sure that login scripts, printing, and NICE security work correctly.

If the server responds as usual, you are finished with the server replacement. You can now uninstall eDirectory from Server A to remove its eDirectory identity, then use the machine for another purpose. Do not bring Server A back up on the network until you remove eDirectory, or it will cause confusion in the network with eDirectory synchronization because Server A and Server B will compete for the same identity.

8 (Conditional) If you were using roll-forward logging on this server, make sure you re-create the roll-forward logs configuration after the restore is complete. After turning on the roll-forward logs, you must also do a new full backup.

The settings are reset to the default after a restore, which means roll-forward logging is turned off. The new full backup is necessary so that you are prepared for any failures that might occur before the next unattended full backup is scheduled to take place.

If Server B does not work correctly and you need Server A’s identity and file system to be available right away, you can do the following:

1 Unplug Server B’s network cable or down the server.

2 Reattach Server A to the network, start it, then open the eDirectory database.

Ignore system messages requesting you to run DSRepair.

- 3 Remove eDirectory from Server B and try the upgrade again.

Restoring eDirectory after a Hardware Failure

A hard disk failure involving the disk partition/volume where eDirectory is located is equivalent to removing eDirectory from the server. (Fortunately, in a multi-server environment, one server can go down while the rest of the servers in the replica ring remain intact.)

To restore eDirectory after a failure of the disk partition/volume that it resides on, follow the procedures for restoring from your backup files as described in [“Preparing for a Restore” on page 384](#) and [“Restoring from Backup Files with iManager” on page 390](#) (or [“Restoring from Backup Files with the eMBox Client” on page 400](#)).

During the new installation, follow any instructions provided by the manufacturer to verify that the server’s hard disks are working. The new hard disk should have at least the same storage capacity as the drive it replaces. Use the local server information files to verify configuration information.

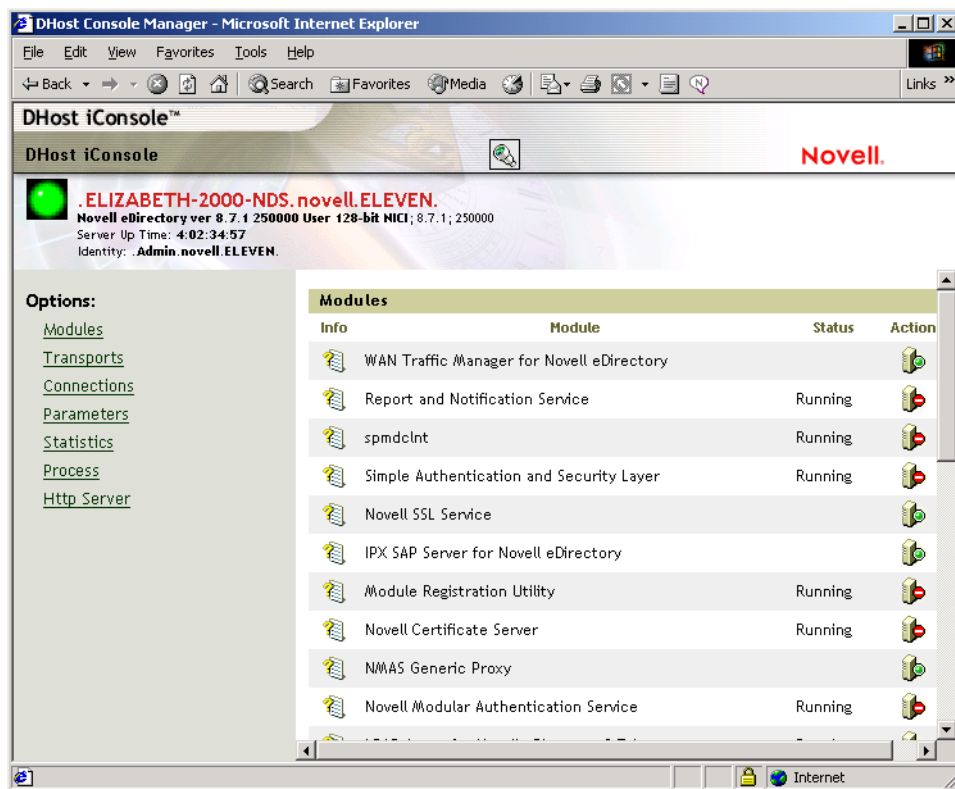
NOTE: If you do not have backup files for the server, use the XBrowse tool to query eDirectory to help you recover server information. You must do this before you remove the Server object or any associated objects from the tree. XBrowse and additional information are available from Novell Support, [Technical Information Document #2960653](#) (<http://support.novell.com/servlet/tidfinder/2960653>).

16 DHost iConsole Manager

DHost iConsole Manager is a Web-based browser administrative tool that lets you:

- ◆ Manage DHost modules
- ◆ Query for DHost configuration parameters
- ◆ View DHost connection information
- ◆ View thread pool statistics
- ◆ View details about protocols registered with the DHost protocol stack manager

Figure 36 DHost iConsole Manager



DHost iConsole Manager can also be used as a diagnostic and debugging tool by letting you access the HTTP server when the eDirectory server is not functioning correctly (see [“Setting the SAdmin Password”](#) on page 462 for more information).

This chapter contains the following information:

- ◆ [“What is DHost?”](#) on page 456

- ◆ “Running DHost iConsole” on page 457
- ◆ “Managing eDirectory Modules” on page 458
- ◆ “Querying for DHost Information” on page 459
- ◆ “Process Stack” on page 461
- ◆ “Setting the SAdmin Password” on page 462

What is DHost?

Novell® eDirectory™ software for Windows, Solaris, Linux, AIX, and HP-UX is built upon the same core code as eDirectory for NetWare®. In order for eDirectory for Windows and UNIX to properly interact with the other versions of eDirectory, they support a subset of NetWare Core Protocol™ (NCP™) services. This is handled by a program called DHost. DHost sits beneath eDirectory and provides functionality on non-NetWare platforms that the NetWare operating system provides naturally.

DHost provides the following NetWare oriented services:

Service	Description
NCP Engine	<p>A packet-based protocol that enables a client to send requests to and receive replies from a NetWare server.</p> <p>For more information, see NetWare Core Protocols (http://developer.novell.com/ndk/doc/ncp/ncp__enu/data/hc4lztgy.html).</p>
Watchdog	<p>Packets used to make sure workstations are still connected to the NetWare server.</p> <p>For more information, see Watchdog Packet Spoofing (http://www.novell.com/documentation/lg/nw65/ipx_enu/data/h0cufuir.html).</p>
Connection Table	<p>A unique number assigned to any process, print server, application, workstation, or other entity that attaches to a NetWare server. The number can be different each time an attachment is made. Connection numbers are used in implementing network security and for network accounting. They reflect the objects place in the file servers connection table. Additionally, they provide an easy way to identify and obtain information about the objects logged in on the network.</p>
Event System	<p>Provides a way for applications to monitor the activity of an individual server.</p>
Thread Pool	<p>A sequence of instructions executed as an independent entity and scheduled by system software.</p>
NCP Extensions	<p>Allows server application developers to write NLM™ software to be implemented in the NetWare OS as NCPs.</p> <p>For more information, see NCP Extension Concepts (http://developer.novell.com/ndk/doc/ncp/index.html?page=/ndk/doc/ncp/ncp__enu/data/a1wftl8.html).</p>

Service	Description
Message Digest	A compressed or condensed form of a document, or an abstract from a document, that functions as a "digital fingerprint" of the larger document. A message digest is used to create a digital signature that is unique to a particular document.

Running DHost iConsole

- ♦ "Running DHost iConsole on NetWare" on page 457
- ♦ "Running DHost iConsole on Windows" on page 457
- ♦ "Running DHost iConsole on Linux, Solaris, AIX, and HP-UX" on page 457

Running DHost iConsole on NetWare

On NetWare, you can access the DHost iConsole through NetWare Remote Manager. `httpstk.nlm` must be running on the eDirectory server in order for you to set or change the SAdmin password.

- 1 Open a Web browser.
- 2 In the address (URL) field, enter the following:

`http://server's TCP/IP address:port`

For example:

`http://137.65.123.11:8008`

NOTE: The default alternate port number is 8008. If you have changed this value on the Configuration page in NetWare Remote Manager, make sure you enter the new port number.

If you have Domain Name Services (DNS) installed on your network for server name-to-IP address resolution, you can also enter the server's DNS name instead of the IP address.

- 3 Specify a username, context, and password.

Running DHost iConsole on Windows

- 1 Open a Web browser.
- 2 In the address (URL) field, enter the following:

`http://server.name:port/dhost`

for example:

`http://MyServer:80/dhost`

You can also use the server IP address to access the DHost iConsole. For example:

`http://137.65.135.150:80/dhost`

- 3 Specify a username, context, and password.

Running DHost iConsole on Linux, Solaris, AIX, and HP-UX

- 1 Open a Web browser.
- 2 In the address (URL) field, enter the following:

`http://server.name:port/dhost`

For example:

```
http://MyServer:80/dhost
```

You can also use the server IP address to access the DHost iConsole. For example:

```
http://137.65.135.150:80/dhost
```





- 3 Specify a username, context, and password.

Managing eDirectory Modules

The Modules page in DHost iConsole provides information about available eDirectory services and their states. You can also use the Modules page to start and stop (load or unload) these services.

You can only load or unload non-interactive modules such as LDAP, SNMP, and HTTPSTK.

The Modules page has the following attributes:

Attribute	Description
Info	Click  to display the module description, file name, module handle, attributes, and the name of so (shared object) of the selected module.
Module	Displays the module name.
Status	Displays whether the module is running or not.
Action	Indicates whether the module can be run or not. A module can be in one of the following three states:  indicates that the module is a system module and cannot be unloaded.  indicates that the module can be loaded and it is ready to load.  indicates that the module is running.

- ◆ [“Loading or Unloading Modules on NetWare” on page 458](#)
- ◆ [“Loading or Unloading Modules on Windows” on page 459](#)
- ◆ [“Loading or Unloading Modules on Linux, Solaris, AIX, and HP-UX” on page 459](#)

For more information on using Novell iManager to load and unload eDirectory services, see [“eDirectory Service Manager” on page 160](#).

Loading or Unloading Modules on NetWare

- 1 Open a Web browser.
- 2 In the address (URL) field, enter the following:

```
http://server's TCP/IP address:port
```

For example:

```
http://137.65.123.11:8008
```

NOTE: The default alternate port number is 8008. If you have changed this value on the Configuration page in NetWare Remote Manager, make sure you enter the new port number.

If you have Domain Name Services (DNS) installed on your network for server name-to-IP address resolution, you can also enter the server's DNS name instead of the IP address.

- 3** Specify a username, context, and password.
- 4** Click List Modules in the Manage Applications list.
- 5** To load a module, enter the name and click Load Module.

If you need to verify whether the module actually loaded, check the Display System Console for Module Load checkbox.

Loading or Unloading Modules on Windows

- 1** Open a Web browser.
- 2** In the address (URL) field, enter the following:



`http://server.name:port/dhost`

for example:

`http://MyServer:80/dhost`

You can also use the server IP address to access the DHost iConsole. For example:

`http://137.65.135.150:80/dhost`

- 3** Specify a username, context, and password.
- 4** Click Modules.
- 5** Click  to load a module, or  to unload a module.

Loading or Unloading Modules on Linux, Solaris, AIX, and HP-UX

- 1** Open a Web browser.
- 2** In the address (URL) field, enter the following:



`http://server.name:port/dhost`

for example:

`http://MyServer:80/dhost`

You can also use the server IP address to access the DHost iConsole. For example:

`http://137.65.135.150:80/dhost`

- 3** Specify a username, context, and password.
- 4** Click Modules.
- 5** Click  to load a module, or  to unload a module.

Querying for DHost Information

Using the DHost iConsole Manager, you can query for the following information:

- ♦ **Configuration parameters**
- ♦ **Protocols registered with the PSTACK manager**

- ◆ [Connection properties](#)
- ◆ [Summary of thread pool](#)

Viewing the Configuration Parameters

Configuration parameters are specific only to UNIX platforms.

In the DHost iConsole Manager, click Parameters. See [“Running DHost iConsole on Linux, Solaris, AIX, and HP-UX”](#) on page 457 for more information.

The configuration parameters are displayed with the following information:

Option	Description
Parameter name	Displays the name of the configuration parameter.
Default value	Displays the default value of the parameter.
Set value	Displays the value currently set.
Minimum value	Displays the minimum limit that can be set for the parameter.
Maximum value	Displays the maximum limit that can be set for the parameter.
Type	Displays the type of value that can be set for the parameter.

For more information, see [“Configuration Parameters”](#) in the *Novell eDirectory 8.7.3 Installation Guide*.

Viewing Protocol Information

In the DHost iConsole Manager, click Transports.

The following protocol information is displayed:

- ◆ ID
- ◆ Protocol
- ◆ Transports

Viewing Connection Properties

In the DHost iConsole Manager, click Connections.

The following connection properties are displayed:

- ◆ Conn
- ◆ Flags
- ◆ Identity
- ◆ Display Name
- ◆ Transport
- ◆ Authentication Name

- ◆ SEV Count
- ◆ Last Access
- ◆ Locked

Viewing the Thread Pools Statistics

In the DHost iConsole Manager, click Statistics.

The following thread pool statistics are displayed:

- ◆ Spawned Threads
- ◆ Dead Threads
- ◆ Idle Threads
- ◆ Worker Thread
- ◆ Peak Worker Thread
- ◆ Ready for Work Thread
- ◆ Ready Queue Peak Worker Threads
- ◆ Ready Queue Max Wait Time
- ◆ Schedule Delay Minimum Time
- ◆ Schedule Delay Maximum Time
- ◆ Schedule Delay Average Time
- ◆ Waiting For Work
- ◆ Peaking Waiting For Work

Process Stack

The process stack contains a list of all threads currently running in the DHost process space. You can get detailed information on a thread by clicking the thread ID. This feature is used mainly as a low-level debugging tool for Novell engineers and support personnel.

This option is available only on Windows.

- 1** Open a Web browser.
- 2** In the address (URL) field, enter the following:
`http://server.name:port/dhost`
for example:
`http://MyServer:80/dhost`
You can also use the server IP address to access the DHost iConsole. For example:
`http://137.65.135.150:80/dhost`
- 3** Specify a username, context, and password.
- 4** Click Process.
- 5** To view the call stack for a thread, click the thread ID.

Setting the SAdmin Password

You can set up a preconfigured admin user which allows access to the HTTP Protocol Stack (HTTPSTK) when eDirectory is not loaded. The preconfigured admin user, SAdmin, has rights that are equivalent to the eDirectory Admin User object. If the server is in a state where eDirectory is not functioning correctly, you can log in to the server as this user and perform all the diagnostic and debugging tasks necessary that do not require eDirectory.

- ◆ “Setting the SAdmin Password on NetWare” on page 462
- ◆ “Setting the SAdmin Password on Windows” on page 462
- ◆ “Setting the SAdmin Password on Linux, Solaris, AIX, and HP-UX” on page 463

Setting the SAdmin Password on NetWare

Use NetWare Remote Manager to enable the SAdmin User object and set or change the password for this object. httpstk.nlm must be running on the eDirectory server in order for you to set or change the SAdmin password.


- 1 Open a Web browser.
- 2 In the address (URL) field, enter the following:

```
http://server's TCP/IP address:port
```

 For example:

```
http://137.65.123.11:8008
```

NOTE: The default alternate port number is 8008. If you have changed this value on the Configuration page in NetWare Remote Manager, make sure you enter the new port number.

 If you have Domain Name Services (DNS) installed on your network for server name-to-IP address resolution, you can also enter the server's DNS name instead of the IP address.
- 3 Specify a username, context, and password.
- 4 Click the Configure button  > Enable Emergency Account (SADMIN User) and Set Password.
- 5 Specify an SAdmin password, then verify the password you just specified.
- 6 Click Set.

Setting the SAdmin Password on Windows

Use the DHOST remote manager page (accessible through the /dhost URL or from the root page) to set the SAdmin password. dhost.exe must be running on the eDirectory server in order for you to set or change the SAdmin password.

- 1 Open a Web browser.
- 2 In the address (URL) field, enter the following:

```
http://server.name:port/dhost
```

 for example:

```
http://MyServer:80/dhost
```

 You can also use the server IP address to access the DHost iConsole. For example:

```
http://137.65.135.150:80/dhost
```

- 3 Specify a username, context, and password.
- 4 Click HTTP Server, then specify an SAdmin password.
- 5 Verify the password you just specified, then click Submit.

Setting the SAdmin Password on Linux, Solaris, AIX, and HP-UX

You can use either of the following to set the SAdmin password on Solaris, Linux, AIX, or HP-UX systems:

- ♦ “DHOST Remote Management Page” on page 463
- ♦ “Ndsconfig” on page 463

DHOST Remote Management Page

Use the DHOST remote manager page (accessible through the /dhost URL or from the root page) to set the SAdmin password. Novell eDirectory server must be running on the eDirectory server in order for you to set or change the SAdmin password.

- 1 Open a Web browser.
- 2 In the address (URL) field, enter the following:

```
http://server.name:port/dhost
```

for example:

```
http://MyServer:80/dhost
```

You can also use the server IP address to access the DHost iConsole. For example:

```
http://137.65.135.150:80/dhost
```

- 3 Specify a username, context, and password.
- 4 Click HTTP Server, then specify an SAdmin password.
- 5 Verify the password you just specified, then click Submit.

Ndsconfig

Use the ndsconfig utility to set the SAdmin password. ndsd must be running on the eDirectory server in order for you to set or change the SAdmin password.

Enter the following at the server console

```
ndsconfig set http.server.sadmin-pwd=password
```

where *password* is the new SAdmin password.

For more information on using the ndsconfig utility, see “ndsconfig Utility Parameters” in the *Novell eDirectory 8.7.3 Installation Guide*.

17

The eDirectory Management Toolbox

The Novell® eDirectory™ Management Toolbox (eMBox) lets you access all of the eDirectory backend utilities remotely as well as on the server.

eMBox works with Novell iManager to provide Web-based access to eDirectory utilities such as DSRepair, DSMerge, Backup and Restore, and Service Manager.

IMPORTANT: Role Based Services must be configured through iManager to the tree that is to be administered in order for eMBox tasks to be run.

All functions are accessible, either on the local server or remotely, through a command line client. You can perform tasks for multiple servers from one server or workstation using the eMBox Client.

For all eDirectory Management Tools (eMTools)—such as Backup, DSRepair, DSMerge, Schema Operations, and eDirectory Service Manager—to run, eMBox must be loaded and running on the eDirectory server.

In this section:

- ♦ [“Using the eMBox Command Line Client” on page 465](#)
- ♦ [“Using the eMBox Logger” on page 474](#)

Using the eMBox Command Line Client

One way to access eMBox is to use its Java command line client. The command line client has two modes: interactive and batch. In the interactive mode, you run the eMBox commands one at a time. In the batch mode, you can run a group of commands unattended. The command line client has logging service for both modes.

The command line client is a Java application. To run it, you must have access to the Java Runtime Environment, Sun JVM 1.3.1, which is installed with eDirectory. You must also have access behind the firewall to the servers you want to manage. You can perform tasks for multiple servers from one server or workstation.

In this section:

- ♦ [“Displaying the Command Line Help” on page 466](#)
- ♦ [“Running the eMBox Command Line Client in Interactive Mode” on page 466](#)
- ♦ [“Running the eMBox Command Line Client in Batch Mode” on page 470](#)
- ♦ [“eMBox Command Line Client Options” on page 472](#)
- ♦ [“Establishing a Secure Connection with the eMBox Client” on page 473](#)
- ♦ [“Finding Out eDirectory Port Numbers” on page 473](#)

Displaying the Command Line Help

To display the eMBox general command line help before going in to the eMBox Client, do the following:

- ◆ NetWare® and UNIX: At the command line, enter `edirutil -?`.
- ◆ Windows: Run
`drive\novell\nds\edirutil.exe -?`

To display the eMBox interactive command line help while you are in the interactive mode, at the eMBox Client prompt enter a question mark (?). For example,
eMBox Client> ?

The help displays information on the command line options like the information in [“eMBox Command Line Client Options” on page 472](#).

Running the eMBox Command Line Client in Interactive Mode

Interactive mode lets you run eMBox commands one at a time.

In this section:

- ◆ [“Running the eMBox Client on an eDirectory Server” on page 466](#)
- ◆ [“Running the eMBox Client on a Workstation” on page 467](#)
- ◆ [“Logging In to a Server” on page 468](#)
- ◆ [“Setting Preferred Languages, Timeout, and Log File” on page 468](#)
- ◆ [“Listing eMTools and Their Services” on page 469](#)
- ◆ [“Running a Particular Service” on page 469](#)
- ◆ [“Logging Out From the Current Server” on page 470](#)
- ◆ [“Exiting the Client” on page 470](#)

Running the eMBox Client on an eDirectory Server

The eMBox Client and Sun JVM 1.3.1 are installed with eDirectory. To open the eMBox Client in interactive mode on an eDirectory server, do the following:

- ◆ NetWare and UNIX: At the command line, enter `edirutil -i`.
- ◆ Windows: Run
`drive\novell\nds\edirutil.exe -i`

The `edirutil` file gives you a shortcut to running the eMBox Client. It points to the Java executable and the default location where the eMBox Client is installed with eDirectory, and for NetWare, it includes the necessary `-ns` option (which is a Java option on NetWare meaning “new screen”). (You can also enter the information manually, as described in [“Setting Up the Path and Classpath for eMBox Client” on page 467](#).)

You must have access behind the firewall to use the eMBox command line client for the servers you want to manage—so if you are remote, you’ll need VPN access.

Running the eMBox Client on a Workstation

To use the eMBox Client on a machine other than an eDirectory server:

- ◆ Copy the eMBoxClient.jar file from an eDirectory server to your machine.
 - ◆ NetWare: sys:\system\embox\eMBoxClient.jar
 - ◆ Windows: \novell\nds\embox\eMBoxClient.jar
 - ◆ UNIX: /usr/lib/nds-modules/embox/eMBoxClient.jar
- ◆ Make sure the machine has Sun JVM 1.3.1 installed.
- ◆ Make sure you have access behind the firewall to use the eMBox command line client for the servers you want to manage.

You can't use the edirutil command on a workstation as a shortcut to getting in to the eMBox Client in interactive mode as you can on a server. You must either set up the environment once in your path and class path, or enter it manually each time. See [“Setting Up the Path and Classpath for eMBox Client” on page 467](#).

Setting Up the Path and Classpath for eMBox Client

If you are running the eMBox Client on an eDirectory server and have not changed the location of Java or the eMBoxClient.jar file, you can use edirutil as a shortcut to running the eMBox Client. (See [“Running the eMBox Client on an eDirectory Server” on page 466](#).)

But if you have changed the default locations, or you are running the eMBoxClient.jar file on a machine that is not a server, or you want to enter the classpath manually, you need to set up the path and classpath for the eMBox Client as explained in this section.

You can run the eMBox Client from anywhere on your machine if you do the following:

- ◆ Add to your path the directory where the Java executable (for example, java.exe) is located, or make sure that Java is already running.

If you are on a server, this is probably already done for you. On Windows and UNIX servers, the directory needs to be in your path. On NetWare, instead of adding the directory to a path, Java needs to be running.

On a workstation, you might need to set it up yourself. For example, on Windows, click Start > Settings > Control Panel > System. On the Advanced tab, click Environment Variables and add the path to the Path variable.

To enter this manually: If the path to the Java executable has not been added to your path, at the command line you will need to first change to the directory containing the Java executable before running embox. For example, on Windows enter

```
cd c:\novell\nds\embox\jre\bin
```

- ◆ Add the path to the eMBoxClient.jar file to your classpath.

NetWare server:

```
set ENVSET=path\eMBoxClient.jar
```

Windows server or workstation:

```
set CLASSPATH=path\eMBoxClient.jar
```

UNIX server or workstation:

```
export CLASSPATH=path/eMBoxClient.jar
```

To enter this manually: An alternative way to specify the classpath is to use the `-cp` flag for Java each time you want to run eMBox:

```
java -cp path/eMBoxClient.jar embox -i
```

For example, on Windows enter

```
java -cp c:\novell\nds\embox\eMBoxClient.jar embox -i
```

WARNING: On a NetWare server only, to avoid an abend you *must* include `-ns` (a Java option on NetWare for “new screen”). For example,

```
java -ns -cp sys:\system\embox\eMBoxClient.jar embox -i
```

After doing both of these steps, you can run the client in interactive mode from anywhere on your machine using the following command:

```
java embox -i
```

WARNING: On a NetWare server only, to avoid an abend you *must* include `-ns` (a Java option on NetWare for “new screen”). For example,

```
java -ns embox -i
```

For information on Java commands, see the Java documentation on the [Sun Web site \(http://java.sun.com\)](http://java.sun.com).

Logging In to a Server

To log in to a server, you need to specify the server name or IP address and the port number to connect to a particular server. A username and password are not needed for public logins.

For example, after opening the eMBox Client in interactive mode, enter

```
login -s 137.65.123.244 -p 8008 -u admin.mycompany  
-w mypassword -n
```

For more information about port numbers, see “[Finding Out eDirectory Port Numbers](#)” on [page 473](#).

Setting Preferred Languages, Timeout, and Log File

The default language is the client system language, so in most cases you won't need to explicitly set a language. Similarly, the default timeout should work in most cases. To set the log file, specify the filename and the mode for opening it (append or overwrite).

See the following table for sample commands.

Command	Description
<code>set -L en,de</code>	Sets the language preference to English and German (in that order).
<code>set -T 100</code>	Sets the timeout to 100 seconds. The timeout setting specifies how long to wait for responses from the server.
<code>set -l mylog.txt -o</code>	Uses mylog.txt as the log file and overwrites when opening it. Default=append

Listing eMTools and Their Services

After logging in to a server, you can use the list command to display a list of the services available on that server.

The list command displays the following eMTools and their services dynamically:

eMTool	Description
backup	Novell eDirectory Backup eMTool
dsmerge	Novell eDirectory Merge eMTool
dsrepair	Novell eDirectory Repair eMTool
dsschema	Novell eDirectory Schema Operations eMTool
service	Novell eDirectory Service Manager eMTool

Use `-r` to force the refresh of the list. Use `-t` to list service details. Use `-f` to list just the command format.

See the following table for sample commands.

Command	Description
<code>list</code>	Lists the eMTools available on the server.
<code>list -r</code>	Refreshes the eMTool list.
<code>list -t backup</code>	Lists Backup services with details.
<code>list -t dsrepair</code>	Lists DSRepair services with details.
<code>list -t dsmerge -f</code>	Lists DSMerge services with command formats only.

Running a Particular Service

You can perform tasks using each of the eMTool services after you have logged in to a server. For example:

Command	Description
<code>dsrepair.rld</code>	Repair local database.
<code>backup.getconfig</code>	Get backup configuration information.

For more information, see the following:

- ◆ [“Using the eMBox Client for Backup and Restore” on page 393](#)
- ◆ [“Using the eMBox Client to Merge Trees” on page 200](#)
- ◆ [“Using the eMBox Client to Repair a Database” on page 225](#)
- ◆ [“Using the eMBox Client Service Manager eMTool” on page 160](#)

Logging Out From the Current Server

To log out from the current session, use the following command:

```
logout
```

If you log in to a different server, you don't need to use this command; you are automatically logged out of the current server.

Exiting the Client

To exit the client, use either of the following commands:

```
exit
```

or

```
quit
```

Running the eMBox Command Line Client in Batch Mode

There are three ways you can run the eMBox Client in batch mode:

- ◆ “Single Tasks” on page 470
- ◆ “Internal Batch File” on page 470
- ◆ “System Batch File” on page 471

You can use a combination of the system and internal batch files for more flexibility and for organizing and reusing commands that you run often.

Single Tasks

You can perform a single eMBox task in batch mode at the command line, simply by entering the command using the `-t` option to specify the tool and task, and omitting the `-i` option (`-i` specifies interactive mode). For example,

```
java embox -s 137.65.123.244 -p 8008 -u admin.mycompany  
-w mypassword -l mylog.txt -t dsrepair.rld -n
```

WARNING: On NetWare only, to avoid an abend you *must* include `-ns` (a Java option on NetWare for “new screen”). For example,

```
java -ns embox -s 137.65.123.244 -p 8008 -u admin.mycompany -w mypassword -l mylog.txt  
-t dsrepair.rld -n
```

For multiple tasks on different servers, or for tasks you perform often, a better alternative is to use an internal batch file. For more information, see the following section, “[Internal Batch File](#).”

Internal Batch File

To run the eMBox Client in batch mode using an eMBox Client internal batch file, you need to create a file which contains a group of eMBox commands you would run in the interactive mode.

An eMBox Client internal batch file lets you run all the commands in the batch file without your attention. You can perform multiple tasks with multiple eMBox tools on the same server without logging in and logging out again for each task. From one server, you can also perform tasks with multiple eMBox tools on multiple servers.

Internal batch files can help you organize and reuse commands that you perform often, so you don't need to enter them manually at the command line each time.

You can go to the command line and run the internal batch file using an eMBox Client command. For example, this command logs in to a server and runs the commands listed in the mybatch.mbx file:

```
java embox -s 137.65.123.244 -p 8008 -u admin.mycompany -w mypassword -l
mylog.txt -o -b mybatch.mbx -n
```

WARNING: On NetWare only, to avoid an abend you *must* include `-ns` (a Java option on NetWare for “new screen”). For example,

```
java -ns embox -s 137.65.123.244 -p 8008 -u admin.mycompany -w mypassword -l mylog.txt
-o -b mybatch.mbx -n
```

Another option is to put the same kind of command in a system batch file, so that you can schedule it to run on the server unattended. See [“System Batch File” on page 471](#).

Here is an example of an eMBox internal batch file. It contains examples of the commands you could run and an example of logging in to a different server. This example assumes that you logged in to a server when you opened the eMBox Client. (Each command must be on a separate line. Lines beginning with # are comments.)

```
# This file is named mybatch.mbx.
# This is an example of commands you could use in
# an eMBox internal command batch file.

# Backup commands
backup.getconfig
backup.backup -b -f mybackup.bak -l backup.log -t -w

# DSRepair commands
dsrepair.rld

# Log in to a different server
login -s 137.65.123.255 -p 8008 -u admin.mycompany -w mypassword -n

# DSmerge commands
dsmerge.pr -u admin.mycompany -p admin.mycompany -n mypassword

# Schema Operations
dsschema.rst
dsschema.dse
dsschema.rls
dsschema.gsu
dsschema.scc
dsschema.irs -n LocalTree

# DSService commands
service.serviceList

# End of example.
```

System Batch File

As with other command line tools, you can create system batch files containing eMBox Client commands and run them manually at the command line or schedule them to run on the server unattended. For example, you can run backups unattended, using system batch files like the examples described in [“Doing Unattended Backups, Using a Batch File with the eMBox Client” on page 396](#).

From one server, you can perform tasks with multiple eMBox tools on multiple servers.

In a system batch file, you can use a combination of eMBox Client single commands and internal batch files for more flexibility and for organizing and reusing commands that you run often. For more information, see “[Internal Batch File](#)” on page 470 above.

Consult the documentation for your operating system or third-party scheduling software for instructions on how to run batch files unattended.

NOTE: On NetWare, you can use third-party scheduling software, or you can consider using [CRON.NLM](http://support.novell.com/servlet/tidfinder/2939440) (<http://support.novell.com/servlet/tidfinder/2939440>), an unsupported tool available for download from Novell Technical Support.

eMBox Command Line Client Options

Option	Description
-? or -h	Display help information
-i	Interactively run eMBox commands one at a time.
-s <i>server</i>	Name or IP address of the eMBox server. Default=127.0.0.1
-p <i>port</i>	Port number of the eMBox server. Default=80
-u <i>user</i>	User DN. For example, admin.mycompany. Default=anonymous
-w <i>password</i>	Password associated with the user specified with -u.
-m <i>mode</i>	Login mode. Default=dclient
-n	Do not try to make a secure SSL connection. Use a nonsecure connection. If you do not use this option, the eMBox Client will try to establish an SSL connection, and you must have the JSSE files in your class path or it will return an error. See “ Establishing a Secure Connection with the eMBox Client ” on page 473 for more information.
-l <i>log file</i>	Name of the log file.
-o	Overwrite the log file when opening it.
-T <i>timeout</i>	How long (in seconds) to wait for responses from the server.
-L <i>language</i>	List of comma-delimited acceptable languages in order of preference, such as en-US,de_DE. This option defaults to the client system language.
-t [<i>tool.</i>] <i>task options</i>	Perform a single service with this connection. The string following -t should be a valid eMBox command.
-b <i>eMBox batch file</i>	Perform a group of services as specified in the batch file. The eMBox commands in the batch file should be put on separate lines. Lines preceded by # are comments.

Establishing a Secure Connection with the eMBox Client

If you use a nonsecure connection, all the information you enter, such as user names and passwords, is sent over the wire in clear text.

If you instead want to establish a secure connection using SSL, do the following:

- ◆ Make sure you don't use the `-n` option in your command when logging in to a server. It specifies a nonsecure connection. A secure connection is the default.
- ◆ Make sure you have the following Java Secure Socket Extension (JSSE) files in your class path:
 - ◆ `jsse.jar`
 - ◆ `jnet.jar`
 - ◆ `jcrt.jar`

If you don't, the eMBox Client will return an error saying that it cannot establish a secure connection.

You can get these files and information about JSSE from the [Sun Web site \(http://java.sun.com/products/jsse\)](http://java.sun.com/products/jsse).

Finding Out eDirectory Port Numbers

When logging in to a server in the eMBox Client, you must specify a port number.

If you specified a port number when you installed eDirectory, use that number.

The default ports are as follows:

- ◆ For NetWare, the default nonsecure port is 8008, and the default secure port is 8009.
- ◆ For other platforms, the default nonsecure port is 8008, and the default secure port is 8010.

The following sections give some additional tips for finding out the port that is assigned to eDirectory:

- ◆ [“On Windows” on page 473](#)
- ◆ [“On NetWare” on page 474](#)
- ◆ [“On UNIX” on page 474](#)

On Windows

- 1** Click Start > Settings > Control Panel.
- 2** Double-click the Novell eDirectory Services icon, then click the Transport tab.
- 3** Look up the secure or nonsecure port.
 - ◆ For the nonsecure port, click the plus sign next to HTTP.
 - ◆ For the secure port, click the plus sign next to HTTPS.

Click the plus sign next to Bound Transports to see the port number.

On NetWare

The Network Address property of a Server object will show you the ports.

You can look up the Network Address property of a server object in the following tools:

- ◆ In iManager, look at the server object using eDirectory Administration > Modify Object, and on the General tab read the drop down list for Network address.
- ◆ In ConsoleOne[®], right-click the server object or select it and click Object > Properties, and look for the Network Addresses drop-down list.

Look for the network addresses that begin with http: or https: and have “/portal” at the end. These are the nonsecure and secure ports used for eMBox tools.

Here’s how to tell what the port number is:

- ◆ If a port number is displayed in the network address, that is the port number that has been assigned.

For example, `http://137.65.188.1:8008/portal` means that port 8008 is being used for eMBox tools.

- ◆ If a portal number is not displayed, and you see only the IP address for the server, that means the default port numbers are being used.

For example, `https://137.65.188.1/portal` is displaying no port number after the IP address, which means that the default secure portal number is being used for eMBox tools: 8009 on NetWare, 8010 on other platforms.

On UNIX

You can use this command to see a list of ports:

```
ndsconfig get | grep http
```

Look for the lines that say `http.server.interface` and then a port number.

You can also look up the port number in iManager or ConsoleOne using the same method described in [“On NetWare” on page 474](#).

Using the eMBox Logger

The eMBox Logger is an infrastructure module that logs all the events for all the eDirectory Management Tools (eMTools) such as DSBackup, DSMerge, and DSRepair. In this release, only one log file is provided in which all eMTools log their operations.

The eMBox Logger is different than the client logging service, which is provided through the log files that you specify when you run the eMBox client (for example, when you specify `-l mylogfile.txt` in an eMBox client command or when you enter `mylogfile.txt` as a log file name in iManager). The eMBox Logger currently records all server messages for tasks that are performed by the eMBox, showing greater detail. By contrast, the client logging service records client messages and messages sent to the client, which give a general report of progress.

Logging is asynchronous, and all operations are logged by default.

This release of the eMBox Logger provides the following features:

- ◆ The ability to change the log file name and location.

By default, log files are created in the `embox\log` directory located in the same directory that eDirectory was installed in.

- ◆ The ability to change the maximum file size, after which the log file will reset.

The maximum file size is 8 MB.

- ◆ The ability to change the logging mode.

You can choose to append all new messages to the log file or to overwrite an existing log file. The Append option is set by default.

- ◆ The ability to start and stop the logging.

By default, the logger is in Start mode when the eMBox starts up. While in Stop mode, no messages are logged.

- ◆ The ability to reset the log file contents.
- ◆ The ability to read the log file from a client machine.

In This Section:


- ◆ [“Using the eMBox Logger Command Line Client” on page 475](#)
- ◆ [“Using the eMBox Logger Feature in Novell iManager” on page 476](#)

Using the eMBox Logger Command Line Client

The following table lists the eMBox Logger command line client options:

Option	Description
<code>logstart</code>	Starts the eMBox logger.
<code>logstop</code>	Stops the eMBox logger.
<code>readlog</code>	Displays the current log file.
<code>getlogstate</code>	Displays the current state of the eMBox logger (Start/Stop).
<code>getloginfo</code>	Displays the name, logging mode(Append/Overwrite), maximum size and the current size of the eMBox log file.
<code>setloginfo [-f filename] [-s size in Kilo bytes] [-a -o]</code>	Lets you set the name, size, and logging mode (Append/Overwrite) of the eMBox log file using the following parameters: <ul style="list-style-type: none"> ◆ <code>-f filename</code> The eMBox log file name. ◆ <code>-s size in KB</code> The maximum size of the log file. ◆ <code>-a</code> New log messages will be appended to the current one. ◆ <code>-o</code> The log file will be overwritten.
<code>emptylog</code>	Clears the contents of the server log file.

Using the eMBox Logger Feature in Novell iManager

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance Utilities > Log Files.
- 3** Specify which server will perform the log file operation, then click Next.
- 4** Authenticate to the server, then click Next.
- 5** Select the log file operation to be performed.
Click Help for details.

18 Troubleshooting Novell eDirectory

The following sections give suggestions and resources for solving issues with Novell® eDirectory™ software:

- ♦ “Resolving Error Codes” on page 477
- ♦ “Naming Objects” on page 477
- ♦ “Determining the eDirectory Version Number” on page 478
- ♦ “Recovering from eDirectory Replica Problems” on page 481
- ♦ “The eDirectory for Windows Server Won't Start” on page 482
- ♦ “The Windows Server Can't Open the eDirectory Database Files” on page 482
- ♦ “Restoring eDirectory on Windows after an Emergency Repair” on page 483
- ♦ “Log Files” on page 483
- ♦ “Troubleshooting LDIF Files” on page 484
- ♦ “Troubleshooting eDirectory on Linux, Solaris, AIX, and HP-UX” on page 500
- ♦ “Troubleshooting ConsoleOne” on page 519
- ♦ “Obituaries” on page 519
- ♦ “Accessing HTTPSTK When DS Is Not Loaded” on page 525
- ♦ “Troubleshooting SNMP” on page 527
- ♦ “eDirectory Install Fails for Container Administrators” on page 529
- ♦ “Migrating the Sun ONE Schema to Novell eDirectory” on page 530
- ♦ “Migrating the Active Directory Schema to Novell eDirectory Using ICE” on page 533

Resolving Error Codes

For a complete list and explanation of eDirectory error codes, see the [Novell Error Codes Web page \(http://www.novell.com/documentation/lg/nwec/index.html\)](http://www.novell.com/documentation/lg/nwec/index.html).

Naming Objects

When you use special characters while naming objects, the -671 No Such Parent error message appears. Avoid using any of the following special characters when naming objects:

\\, * ? .

Determining the eDirectory Version Number

The following sections list ways you can determine the version of eDirectory installed on a server:

- ◆ “NetWare” on page 478
- ◆ “Windows” on page 479
- ◆ “Linux” on page 479
- ◆ “Solaris” on page 480
- ◆ “AIX” on page 480
- ◆ “HP-UX” on page 481

NetWare

- ◆ Run `ds.nlm`, or any other `.nlm`.

At the server console, enter `ds.nlm`. This displays both the marketing string (for example, Novell eDirectory 8.7) and the internal build number (for example, DS v10410.xx).

- ◆ Run iMonitor.

On the Agent Summary page, click Known Servers. Then under Servers Known to Database, click Known Servers. The Agent Revision column displays the internal build number for each server. For example, an Agent Revision number for Novell eDirectory 8.7.1 might be 10510.64.

For information on running iMonitor, see “[Accessing iMonitor](#)” on page 165.

- ◆ Run `nwconfig`.

At the server console, enter `nwconfig`, then select Product Info. If the install registered eDirectory (which it does with NDS[®] eDirectory 8.5 and later), this will display what was registered during the install. This is typically a hybrid of marketing and build numbers. For example, you might see marketing version eDir 8.5 and build version 85.01.

- ◆ LDAP shows configuration through DSTrace.

This is true for most utilities (for example, DSRepair or DSMerge) as they load. This method will display the internal build number.

For more information on DSTrace, see [Looking Into the Directory Services Trace \(DSTrace\) Options](#) (<http://developer.novell.com/research/sections/netmanage/dirprimer/2001/august/spv.htm>) and [More on Using the DSTrace Command](#) (<http://developer.novell.com/research/sections/netmanage/dirprimer/2001/septembe/p010901.htm>).

- ◆ Read the eDirectory download filename.

The eDirectory download filename usually matches the marketing string. For example, the download filename for Novell eDirectory 8.7.1 is `edir871.exe`.

- ◆ Enter `version` at a console prompt.

This will display the eDirectory version.

Windows

- ◆ Run iMonitor.

On the Agent Summary page, click Known Servers. Then under Servers Known to Database, click Known Servers. The Agent Revision column displays the internal build number for each server. For example, an Agent Revision number for Novell eDirectory 8.7.1 might be 10510.64.

For information on running iMonitor, see [“Accessing iMonitor” on page 165](#).

- ◆ Run NDSCons.exe.

In the Windows Control Panel, double-click Novell eDirectory Services. In the Services column, select ds.dlm, then click Configure. The Agent tabs displays both the marketing string (for example, Novell eDirectory 8.7.1) and the internal build number (for example, 10510.64).

- ◆ Run an eDirectory utility.

Most eDirectory utilities have an About option on their Help menu that displays the version number of the utility (for example, Merge Graft Utility 10510.35). Some utilities include the internal build version in the main label of the utility (for example, DSRepair - Version 10510.37).

To load an eDirectory utility (such as DSMerge or DSRepair), double-click Novell eDirectory Services in the Windows Control Panel. In the Services column, select the utility, then click Start.

- ◆ View the properties of an eDirectory .dlm file.

Right-click the .dlm in Windows Explorer, then click the Version tab in the Properties dialog box. This will display the version number of the utility. The default location for eDirectory .dlm files is C:\novell\NDS.

Linux

- ◆ Run ndsstat.

The ndsstat utility displays information related to eDirectory servers, such as the eDirectory tree name, the fully distinguished server name, and the eDirectory version. In the following example, eDirectory 8.7.1 is the product version (marketing string), and 10510.65 is the binary version (internal build number).

```
osg-dt-srv17: />ndsstat
Tree Name: SNMP-HPUX-RASH
Server Name: .CN=osg-dt-srv17.O=novell.T=SNMP-HPUX-RASH.
Binary Version: 10510.65
Root Most Entry Depth: 0
Product Version: NDS/Unix - NDS eDirectory v8.7.1 [DS]
```

For information on running ndsstat, see [Appendix B, “Novell eDirectory UNIX Commands and Usage,” on page 541](#), or the ndsstat man page (ndsstat.1m).

- ◆ Run ndsd --version.

For information on running ndsd, see [Appendix B, “Novell eDirectory UNIX Commands and Usage,” on page 541](#), or the ndsd man page (ndsd.1m).

- ◆ Run iMonitor.

On the Agent Summary page, click Known Servers. Then under Servers Known to Database, click Known Servers. The Agent Revision column displays the internal build number for each

server. For example, an Agent Revision number for Novell eDirectory 8.7.1 might be 10510.64.

For information on running iMonitor, see [“Accessing iMonitor” on page 165](#).

- ◆ Run `rpm -qi NDSserv`.

Entering this command will display similar information to `nds --version`.

Solaris

- ◆ Run `ndsstat`.

The `ndsstat` utility displays information related to eDirectory servers, such as the eDirectory tree name, the fully distinguished server name, and the eDirectory version. In the following example, eDirectory 8.7.1 is the product version (marketing string), and 10510.65 is the binary version (internal build number).

```
osg-dt-srv17: />ndsstat
Tree Name: SNMP-HPUX-RASH
Server Name: .CN=osg-dt-srv17.O=novell.T=SNMP-HPUX-RASH.
Binary Version: 10510.65
Root Most Entry Depth: 0
Product Version: NDS/Unix - NDS eDirectory v8.7.1 [DS]
```

For information on running `ndsstat`, see [Appendix B, “Novell eDirectory UNIX Commands and Usage,” on page 541](#), or the `ndsstat` man page (`ndsstat.1m`).

- ◆ Run `nds --version`.

For information on running `nds`, see [Appendix B, “Novell eDirectory UNIX Commands and Usage,” on page 541](#), or the `nds` man page (`nds.1m`).

- ◆ Run iMonitor.

On the Agent Summary page, click Known Servers. Then under Servers Known to Database, click Known Servers. The Agent Revision column displays the internal build number for each server. For example, an Agent Revision number for Novell eDirectory 8.7.1 might be 10510.64.

For information on running iMonitor, see [“Accessing iMonitor” on page 165](#).

- ◆ Run `pkginfo -l NDSserv`.

Entering this command will display similar information to `nds --version`.

AIX

- ◆ Run `ndsstat`.

The `ndsstat` utility displays information related to eDirectory servers, such as the eDirectory tree name, the fully distinguished server name, and the eDirectory version. In the following example, eDirectory 8.7.1 is the product version (marketing string), and 10510.65 is the binary version (internal build number).

```
osg-dt-srv17: />ndsstat
Tree Name: SNMP-HPUX-RASH
Server Name: .CN=osg-dt-srv17.O=novell.T=SNMP-HPUX-RASH.
Binary Version: 10510.65
Root Most Entry Depth: 0
Product Version: NDS/Unix - NDS eDirectory v8.7.1 [DS]
```


For information on running `ndsstat`, see [Appendix B, “Novell eDirectory UNIX Commands and Usage,” on page 541](#), or the `ndsstat` man page (`ndsstat.1m`).

- ◆ Run `ndsd --version`.

For information on running `ndsd`, see [Appendix B, “Novell eDirectory UNIX Commands and Usage,” on page 541](#), or the `ndsd` man page (`ndsd.1m`).

- ◆ Run `iMonitor`.

On the Agent Summary page, click **Known Servers**. Then under **Servers Known to Database**, click **Known Servers**. The Agent Revision column displays the internal build number for each server. For example, an Agent Revision number for Novell eDirectory 8.7.1 might be 10510.64.

For information on running `iMonitor`, see [“Accessing iMonitor” on page 165](#).

HP-UX

- ◆ Run `ndsstat`.

The `ndsstat` utility displays information related to eDirectory servers, such as the eDirectory tree name, the fully distinguished server name, and the eDirectory version. In the following example, eDirectory 8.7.1 is the product version (marketing string), and 10510.65 is the binary version (internal build number).

```
osg-dt-srv17: />ndsstat
Tree Name: SNMP-HPUX-RASH
Server Name: .CN=osg-dt-srv17.O=novell.T=SNMP-HPUX-RASH.
Binary Version: 10510.65
Root Most Entry Depth: 0
Product Version: NDS/Unix - NDS eDirectory v8.7.1 [DS]
```

For information on running `ndsstat`, see [Appendix B, “Novell eDirectory UNIX Commands and Usage,” on page 541](#), or the `ndsstat` man page (`ndsstat.1m`).

- ◆ Run `ndsd --version`.

For information on running `ndsd`, see [Appendix B, “Novell eDirectory UNIX Commands and Usage,” on page 541](#), or the `ndsd` man page (`ndsd.1m`).

- ◆ Run `iMonitor`.

On the Agent Summary page, click **Known Servers**. Then under **Servers Known to Database**, click **Known Servers**. The Agent Revision column displays the internal build number for each server. For example, an Agent Revision number for Novell eDirectory 8.7.1 might be 10510.64.

For information on running `iMonitor`, see [“Accessing iMonitor” on page 165](#).

Recovering from eDirectory Replica Problems

eDirectory offers the Novell robust directory service and the fault tolerance inherent in replication. Replication allows you to keep copies of the eDirectory database, or portions of it, on multiple servers at once.

You should always keep multiple replicas of eDirectory partitions. If you do so and one replica becomes corrupted or is lost because of a failed hard disk, you can delete that replica using `ConsoleOne`[®] or `Novell iManager` and replace it with a new one from the intact replica.

For more information on deleting replicas, see [“Deleting a Replica” on page 118](#).

The eDirectory for Windows Server Won't Start

If the eDirectory server fails to start when you boot the Windows server, a message will notify you that the service failed to start.

If there are no other eDirectory database replicas, users can't log in.

If there are other replicas, logging in might be slow and you will see communication errors and synchronization errors on the servers holding those replicas.

- ◆ The eDirectory server entries in the Windows Registry might have been edited, or the Windows Registry might be corrupt.
- ◆ eDirectory database files might have been corrupted or deleted.
- ◆ If the eDirectory server can't start because another service didn't start, you can get more information from Start > Programs > Administrative Tools > Event Viewer.

You'll need to resolve the related-service problem before starting the eDirectory server.

- ◆ The Registry or eDirectory executable files are corrupted or lost. Run the SAMMIG utility in the system directory. Select Uninstall NDS on Windows NT and include new eDirectory information in the NT domain. Continue with the Uninstall until completed. Then restart sammig.exe and proceed to install eDirectory.
- ◆ Database files have been corrupted or deleted. If the eDirectory server comes up on the NT server but the service can't open the eDirectory database files, see [“The Windows Server Can't Open the eDirectory Database Files” on page 482](#).
- ◆ The eDirectory server is not connected to a hub or switch or directly to a workstation (using a crossover cable). Connect the server to a hub or switch.

The Windows Server Can't Open the eDirectory Database Files

If the eDirectory server can't open the database files, a message on the Windows server will notify you.

If there are no other database replicas, users can't log in.

If there are other replicas, logging in might be slow and you will see communication errors and synchronization errors on the servers holding those replicas.

- ◆ The database files might have been corrupted through disk errors on the NT/2000 server.
- ◆ Someone might have deleted one or more of the database files.

If other replicas of the eDirectory database exist, complete the following steps:

- 1** Start Novell iManager from an administrative workstation.
- 2** Remove the corrupted replica from the replica ring.
See [“Deleting a Replica” on page 118](#) for more information.
- 3** Run the sammig.exe utility in the system directory (usually c:\winnt\system32) on the NT server or from the Start menu (Start > Programs > Administrative Tools (Common) > Migration Tool for NetWare).
- 4** Select the option to create a new replica on the eDirectory server.

If this eDirectory server holds the only replica of the partition, complete the following steps:

- 1 Run the `sammig.exe` utility in the system directory (usually `c:\winnt\system32`) on the NT Server or from the Start menu (Start > Programs > Administrative Tools (Common) > Migration Tool for NetWare).
- 2 Select Uninstall NDS on Windows and revert to the previous Windows domain state.
- 3 Continue with the Uninstall until it has completed.
- 4 Restart the Migration Tool for NetWare and proceed to install eDirectory on Windows.
- 5 Move the User objects from the NT/2000 domain to the eDirectory tree.

Restoring eDirectory on Windows after an Emergency Repair

When you are forced to do an emergency repair on a Windows server and there is no Emergency Repair disk, or the Emergency Repair disk was created before an eDirectory installation, the eDirectory client is removed and Registry settings are deleted. The `nds4nter.exe` utility both restores the necessary Registry settings and reloads eDirectory files.

Run `nds4nter.exe` from the `\i386\goodies` directory.

After an emergency repair is performed, run the Emergency Repair utility from the CD. The utility will first restore some of the Registry settings, then it will launch the eDirectory installation. The installation will copy the files then you must select the reboot option. After rebooting, users will have access to the migrated domains.

Log Files

This section contains information on the following log files stored in the local replica directory (*path*\DIBFiles):

- ♦ [“modschema.log” on page 483](#)
- ♦ [“dsinstall.log” on page 483](#)

modschema.log

The `modschema.log` file contains the results of all schema extensions that are applied when an eDirectory server is installed into an existing tree. Each line of the log states which class or attribute is being added or modified and gives the status of the modification attempt.

This log is created or overwritten each time the install process is run, so it only represents the results of the last attempt. In addition to the eDirectory schema extensions, this log contains the results of any other schema extensions (such as LDAP or SAS) applied by the DSINSTALL front end prior to adding the new eDirectory server.

This log will not be generated when a standalone server is installed or if the version of the target server is NDS 7.01 or later.

dsinstall.log

The first part of the log lists environment variables that are set. The second part contains status messages documenting the eDirectory installation process.

Troubleshooting LDIF Files

The Novell Import Conversion Export utility lets you easily import LDIF files into and export LDIF files from eDirectory. For more information, see [“Novell Import Conversion Export Utility” on page 125](#).

In order for an LDIF import to work properly, you must start with an LDIF file that the Novell Import Conversion Export utility can read and process. This section describes the LDIF file format and syntax and provides examples of correct LDIF files.

Understanding LDIF

LDIF is a widely used file format that describes directory information or modification operations that can be performed on a directory. LDIF is completely independent of the storage format used within any specific directory implementation, and is typically used to export directory information from and import data to LDAP servers.

LDIF is usually easy to generate. This makes it possible to use tools like awk or perl to move data from a proprietary format into an LDAP directory. You can also write scripts to generate test data in LDIF format.

LDIF File Format

Novell Import Conversion Export imports require LDIF 1 formatted files. The following are the basic rules for an LDIF 1 file:

- ◆ The first noncomment line must be version: 1.
- ◆ A series of one or more records follows the version.
- ◆ Each record is composed of fields, one field per line.
- ◆ Lines are separated by either a new line or a carriage return/new line pair.
- ◆ Records are separated by one or more blank lines.
- ◆ There are two distinct types of LDIF records: content records and change records. An LDIF file can contain an unlimited number of records, but they all must be of the same type. You can't mix content records and change records in the same LDIF file.
- ◆ Any line beginning with the pound sign (#) is a comment and is ignored when processing the LDIF file.

LDIF Content Records

An LDIF content record represents the contents of an entire entry. The following is an example of an LDIF file with four content records:

```
1 version: 1
2 dn: c=US
3 objectClass: top
4 objectClass: country
5
6 dn: l=San Francisco, c=US
7 objectClass: top
8 objectClass: locality
9 st: San Francisco
10
11 dn: ou=Artists, l=San Francisco, c=US
```

```

12 objectClass: top
13 objectClass: organizationalUnit
14 telephoneNumber: +1 415 555 0000
15
16 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US
17 sn: Michaels
18 givenname: Peter
19 objectClass: top
20 objectClass: person
21 objectClass: organizationalPerson
22 objectClass: inetOrgPerson
23 telephonenumber: +1 415 555 0001
24 mail: Peter.Michaels@aaa.com
25 userpassword: Peter123
26

```

This LDIF file is composed of the following parts:

Component	Description
Version Specifier	<p>The first line of an LDIF file contains the version. Zero or more spaces are allowed between the colon and the version number, which is currently defined to be 1.</p> <p>If the version line is missing, any application processing the LDIF file is allowed to assume that the file is version 0. It's also possible that the LDIF file could be rejected as syntactically incorrect. Novell utilities that process LDIF assume a file version of 0 when the version line is missing.</p>
Distinguished Name Specifier	<p>The first line of every content record (lines 2, 6, 11, and 16 in the example above) specifies the DN of the entry that it represents.</p> <p>The DN specifier must take one of the following two forms:</p> <ul style="list-style-type: none"> ◆ dn: <i>safe_UTF-8_distinguished_name</i> ◆ dn:: <i>Base64_encoded_distinguished_name</i>
Line Delimiters	<p>The line separator can be either a line feed or a carriage return/line feed pair. This resolves a common incompatibility between Linux and Solaris text files, which use a line feed as the line separator, and MS-DOS* and Windows text files, which use a carriage return/line feed pair as the line separator.</p>
Record Delimiters	<p>Blank lines (lines 5, 10, 15, and 26 in the example above) are used as record delimiters.</p> <p>Every record in an LDIF file including the last record must be terminated with a record delimiter (one or more blank lines). Although some implementations will silently accept an LDIF file without a terminating record delimiter, the LDIF specification requires it.</p>
Attribute Value Specifier	<p>All other lines in a content records are value specifiers. Value specifiers must take on one of the following three forms:</p> <ul style="list-style-type: none"> ◆ Attribute description: <i>value</i> ◆ Attribute description:: <i>Base64_encoded_value</i> ◆ Attribute description: < <i>URL</i>

LDIF Change Records

LDIF change records contain modifications to be made to a directory. Any of the LDAP update operations (add, delete, modify, and modify DN) can be represented in an LDIF change record.

LDIF change records use the same format for the distinguished name specifier, attribute value specifier, and record delimiter as LDIF content records. (See [“LDIF Content Records” on page 484](#) for more information.) The presence of a changetype field is what distinguishes an LDIF change record from an LDIF content record. A changetype field identifies the operation specified by the change record.

A changetype field can take one of the following five forms:

Form	Description
changetype: add	A keyword indicating that the change record specifies an LDAP add operation.
changetype: delete	A keyword indicating that the change record specifies an LDAP delete operation.
changetype: moddn	A keyword indicating that the change record specifies an LDAP modify DN operation if the LDIF processor is bound to the LDAP server as a version 3 client or a modify RDN operation if the LDIF processor is bound to the LDAP server as a version 2 client.
changetype: modrdn	A synonym for the moddn change type.
changetype: modify	A keyword indicating that the change record specifies an LDAP modify operation.

The Add Change Type

An add change record looks just like a content change record (see [“LDIF Content Records” on page 484](#)) with the addition of the changetype: add field immediately before any attribute value fields.

All records must be the same type. You can't mix content records and change records.

```

1 version: 1
2 dn: c=US
3 changetype: add
4 objectClass: top
5 objectClass: country
6
7 dn: l=San Francisco, c=US
8 changetype: add
9 objectClass: top
10 objectClass: locality
11 st: San Francisco
12
14 dn: ou=Artists, l=San Francisco, c=US
15 changetype: add
16 objectClass: top
17 objectClass: organizationalUnit
18 telephoneNumber: +1 415 555 0000
19
20 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US
21 changetype: add

```

```

22 sn: Michaels
23 givenname: Peter
24 objectClass: top
25 objectClass: person
26 objectClass: organizationalPerson
27 objectClass: iNetOrgPerson
28 telephonenumber: +1 415 555 0001
29 mail: Peter.Michaels@aaa.com
30 userpassword: Peter123
31

```

The Delete Change Type

Because a delete change record specifies the deletion of an entry, the only fields required for a delete change record are the distinguished name specifier and a delete change type.

The following is an example of an LDIF file used to delete the four entries created by the LDIF file shown in [“The Add Change Type” on page 486](#).

IMPORTANT: To delete entries you have previously added, reverse the order of the entries. If you don't do this, the delete operation fails because the container entries are not empty.

```

1 version: 1
2 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US
3 changetype: delete
4
5 dn: ou=Artists, l=San Francisco, c=US
8 changetype: delete
9
10 dn: l=San Francisco, c=US
11 changetype: delete
12
13 dn: c=US
14 changetype: delete
15

```

The Modify Change Type

The modify change type lets you to specify the addition, deletion, and replacement of attribute values for an entry that already exists. Modifications take one of the following three forms:

Element	Description
add: attribute type	A keyword indicating that subsequent attribute value specifiers for the attribute type should be added to the entry.
delete: attribute type	A keyword indicating that values of the attribute type are to be deleted. If attribute value specifiers follow the delete field, the values given are deleted. If no attribute value specifiers follow the delete field, then all values are deleted. If the attribute has no values, this operation will fail, but the desired effect will still be achieved because the attribute had no values to be deleted.

Element	Description
replace: attribute type	<p>A keyword indicating that the values of the attribute type are to be replaced. Any attribute value specifiers that follow the replace field become the new values for the attribute type.</p> <p>If no attribute value specifiers follow the replace field, the current set of values is replaced with an empty set of values (which causes the attribute to be removed). Unlike the delete modification specifier, if the attribute has no values, the replace will still succeed. The net effect in both cases is the same.</p>

The following is an example of a modify change type that will add an additional telephone number to the cn=Peter Michaels entry.

```

1 version: 1
2 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US
3 changetype: modify
4 # add the telephone number to cn=Peter Michaels
4 add: telephonenumber
5 telephonenumber: +1 415 555 0002
6

```

Just as you can combine a mixture of modifications in a single LDAP modify request, you can specify multiple modifications in a single LDIF record. A line containing only the hyphen (-) character is used to mark the end of the attribute value specifications for each modification specifier.

The following example LDIF file contains a mixture of modifications:

```

1 version: 1
2
3 # An empty line to demonstrate that one or more
4 # line separators between the version identifier
5 # and the first record is legal.
6
7 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US
8 changetype: modify
9 # Add an additional telephone number value.
10 add: telephonenumber
11 telephonenumber: +1 415 555 0002
12 -
13 # Delete the entire facsimiletelephonenumber attribute.
14 delete: facsimileTelephoneNumber
15 -
16 # Replace the existing description (if any exists)
17 # with two new values.
18 replace: description
19 description: guitar player
20 description: solo performer
21 -
22 # Delete a specific value from the telephonenumber
23 # attribute.
24 delete: telephonenumber
25 telephonenumber: +1 415 555 0001
26 -
27 # Replace the existing title attribute with an empty
28 # set of values, thereby causing the title attribute to

```



```

29 # be removed.
30 replace: title
31 -
32

```

The Modify DN Change Type

The modify DN change type lets you rename an entry, move it, or both. This change type is composed of two required fields and one optional field.

Field	Description
newrdn (required)	<p>Gives the new name for the entry that will be assigned while processing this record. The new RDN specifier must take of the following two forms:</p> <ul style="list-style-type: none"> ♦ newrdn: <i>safe_UTF-8_relative_distinguished_name</i> ♦ newrdn:: <i>Base64_encoded_relative_distinguished_name</i> <p>The new RDN specifier is required in all LDIF records with a modify DN change type.</p>
deleteoldrdn (required)	<p>The delete old RDN specifier is a flag that indicates whether the old RDN should be replaced by the newrdn or if it should be kept. It takes one of the two following forms:</p> <ul style="list-style-type: none"> ♦ deleteoldrdn: 0 Indicates that the old RDN value should be kept in the entry after it is renamed. ♦ deleteoldrdn: 1 Indicates that the old RDN value should be deleted when the entry is renamed.
newsuperior (optional)	<p>The new superior specifier gives the name of the new parent that will be assigned to the entry while processing the modify DN record. The new superior specifier must take of the following two forms:</p> <ul style="list-style-type: none"> ♦ newsuperior: <i>safe_UTF-8_distinguished_name</i> ♦ newsuperior:: <i>Base64_encoded_distinguished_name</i> <p>The new superior specifier is optional in LDIF records with a modify DN change type. It is only given in cases where you want to reparent the entry.</p>

The following is an example of a modify DN change type that shows how to rename an entry:

```

1 version: 1
2
3 # Rename ou=Artists to ou=West Coast Artists, and leave
4 # its old RDN value.
5 dn: ou=Artists,l=San Francisco,c=US
6 changetype: moddn
7 newrdn: ou=West Coast Artists
8 deleteoldrdn: 1
9

```

The following is an example of a modify DN change type that shows how to move an entry:

```

1 version: 1
2

```

```

3 # Move cn=Peter Michaels from
4 # ou=Artists,l=San Francisco,c=US to
5 # ou=Promotion,l=New York,c=US and delete the old RDN.
5 dn: cn=Peter Michaels,ou=Artists,l=San Francisco,c=US
6 changetype: moddn
7 newrdn: cn=Peter Michaels
8 deleteoldrdn: 1
9 newsuperior: ou=Promotion,l=New York,c=US
10

```

The following is an example of a modify DN change type that shows how to move an entry and rename it at the same time:

```

1 version: 1
2
3 # Move ou=Promotion from l=New York,c=US to
4 # l=San Francisco,c=US and rename it to
5 # ou=National Promotion.
5 dn: ou=Promotion,l=New York,c=US
6 changetype: moddn
7 newrdn: ou=National Promotion
8 deleteoldrdn: 1
9 newsuperior: l=San Francisco,c=US
10

```

IMPORTANT: The LDAP 2 modify RDN operation doesn't support moving entries. If you try to move an entry using the LDIF newsuperior syntax with an LDAP 2 client, the request will fail.

Line Folding within LDIF Files

To fold a line in an LDIF file, simply insert a line separator (a newline or a carriage return/newline pair) followed by a space at the place where you want the line folded. When the LDIF parser encounters a space at a beginning of the line, it knows to concatenate the rest of the data on the line with the data on the previous line. The leading space is then discarded.

You should not fold lines in the middle of a multibyte UTF-8 character.

The following is an example of an LDIF file with a folded line (see lines 13 and 14):

```

1 version: 1
2 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US
3 sn: Michaels
4 givenname: Peter
5 objectClass: top
6 objectClass: person
7 objectClass: organizationalPerson
8 objectClass: iNetOrgPerson
9 telephonenumber: +1 415 555 0001
10 mail: Peter.Michaels@aaa.com
11 userpassword: Peter123
12 description: Peter is one of the most popular music
13   ians recording on our label. He's a big concert dr
14   aw, and his fans adore him.
15

```

Hashed Password Representation in LDIF Files

The hashed password is represented as base64 data in the LDIF file. The attribute name *userpassword* should be followed with the name of the encryption used for hashing the password. This name should be given within a pair of flower brackets “{ }” as shown below:

Example 1

For SHA hashed passwords:

```
1 version: 1
2 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US
3 sn: Michaels
4 userpassword: {SHA}xcbdh46ngh37jsd0naSFDedjAS30dm
5 objectclass: inetOrgPerson
```

Example 2

For SSHA hashed passwords:

```
1 version: 1
2 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US
3 sn: Michaels
4 userpassword: {SSHA}sGs948DFGkakdfkasDF34DF4dS3sk15DFS
5 objectclass: inetOrgPerson
```

Example 3

For Digest MD5 hashed passwords:

```
1 version: 1
2 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US
3 sn: Michaels
4 userpassword: {MD5}a451kSDF234SDFG62dsfsf2DG2QEvqdmnk430
5 objectclass: inetOrgPerson
```

Debugging LDIF Files

If you have problems with an LDIF file, consider the following:

- ◆ [“Enabling Forward References” on page 491](#)
- ◆ [“Checking the Syntax of LDIF Files” on page 494](#)
- ◆ [“Using the LDIF Error File” on page 495](#)
- ◆ [“Using LDAP SDK Debugging Flags” on page 495](#)

Enabling Forward References

You might occasionally encounter LDIF files in which a record to add one entry comes before a record to add its parents. When this happens, an error is generated because the new entry's parent does not exist when the LDAP server attempts to add the entry.

To solve this problem, simply enable the use of forward references. When you enable the creation of forward references and an entry is going to be created before its parent exists, a placeholder called a forward reference is created for the entry's parent to allow the entry to be successfully created. If a later operation creates the parent, the forward reference is changed into a normal entry.

It is possible that one or more forward references will remain after your LDIF import is complete (if, for example, the LDIF file never created the parent for an entry). In this case, the forward reference will appear as an Unknown object in ConsoleOne and iManager. Although you can search on a forward reference entry, you cannot read attributes (except objectClass) from the forward reference entry because it does not have any attributes or attribute values. However, all LDAP operations will work normally on the real object entries located below the forward reference.

Identifying Forward Reference Entries


Forward reference entries have an object class of Unknown and also have their internal NDS EF_REFERENCE entry flag set. In ConsoleOne and iManager, entries with an object class of Unknown are represented by a round yellow icon with a question mark in the center. You can use LDAP to search for objects with an Unknown object class, although there is currently no way to access the entry flag settings through LDAP to be sure that they are forward reference entries.

Changing Forward Reference Entries into Normal Objects

You can change a forward reference entry into a normal object by simply creating it (using, for example, an LDIF file or an LDAP client request). When you ask eDirectory to create an entry that already exists as a forward reference, eDirectory transforms the existing forward reference entry into the object you asked it to create.

Using the Novell eDirectory Import Convert Export Wizard


To enable forward references during an LDIF import:

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance > Import Convert Export Wizard.
- 3 Click Import Data from File on Disk, then click Next.
- 4 Select LDIF as the type of file you want to import.
- 5 Specify the name of the file containing the data you want to import, specify the appropriate options, then click Next.
- 6 Specify the LDAP server where the data will be imported.
- 7 Add the appropriate options, as described in the following table:

Option	Description
Server DNS name/IP address	DNS name or IP address of the destination LDAP server
Port	Integer port number of the destination LDAP server
DER File	Name of the DER file containing a server key used for SSL authentication
Login method	Authenticated Login or Anonymous Login (for the entry specified in the User DN field)
User DN	Distinguished name of the entry that should be used when binding to the server-specified bind operation
Password	Password attribute of the entry specified in the User DN field

- 8** Under Advanced Settings, click Allow Forward References.
- 9** Click Next, then click Finish.

To enable forward references during a data-to-data server migration:

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Click eDirectory Maintenance > Import Convert Export Wizard.
- 3** Click Migrate Data Between Servers, then click Next.
- 4** Specify the LDAP server holding the entries you want to migrate.
- 5** Add the appropriate options, as described in the following table:

Option	Description
Server DNS name/IP address	DNS name or IP address of the source LDAP server
Port	Integer port number of the source LDAP server
DER file	Name of the DER file containing a server key used for SSL authentication
Login method	Authenticated Login or Anonymous Login (for the entry specified in the User DN field)
User DN	Distinguished name of the entry that should be used when binding to the server-specified bind operation
Password	Password attribute of the entry specified in the User DN field

- 6** Under Advanced Settings, click Allow Forward References.
- 7** Click Next.
- 8** Specify the search criteria (described below) for the entries you want to migrate:

Option	Description
Base DN	Base distinguished name for the search request If this field is left empty, the base DN defaults to " " (empty string).
Scope	Scope of the search request
Filter	RFC 2254-compliant search filter The default is objectclass=*
Attributes	Attributes you want returned for each search entry

- 9** Click Next.
- 10** Specify the LDAP server where the data will be migrated.
- 11** Click Next, then click Finish.

NOTE: Ensure that the schema is consistent across LDAP Services.

Using the Novell Import Conversion Export Utility Command Line Interface

To enable forward references in the command line interface, use the -F LDAP destination handler option.


For more information, see [“LDIF Destination Handler Options” on page 132](#).

Checking the Syntax of LDIF Files

You can check the syntax of an LDIF file before you process the records in the file by using the Display Operations But Do Not Perform LDIF source handler option.

The LDIF source handler always checks the syntax of the records in an LDIF file as it processes them. Using this option disables the processing of the records and lets you verify the syntax.

Using the Novell eDirectory Import Convert Export Wizard

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Maintenance > Import Convert Export Wizard.
- 3 Click Import Data from File on Disk, then click Next.
- 4 Select LDIF as the type of file you want to import.
- 5 Specify the name of the file containing the data you want to import, specify the appropriate options.
- 6 Under Advanced Settings, click Display Operations But Do Not Perform, then click Next.
- 7 Specify the LDAP server where the data will be imported.
- 8 Add the appropriate options, as described in the following table:

Option	Description
Server DNS name/IP address	DNS name or IP address of the destination LDAP server
Port	Integer port number of the destination LDAP server
DER File	Name of the DER file containing a server key used for SSL authentication
Login method	Authenticated Login or Anonymous Login (for the entry specified in the User DN field)
User DN	Distinguished name of the entry that should be used when binding to the server-specified bind operation
Password	Password attribute of the entry specified in the User DN field

- 9 Click Next, then click Finish.

Using the Novell Import Conversion Export Utility Command Line Interface

To check the syntax of an LDIF file in the command line interface, use the -n LDIF source handler option.

For more information, see [“LDIF Source Handler Options” on page 131](#).

Using the LDIF Error File

The Novell Import Conversion Export utility automatically creates an LDIF file listing any records that failed processing by the destination handler. You can edit the LDIF error file generated by the utility, fix the errors, then reapply it to the server to finish an import or data migration that contained failed records.

Using the Novell eDirectory Import/Export Wizard

This feature is available only in ConsoleOne.

- 1** In ConsoleOne, click Wizard > NDS Import/Export.
- 2** Click the task you want to perform.
- 3** Click Advanced.
- 4** In the Log File field, specify a filename where output messages (including error messages) will be logged.
- 5** In the LDIF Output File for Failed Records field, specify a filename where entries that fail are output in LDIF format.

You can use this file to examine or correct errors. You can also reapply a modified (corrected) version of this file to the directory.
- 6** Click Close.
- 7** Follow the online instructions to finish your selected task.

Using the Novell Import Conversion Export Utility Command Line Interface

To configure error log options in the command line utility, use the `-l` general option.

For more information, see [“General Options” on page 129](#).

Using LDAP SDK Debugging Flags

To understand some LDIF problems, you might need to see how the LDAP client SDK is functioning. You can set the following debugging flags for the LDAP source handler, the LDAP destination handler, or both.

Value	Description
0x0001	Trace LDAP function calls.
0x0002	Print information about packets.
0x0004	Print information about arguments.
0x0008	Print connections information.
0x0010	Print BER encoding and decoding information.
0x0020	Print search filter information.
0x0040	Print configuration information.
0x0080	Print ACL information.
0x0100	Print statistical information.

Value	Description
0x0200	Print additional statistical information.
0x0400	Print shell information.
0x0800	Print parsing information.
0xFFFF (-1 Decimal)	Enable all debugging options.

To enable this functionality, use the `-e` option for the LDAP source and LDAP destination handlers. The integer value you give for the `-e` option is a bitmask that enables various types of debugging information in the LDAP SDK.

For more information, see [“LDAP Source Handler Options” on page 132](#) and [“LDAP Destination Handler Options” on page 134](#).

Using LDIF to Extend the Schema

Because LDIF can represent LDAP update operations, you can use LDIF to modify the schema.

Adding a New Object Class

To add a class, simply add an attribute value that conforms to the specification for `NDSObjectClassDescription` to the `objectClasses` attribute of the `subschemaSubentry`.

```
NDSObjectClassDescription = "(" whsp
    numericoid whsp
    [ "NAME" qdescrs ]
    [ "DESC" qdstring ]
    [ "OBSOLETE" whsp ]
    [ "SUP" oids ]
    [ ( "ABSTRACT" / "STRUCTURAL" / "AUXILIARY" ) whsp ]
    [ "MUST" oids ]
    [ "MAY" oids ]
    [ "X-NDS_NOT_CONTAINER" qdstrings ]
    [ "X-NDS_NONREMOVABLE" qdstrings ]
    [ "X-NDS_CONTAINMENT" qdstrings ]
    [ "X-NDS_NAMING" qdstrings ]
    [ "X-NDS_NAME" qdstrings ]
whsp ")"
```

The following example LDIF file adds the `person` objectClass to the schema:

```
1 version: 1
2 dn: cn=schema
3 changetype: add
4 objectClasses: ( 2.5.6.6 NAME 'person' DESC 'Standard
5   ObjectClass' SUP ndsLoginProperties STRUCTURAL MUST
6   (cn $ sn) MAY (description $ seeAlso $ telephoneNum
7   ber $ fullName $ givenName $ initials $ uid $ userPa
8   ssword) X-NDS_NAMING ('cn' 'uid') X-NDS_CONTAINMENT
9   ('organization' 'organizationalUnit' 'domain') X-NDS
10  _NAME 'Person' X-NDS_NOT_CONTAINER '1' X-NDS_NONREMO
11  VABLE '1')
12
```


Mandatory Attributes

Mandatory attributes are listed in the MUST section of the object class description. For the person object class, the mandatory attributes are cn and sn.

Optional Attributes

Optional attributes are listed in the MAY section of the object class description. The optional attributes in the person object class are description, seeAlso, telephoneNumber, fullName, givenName, initials, uid, and userPassword.

NOTE: The userPassword attribute cannot be used as an optional (MAY) attribute. The operation will fail if you try to use it as a mandatory (MUST) attribute in the new objectClass using this LDIF format to extend the schema.

Containment Rules

The object classes that can contain the object class being defined are given in the X-NDS_CONTAINMENT section of the object class description. The person object class can be contained by the organization, organizationalUnit, and domain object classes.

Adding a New Attribute

To add an attribute, simply add an attribute value that conforms to the specification for NDSAttributeTypeDescription to the attributes attribute of the subschemaSubentry.

```
NDSAttributeTypeDescription = "(" whsp
  numericoid whsp ; AttributeType identifier
  [ "NAME" qdescrs ] ; name used in AttributeType
  [ "DESC" qdstring ] ; description
  [ "OBSOLETE" whsp ]
  [ "SUP" woid ] ; derived from this other AttributeType
  [ "EQUALITY" woid ] ; Matching Rule name
  [ "ORDERING" woid ] ; Matching Rule name
  [ "SUBSTR" woid ] ; Matching Rule name
  [ "SYNTAX" whsp noidlen whsp ] ; Syntax OID
  [ "SINGLE-VALUE" whsp ] ; default multi-valued
  [ "COLLECTIVE" whsp ] ; default not collective
  [ "NO-USER-MODIFICATION" whsp ] ; default user modifiable
  [ "USAGE" whsp AttributeUsage ] ; default userApplications
  [ "X-NDS_PUBLIC_READ" qdstrings ]
      ; default not public read ('0')
  [ "X-NDS_SERVER_READ" qdstrings ]
      ; default not server read ('0')
  [ "X-NDS_NEVER_SYNC" qdstrings ]
      ; default not never sync ('0')
  [ "X-NDS_NOT_SCHED_SYNC_IMMEDIATE" qdstrings ]
      ; default sched sync immediate ('0')
  [ "X-NDS_SCHED_SYNC_NEVER" qdstrings ]
      ; default schedule sync ('0')
  [ "X-NDS_LOWER_BOUND" qdstrings ]
      ; default no lower bound('0')
      ;(upper is specified in SYNTAX)
  [ "X-NDS_NAME_VALUE_ACCESS" qdstrings ]
      ; default not name value access ('0')
  [ "X-NDS_NAME" qdstrings ] ; legacy NDS name
whsp ")"
```

The following example LDIF file adds the title attribute type to the schema:

```

1 version: 1
2 dn: cn=schema
3 changetype: add
4 attributeTypes: ( 2.5.4.12 NAME 'title' DESC 'Standa
5 rd Attribute' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{
6 64} X-NDS_NAME 'Title' X-NDS_NOT_SCHED_SYNC_IMMEDIA
7 TE '1' X-NDS_LOWER_BOUND '1')
8

```

Single-Valued versus Multivalued

An attribute defaults to multivalued unless it is explicitly made single-valued. The following example LDIF file makes title single-valued by adding the SINGLE-VALUE keyword after the SYNTAX section:

```

1 version: 1
2 dn: cn=schema
3 changetype: add
4 attributeTypes: ( 2.5.4.12 NAME 'title' DESC 'Standa
5 rd Attribute' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{
6 64} SINGLE-VALUE X-NDS_NAME 'Title' X-NDS_NOT_SCHED
7 _SYNC_IMMEDIATE '1' X-NDS_LOWER_BOUND '1')
8

```

Adding an Optional Attribute to an Existing Object Class

Although adding new schema elements is an acceptable practice, modifying or extending existing schema elements is usually dangerous. Because every schema element is uniquely identified by an OID, when you extend a standard schema element, you effectively create a second definition for the element even though it still uses the original OID. This can cause incompatibility problems.

There are times when it is appropriate to change schema elements. For example, you might need to extend or modify new schema elements as you refine them during development. Instead of adding new attributes directly to a class, you should generally use auxiliary classes only to

- ◆ Add new attributes to an existing object class.
- ◆ Subclass an existing object class.

Adding or Removing Auxiliary Classes

The following sample LDIF file creates two new attributes, creates an auxiliary class with these new attributes, then adds an inetOrgPerson entry with the auxiliary class as an object class of the entry and with values for the auxiliary class attributes.

```

version: 1
# Add an attribute to track a bear's hair. The attribute is
# multi-valued, uses a case ignore string syntax,
# and has public read rights
# Values may include: long hair, short, curly, straight,
# none, black, and brown
# X-NDS_PUBLIC_READ '1' The 1 allows public read,
# 0 denies public read
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: ( 2.16.840.1.113719.1.186.4.10 NAME
'bearHair' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
X-NDS_PUBLIC_READ '1' )

```

```
# add an attribute to store a bear's picture
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: ( 2.16.840.1.113719.1.186.4.11 NAME
'bearPicture' SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
SINGLE-VALUE )

# create an Auxiliary class for the bearfeatures
dn: cn=schema
changetype: modify
add: objectclasses
objectclasses: (2.16.840.1.113719.1.186.6.101 NAME
'bearFeatures' MAY (bearHair $ bearPicture) AUXILIARY)

# now create a user named bobby
dn: cn=bobby,o=bearcave
changetype: add
cn: bobby
sn: bear
givenName: bobby
bearHair: Short
bearHair: Brown
bearHair: Curly
bearPicture:< file:///c:/tmp/alien.jpg
objectClass: top
objectClass: person
objectClass: inetOrgPerson
objectClass: bearFeatures

# now create a person named john that will later be changed
# into a bear when bearFeatures is added to its objectClass
# list
dn: cn=john,o=bearcave
changetype: add
cn: John
sn: bear
givenName: john
objectClass: top
objectClass: person
objectClass: inetOrgPerson

# now morph john into a bear by adding bearFeatures
dn: cn=john,o=bearcave
changetype: modify
add: objectClass
objectClass: bearFeatures
-
add: bearHair
bearHair: long
bearHair: black
#bearPicture:< file:///c:/tmp/john.jpg>
-

# to morph john back to a person, simply delete the
# objectClass bearFeatures
dn: cn=john,o=bearcave
changetype: modify
delete: objectClass
objectClass: bearFeatures
```

When removing auxiliary classes, you don't have to delete all of the values associated with the auxiliary class when you remove the auxiliary class from the objectClass list. eDirectory does this automatically.

If the auxiliary class had MUST attributes, they must all be specified in the same modify operation that adds the auxiliary class to the objectClass list, or the modification will fail.

Known Problems with XML Parsing

XML processing of any LDIF Record (LDIF format or records generated from LDAP server) will not succeed if the individual records will not satisfy all the XML rules specified in the XML file

Troubleshooting eDirectory on Linux, Solaris, AIX, and HP-UX

This section includes information for troubleshooting eDirectory on Linux, Solaris, AIX, and HP-UX platforms.

- ◆ [“Repeated eDirectory Logins” on page 500](#)
- ◆ [“Novell Public Key Infrastructure Services” on page 500](#)
- ◆ [“NMAS on UNIX” on page 501](#)
- ◆ [“LDAP Services” on page 501](#)
- ◆ [“Novell Import Convert Export Utility” on page 502](#)
- ◆ [“ndsmerge Utility” on page 503](#)
- ◆ [“ndstrace Utility” on page 503](#)
- ◆ [“ndsbackup Utility” on page 503](#)
- ◆ [“Installation and Configuration” on page 503](#)
- ◆ [“Using Ndsrepair” on page 504](#)
- ◆ [“Using ndstrace” on page 511](#)

Repeated eDirectory Logins

Repeated eDirectory logins can use up the available memory. Disable the Login Update attribute using ndsmonitor to overcome this problem.

Novell Public Key Infrastructure Services

PKI Operations Not Working

If PKI operations in ConsoleOne or iManager are not working, it could be because Novell PKI Services are not running on the Linux, Solaris, AIX, or HP-UX host. Start the PKI Services by entering `npki -1`.

If you cannot create certificates, you need to ensure that the NICI module has been properly installed. See [“Initializing the NICI Module on the Server” on page 81](#). To verify if NICI is initialized, see [“Verifying Whether NICI Is Installed and Initialized on the Server” on page 81](#).


LDAP Search from Netscape Address Book Fails

If you are using an export version of the Netscape browser and a KMO key size larger than 512 bits associated with the LDAP Server object, the LDAP search from the Netscape Address Book might fail.

Use a domestic version of the Netscape browser in such cases.

Removing the configuration of an eDirectory server that is acting as a treekey server in a multiserver tree after having moved the existing eDirectory objects to a different server fails with the error code for Crucial Replica.

To complete the operation, change the Key Server DN attribute in the W0 object under Security Container > KAP to another server in the tree that has downloaded the treekey from this server.

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click eDirectory Administration > Modify Object.
- 3 Specify the name and context of the W0 object (usually W0.KAP.Security), then click OK.
- 4 In the Valued Attributes column, select NDSPKI:SD Key Server DN, then click Edit.
- 5 Specify the name and context of a different server in the Security Domain Key Server's DN field, then click OK.
- 6 Click Apply, then click OK.

While Uninstalling the eDirectory Server holding the CA, the KMOs created on that server will be moved to another server in the tree and become invalid

You should re-create the CA and KMOs for the tree. See [“Creating an Organizational Certificate Authority Object” on page 82](#) and [“Creating a Server Certificate Object” on page 83](#) for more information.

We recommend that you do not uninstall the eDirectory server where the CA for the tree has been created.

NMAS on UNIX

Unable to Log In Using Any Method

After installing and configuring NMAS, restart NDS Server.

After reinstalling a method after you have uninstalled a previous instance of that method, restart NDS Server.

The User Added Using the ICE Utility Is Unable to Log In Using Simple Password

While adding users with simple passwords through the Novell Import Conversion Export utility, use the -l option.

LDAP Services

This section identifies some common problems you might experience with LDAP Services for eDirectory and how to solve them.

Ensure that the LDAP server is up before issuing a request from an LDAP client. To do so, look for the following message in the `/var/nds/ndsd.log`:

```
LDAP v3 for Novell eDirectory 8.7.3 started
```

For more information, see [Chapter 12, “Configuring LDAP Services for Novell eDirectory,” on page 283](#).

LDAP Clients Cannot Bind to LDAP Services for eDirectory

If an LDAP client cannot bind to LDAP Services for eDirectory, check the following:

- Is the user entering the correct username and password?
- Is the user entering an LDAP form of the name?
- Has the password expired?
- Has the server been reconfigured?

LDAP Server Isn't Using a New Configuration

Processing LDAP server configuration updates can be affected by currently bound LDAP clients.

Configuration changes are updated dynamically. The LDAP server checks for configuration changes periodically (every 30 minutes). When a change is detected, new clients cannot bind to the LDAP server during the reconfiguration process.

The LDAP server stops processing new LDAP requests for any clients currently bound and waits for any active LDAP requests to complete before updating the configuration.

LDAP operations fail when a tree is renamed using the `ndsmerge` utility. To work properly, the LDAP server must be refreshed or restarted after a tree is renamed.

Failure of Secure LDAP Connection

Ensure the following:

- The Certificate Authority and the Key Material object (KMO) have been created for the LDAP server.
- The KMO has been associated with the LDAP server.
- The specified CA expiration date has not elapsed. Verify whether the system date exceeds the expiration date.
- The LDAP server is listening on the secure LDAP port. The default is 636.
- SSL is enabled for LDAP Server object in iManager.

For more information, see [“Ensuring Secure eDirectory Operations on Linux, Solaris, AIX, and HP-UX Systems” on page 81](#).

Novell Import Convert Export Utility

If an LDAP server is refreshed or unloaded, while a Novell Import Conversion Export operation is running, the `LBURP operation is timed out` message is displayed on the Novell Import Conversion Export screen. The server recovers later, when the `LBURP` operation times out.

ndsmerge Utility

The PKI servers are not active after a merge operation. They must be restarted using the `npki -l` command.

Merge operations might not be successful on different versions of the product. If your server is running an older version of NDS or eDirectory, update to the latest version of eDirectory, then continue the merge operations.

The merging of two trees will not succeed if containers with similar names subordinate to a tree are present in both the source and target trees. Rename one of the containers, then continue with the merge operation.

During the graft operation, error message `-611 Illegal Containment` might appear. Modify the schema by running `ndsrepair(1)`. Then run `ndsrepair -S` and select **Optional Schema Enhancements**.

ndstrace Utility

When you turn on the `ndstrace(1)` screen, an error message might display indicating that a primary object is invalid for the reference link. You can ignore this message if eDirectory is functioning correctly.

ndsbackup Utility

While backing up eDirectory, `NDS Error: Connect to NDS server failed` might display. This might be caused by eDirectory listening on a port other than the default port 524. At the command line, enter the port number that eDirectory was configured on. For example, if eDirectory is configured on port number 1524, enter the following:

```
ndsbackup sR 164.99.148.82:1524
```

Installation and Configuration

Installation Not Successful

- ◆ Check for the following error message in the `/var/adm/messages` directory:

```
Unable to bind to SLP Multicast Address. Multicast route not added?
```

This message is displayed if the Linux or Solaris machine is not configured for a multicast route address.

Add the multicast route address and restart the `slpuasa` daemon.
- ◆ If the `-632: Error description System failure` error message appears during installation, exit from the installation process.

Set the `n4u.base.slp.max-wait` parameter to a larger value, such as 50, in the `/etc/nds.conf` file, then restart the installation process.
- ◆ If you are installing eDirectory into a NetWare 5.1 tree, upgrade the eDirectory Master to NetWare 5.1 Support Pack 5 or later.

For more information, see [“Installing or Upgrading Novell eDirectory on NetWare”](#) in the *Novell eDirectory 8.7.3 Installation Guide*.

- ♦ If you tried to upgrade an eDirectory for Solaris 2.0 installation and it was not successful, the installation might not complete the second time.

Delete the `/var/nds/.n4s_upgrade` file and try the installation again.

- ♦ During installation, if the `Tree Name Not Found` error message is displayed, do the following:
 - 1** Check whether multicast routing is enabled on the Solaris host that you are installing the product on.
 - 2** Specify the IP address of the master server of the Tree partition.

Installation Takes a Long Time

When you are installing eDirectory into an existing tree and the installation takes a long time to complete, look at the `DSTrace` screen on the server. If the `-625 Transport failure` message is displayed, you need to reset the address cache.

To reset the address cache, enter the following command at the system console:

```
set dstrace = *A
```

Unable to Install into an Existing Tree over the WAN

You need a NetWare 5 or later server to install eDirectory on a Linux or Solaris system over the WAN.

- 1** Enter the following command at the server console to run the Directory Agent (DA) on the NetWare server:

```
slpda
```

- 2** On the server containing the master replica, edit the `DA_ADDR` parameter in `slpuasa.conf`:

```
DA_ADDR = IP_address_of_the_NetWare_server_where_the_DA_
is_running
```

- 3** Restart the `slpuasa` daemon.
- 4** Install eDirectory over the WAN on the Linux or Solaris system.

- 4a** Run `nds-install` to add the product packages.

Do not configure the product. See “[Linux, Solaris, and AIX Packages for Novell eDirectory](#)” in the *Novell eDirectory 8.7.3 Installation Guide* for more information.

- 4b** Edit the `/etc/nds.conf` and add the following parameters:

```
n4u.uam.ncp-retries = 5
n4u.base.slp.max-wait = 20
```

- 4c** Edit the `/etc/slpuasa.conf` to add the following parameter:

```
DA_ADDR = IP_address_of_the_NetWare_server_where_the
_DA_is_running
```

- 4d** Run `ndsconfig` to configure eDirectory.

Using Ndsrepair

This section consists of the following:

- ♦ “[Syntax](#)” on page 505

- ◆ [“Troubleshooting ndsrepair” on page 511](#)

Use the ndsrepair utility at the server console to do the following:

- ◆ Correct eDirectory problems such as bad records, schema mismatches, bad server addresses, and external references.
- ◆ Make advanced changes to the eDirectory schema.
- ◆ Perform the following operations on the eDirectory database:
 - ◆ Check the structure of the database automatically without closing the database and without database intervention.
 - ◆ Check the database index.
 - ◆ Repair the database without closing the database and locking out users.
 - ◆ Reclaim free space by discarding empty records.

Syntax

To run ndsrepair, use the following syntax:

```
ndsrepair {-U| -P| -S| -C| -E| -N| -T| -J entry_id
[-A yes|no] [-O yes|no] [-F filename] [-Ad]
```

or

```
ndsrepair -R [-l yes|no [-u yes|no] [-m yes|no] [-i yes|no] [-f yes|no] [-d
yes|no] [-t yes|no] [-o yes|no] [-r yes|no] [-v yes|no] [-c yes|no] [-A
yes|no] [-O yes|no] [-F filename]
```

IMPORTANT: The -Ad option should not be used without prior direction from Novell Support personnel.

Ndsrepair Options

Option	Description
-U	Unattended Full Repair option. Instructs ndsrepair to run and exit without further user intervention. This is the suggested means of repair unless you are told by Novell Support to perform certain operations manually. You can view the log file after the repair has completed to determine what changes ndsrepair has made.
-P	Replica and Partition Operations option. Lists the partitions that have replicas stored in the current server's eDirectory database files. The Replica options menu provides options to repair replicas, cancel a partition operation, schedule synchronization, and designate the local replica as the master replica. For more information, see “Replica and Partition Operations Option” on page 508 .
-S	Global Schema Operations option. This option contains several schema operations that might be necessary to bring the server's schema into compliance with the master of the Tree object. However, these operations should be used only when necessary. The local and unattended repair operations already verify the schema.
-C	Check External Reference Object option. Checks each external reference object to determine if a replica containing the object can be located. If all servers that contain a replica of the partition with the object are inaccessible, the object will not be found. If the object cannot be found, a warning is posted.

Option	Description
-E	Report Replica Synchronization option. Reports replica synchronization status for every partition that has a replica on the current server. This operation reads the synchronization status attribute from the replica's Tree object on each server that holds replicas of the partitions. It displays the time of the last successful synchronization to all servers and any errors that have occurred since the last synchronization. A warning message is displayed if synchronization has not completed within 12 hours.
-N	Servers Known to This Database option. Lists all servers known to the local eDirectory database. If your current server contains a replica of the Tree partition, this server displays a list of all servers in the eDirectory tree. Select one server to cause the server options to be executed.
-J	Repairs a single object on the local server. You will need to provide the Entry ID (in hexadecimal format) of the object you want to repair. You can use this option instead of using the Unattended Repair (-U) option to repair one particular object that is corrupted. The Unattended Repair option can take many hours depending on the size of database. This option will help you save time.
-T	Time Synchronization option. Contacts every server known to the local eDirectory database and requests information about each server's time synchronization status. If this server contains a replica of the Tree partition, then every server in the eDirectory tree will be polled. The version of eDirectory that is running on each server is also reported.
-A	Append to the existing log file. The information is added to the existing log file. By default, this option is enabled.
-O	Logs the output in a file. By default, this option is enabled.
-F <i>filename</i>	Logs the output in the specified file.
-R	Repair the Local Database option. Repairs the local eDirectory database. Use the repair operation to resolve inconsistencies in the local database so that it can be opened and accessed by eDirectory. This option has suboptions that facilitate repair operations on the database. It has function modifiers which are explained in "Function Modifiers Used with the -R Option" on page 506 .

Function Modifiers Used with the -R Option

Modifier	Description
-l	Locks the eDirectory database during the repair operation.
-u	Uses a temporary eDirectory database during the repair operation.
-m	Maintains the original unrepaired database.
-i	Checks the eDirectory database structure and the index.
-f	Reclaims the free space in the database.
-d	Rebuilds the entire database.

Modifier	Description
-t	Performs a tree structure check. Set it to Yes to check all the tree structure links for correct connectivity in the database. Set it to No to skip the check. Default=Yes
-o	Rebuilds the operational schema.
-r	Repairs all the local replicas.
-v	Validates the stream files.
-c	Checks local references.

Global Schema Operations

You can use the `ndsrepair -S` ([-Ad] advanced switch) option to display a list showing all the schema operations that you can perform. The following table shows the available options.

Option	Description
Request Schema From Master Server	Requests the master replica of the root of the tree to synchronize its schema to this server. Any changes to the schema will be propagated to this server from the master replica of the Tree object for the next 24 hours. If all servers request the schema from the master replica, network traffic can increase.
Reset Local Schema	Invokes a schema reset that clears the time stamps on the local schema and requests an inbound schema synchronization. This option is unavailable if executed from the master replica of the Tree partition. This is to ensure that all servers in the tree are not reset at the same time.
Post NetWare 5 Schema Update	Extends and modifies the schema for compatibility with post-NetWare 5 DS changes. This option requires that the server where <code>ndsrepair</code> is run contains a replica of the Tree partition, and that the state of the replica is On.
Optional Schema Enhancements	Extends and modifies the schema for containment and other schema enhancements. This option requires this server to contain a replica of the Tree partition, and the replica state must be On. In addition, all NetWare 4 servers in the tree must have the following versions of eDirectory: <ul style="list-style-type: none"> ◆ NetWare 4.10 servers must have NDS 5.17 or later ◆ NetWare 4.11/4.2 servers must have NDS 6.03 or later <p>Previous versions of NDS will not be able to synchronize these changes.</p>

Option	Description
Import Remote Schema (Advanced Switch Option)	Select an eDirectory tree that contains the schema you want to add to the schema of the current tree. After you select a tree, the server that holds the master replica of the Tree partition is contacted. The schema from that server will be used to extend the schema on the current tree.
Declare a New Epoch (Advanced Switch Option)	When you declare a new schema epoch, the master replica of the Tree partition is contacted and illegal time stamps are repaired on the schema declared on that server. All other servers receive a new copy of the schema including the repaired time stamps. If the receiving server contains a schema that was not in the new epoch, objects and attributes that use the old schema are changed to the Unknown object class or attribute.

Replica and Partition Operations Option

Enter the following command to display information about each replica stored on the server:

```
ndsrepair -P
```

Select the required replica. The following options are displayed:

- ◆ Repair All Replicas
 - Repairs all replicas displayed in the replica table.
- ◆ Repair Selected Replica
 - Repairs only the selected replica listed in the replica table.

IMPORTANT: Repairing a replica consists of checking each object in the replica for consistency with the schema and data according to the syntax of the attribute. Other internal data structures associated with the replica are also checked. If you have not repaired the local eDirectory database in the last 30 minutes, you should do so before repairing any replicas.
- ◆ Schedule Immediate Synchronization
 - Schedules the immediate synchronization of all the replicas. This is useful if you are viewing the ndstrace screen and want to view eDirectory information for the synchronization process without having to wait for it to run as normally scheduled.
- ◆ Cancel Partition Operation
 - Cancels a partition operation on the selected partition. This option might be necessary if an operation appears to be incomplete or is not completing due to problems in the eDirectory tree, such as a missing server or bad communication links. Some operations might not be cancelled if they have progressed too far.
- ◆ Designate This Server as the New Master Replica
 - Designates the local replica of the selected partition as the new master replica. Use this option to designate a new master replica if the original master replica is lost.
- ◆ Report Synchronization Status of All Servers
 - Reports replica synchronization status of all partitions on the current server. It displays the time of the last successful synchronization to all servers and any errors that have occurred since the last synchronization.

- ◆ Synchronize the Replica on All Servers

Determines the complete synchronization status on every server that has a replica of the selected partition. This helps you determine the health of a partition. If all of the servers with a replica of the partition are synchronizing properly, then the partition is considered healthy. Each server performs an immediate synchronization to every other server in the replica ring. Servers do not synchronize to themselves. Therefore, the status for the current server's own replicas is displayed as Host.

- ◆ Repair Ring, All Replicas

Repairs the replica ring of all the replicas displayed in the replica table.

- ◆ Repair Ring, Selected Replica

Repairs the replica ring of selected replica listed in the replica table.

IMPORTANT: Repairing a replica ring consists of checking the replica ring information on each server that contains a replica of a given partition and validating remote ID information. If you have not repaired the local eDirectory database in the last 30 minutes, you should do so before repairing all or selected rings. You can repair the local database using the -R option. For more information, see [“-R” on page 506](#).

- ◆ View Replica Ring

Displays a list of all servers that contain a replica of the selected partition. This set of servers is called the replica ring. The replica ring list shows information about the type of replica and current status for each server in the ring. Select a server after viewing the replica ring to view server options.

Server Options

- ◆ Report Synchronization Status on the Selected Server

Reports replica synchronization status for a selected partition that has a replica on a selected server. This operation reads the synchronization status attribute from the replica root object on each server that holds replicas of the partitions. It displays the time of the last successful synchronization to all servers and any errors that have occurred since the last synchronization. This option displays a warning message if synchronization has not completed within 12 hours.

- ◆ Synchronize the Replica on the Selected Server

Determines the complete synchronization status on the selected server that has a replica of the selected partition. This helps you determine the health of a partition. If the server with a replica on the partition is synchronizing properly, the partition is considered healthy. The server is immediately synchronized to every other server in the replica ring. The server does not synchronize with itself. Therefore, the status for the current server's own replica is displayed as Host.

- ◆ Send All Objects to Every Replica in the Ring

Sends all objects from the selected server in the replica ring to all other servers that contain a replica of the partition. This operation can generate a lot of network traffic. Use this option to ensure that the selected partition's replica on the selected server in the replica ring is synchronized with all other servers in the replica ring. This operation cannot be performed on a server that contains only a subordinate reference replica of the partition.

- ◆ Receive All Objects from the Master to This Replica

Receives all objects from the master replica to the replica on the selected servers. This operation can generate a lot of network traffic. Use this option to ensure that the selected

partition's replica on the selected server in the replica ring is synchronized with the master replica. This operation cannot be performed on a server that contains only a master replica.

- ◆ View Entire Server's Name

Used to view the complete server name when the width of the server name is too long to view from within the server table.

- ◆ Remove This Server from Replica Ring

(Advanced switch option.) Removes a selected server from the selected replica stored on the current server. If a server appears in the replica ring but it is no longer part of the eDirectory tree or no longer contains a replica of the partition, delete the Server object using iManager. When the Server object has been deleted, the object should eventually be excluded from the replica ring.

WARNING: Misuse of this operation can cause irrevocable damage to the eDirectory database. You should not use this option unless directed by Novell Support personnel.

- ◆ View Entire Partition Name

Determines the complete distinguished partition name when the width of the partition is too great to view from within the replica table.

- ◆ Repair Time Stamps and Declare a New Epoch

(Advanced switch option.) Provides a new point of reference to the master replica so that all updates to replicas of the selected partition are current. This operation is always performed on the master replica of a partition. The master replica does not need to be in the local replica on this server. Time stamps are placed on objects when they are created or modified and they must be unique. All time stamps in a master replica are examined. If any time stamps are post-dated to the current network time, they are replaced with a new time stamp.

- ◆ Destroy the Selected Replica on This Server

(Advanced switch option.) Removes the selected replica on this server. Using this option is not recommended. Use this option only when all other utilities are unable to delete the replica.

- ◆ Delete Unknown Leaf Objects

(Advanced switch option.) Deletes all objects in the local eDirectory database that have the unknown object class and maintain no subordinate objects. This option marks Unknown objects for deletion. The deletion will later be synchronized to other replicas in the eDirectory tree.

WARNING: Use this option only when the objects cannot be modified or deleted using ConsoleOne or iManager.

Options on Servers Known to This Database

The following repair options are available for servers:

- ◆ Repair All Network Address

Checks the network address for every server in the local eDirectory database. This option searches the SLP directory agent, depending on the transport protocol available, for each server's name. Each address is then compared to the Server object's network address property and the address record of each replica property of every partition Tree object. If the addresses are different, they are updated to be the same.

- ◆ Repair Selected Server's Network Address

Checks the network address for a specific server in the local eDirectory database files. This option searches the SLP directory agent, depending on the transport protocols currently bound for the server's name.

- ◆ View Entire Server's Name

Displays the complete name of the server when the width of the server name is too great to view from within the server's table. This option is the same as the `-P` option. For more information, see [“-P” on page 505](#).

Examples

To perform an unattended repair and log events in the `/root/ndsrepair.log` file, or to append events to the log file if it already exists, enter the following command:

```
ndsrepair -U -A no -F /root/ndsrepair.log
```

To display a list of all global schema operations along with the advanced options, enter the following command:

```
ndsrepair -S -Ad
```

To repair the local database by forcing a database lock, enter the following command:

```
ndsrepair -R -l yes
```

NOTE: The input for the `ndsrepair` command can be redirected from an option file. The option file is a text file that can contain replica and partition operation-related options and suboptions that do not require authentication to the server. Each option or suboption is separated by a new line. Make sure that the contents of the file are in the proper sequence. If the contents are not in the proper sequence, the results will be unpredictable.

Troubleshooting ndsrepair

Error -786 While Running Ndsrepair

While doing `ndsrepair` you need to have three times the size of DIB free in that specific partition of your machine.

Using ndstrace

The `ndstrace` utility has three main parts:

- ◆ [“Basic Functions” on page 511](#)
- ◆ [“Debugging Messages” on page 512](#)
- ◆ [“Background Processes” on page 515](#)

Basic Functions

The basic functions of `ndstrace` are used to

- ◆ View the status of the `ndstrace` screen in Linux, Solaris, AIX, or HP-UX.
- ◆ Initiate limited synchronization processes.

To start the `ndstrace` screen, enter the following command at the server prompt:

```
/usr/bin/ndstrace
```

To initiate the basic ndstrace functions, enter commands at the server prompt using the following syntax:

```
set ndstrace= command_option
```

The following table lists the command options that you can enter.

Option	Description
ON	Starts the eDirectory trace screen with basic trace messages.
OFF	Disables the trace screen.
ALL	Starts the eDirectory trace screen and displays all the trace messages.
AGENT	Starts the eDirectory trace screen with the trace messages that are equivalent to the ON, BACKLINK, DSAGENT, JANITOR, RESNAME, and VCLIENT flags.
DEBUG	Turns on a predefined set of trace messages typically used for debugging. The flags set are ON, BACKLINK, ERRORS, EMU, FRAGGER, INIT, INSPECTOR, JANITOR, LIMBER, MISC, PART, RECMAN, REPAIR, SCHEMA, SKULKER, STREAMS, and VCLIENT.
NODEBUG	Leaves the trace screen enabled, but turns off all debugging messages previously set. This option also leaves the messages set to the ON command option.

Debugging Messages

When the ndstrace screen is enabled, the information displayed is based on a default set of filters. If you want to view more or less than the default, you can manipulate the filters using the debugging message flags. The debugging messages help you determine the status of eDirectory and verify that everything is working well.

Each eDirectory process has a set of debugging messages. To view the debugging messages on a particular process, use a plus sign (+) and the process name or option. To disable the display of a process, use a minus sign (-) and the process name or option. The following are some examples:

Message	Description
set ndstrace = +SYNC	Enables the synchronization messages.
set ndstrace = -SYNC	Disables the synchronization messages.
set ndstrace = +SCHEMA	Enables the schema messages.

You can also combine the debugging message flags by using the Boolean operators & (which means AND) and | (which means OR). The syntax for controlling the debugging messages at the server console is as follows:

```
set ndstrace = +trace_flag [trace_flag]
```

or

```
set ndstrace = +trace_flag> [&trace_flag]
```


The following table describes the trace flags for the debugging messages. You can enter abbreviations for each of the trace flags.

Trace Flag	Description
ABUF	Messages and information related to inbound and outbound packet buffers that contain data being received in conjunction with, or in response to, an eDirectory request.
ALOC	Messages to show the details of memory allocation.
AREQ	Messages related to inbound requests from other servers or clients.
AUTH	Messages and error reports relating to authentication.
BASE	Debug error messages at the minimum debugging level.
BLNK	Backlink and inbound obituary messages and error reports.
CBUF	Messages related to outbound DS Client requests.
CHNG	Change cache messages.
COLL	Status and error reports concerning an object's update information when the update has been previously received.
CONN	Messages that show information about the servers your server is trying to connect to, and about errors and timeouts that might be causing your server not to connect.
DNS	Messages about the eDirectory-integrated DNS server processes.
DRLK	Distributed reference link messages.
DVRS	Messages to show DirXML [®] driver-specific areas that eDirectory might be working on.
DXML	Messages to show details of DirXML events.
FRAG	Messages from the NCP [™] fragger which breaks eDirectory messages into NCP-sized messages.
IN	Messages related to inbound requests and processes.
INIT	Messages related to the initialization of eDirectory.
INSP	Messages related to the integrity of objects in the source server's local database. Using this flag increases the demands on the source server's disk storage system, memory, and processor. Do not leave this flag enabled unless objects are being corrupted.
JNTR	Messages related to the following background processes: janitor, replica synchronization, and flat cleaner.
LDAP	Messages related to the LDAP server.
LMBR	Messages related to the limber process.
LOCK	Messages related to the use and manipulation of the source server's local database locks.

Trace Flag	Description
LOST	Messages related to lost entries.
MISC	Messages from different sources in eDirectory.
MOVE	Messages from the move partition or move subtree operations.
NCPE	Messages to show the server receiving NCP-level requests.
NMON	Messages related to iMonitor.
OBIT	Messages from the obituary process.
PART	Messages related to partition operations from background processes and from request processing.
PURG	Messages about the purge process.
RECM	Messages related to the manipulation of the source server's database.
RSLV	Reports related to the processing of resolve name requests.
SADV	Messages related to the registration of tree names and partitions with Service Location Protocol (SLP).
SCMA	Messages related to the schema synchronization process.
SCMD	Messages showing the details of schema-related operations. They give details of both inbound and outbound synchronization.
SKLK	Messages related to the replica synchronization process.
SPKT	Messages related to eDirectory NCP server-level information.
STRM	Messages related to the processing of attributes with a Stream syntax.
SYDL	Messages showing more details during the replication process.
SYNC	Messages about inbound synchronization traffic (what is being received by the server).
TAGS	Displays the tag string that identifies the trace option that generated the event on each line displayed by the trace process.
THRD	Messages to show when any background processes (threads) begin and end.
TIME	Messages about the transitive vectors that are used during the synchronization process.
TVEC	Messages related to the following attributes: Synchronize Up To, Replica Up To, and Transitive Vector.
VCLN	Messages related to the establishment or deletion of connections with other servers.

As you use the debugging messages in `ndstrace`, you will find that some of the trace flags are more useful than others. One of the favorite `ndstrace` settings of Novell Support is actually a shortcut:

```
set ndstrace = A81164B91
```

This setting enables a group of debugging messages.

Background Processes

In addition to the debugging messages, which help you check the status of eDirectory, there is a set of commands that force the eDirectory background processes to run. To force the background process to run, place an asterisk (*) before the command. For example:

```
set ndstrace = *H
```

You can also change the status, timing, and control for a few of the background processes. To change these values, place an exclamation point (!) before the command and enter a new parameter or value. For example:

```
set ndstrace = !H 15 (parameter_value_in_minutes)
```

The following is the syntax for each statement controlling the background processes of eDirectory:

```
set ndstrace = *trace_flag [parameter]
```

or

```
set ndstrace = !trace_flag [parameter]
```

The following table lists the trace flags for the background processes, any required parameters, and the process the trace flags will display.

Trace Flag	Parameters	Description
*A	None	Resets the address cache on the source server.
*AD	None	Disables the address cache on the source server.
*AE	None	Enables the address cache on the source server.
*B	None	Schedules the backlink process to begin execution on the source server in one second.
!B	Time	Sets the interval (in minutes) for the backlink process. Default=1500 minutes (25 hours) Range=2 to 10080 minutes (168 hours)
*CT	None	Displays the source server's outbound connection table and the current statistical information for the table. These statistics do not give any information about the inbound connections from other servers or clients to the source server.
*CTD	None	Displays, in comma-delimited format, the source server's outbound connection table and the current statistical information for the table. These statistics do not give any information about the inbound connections from other servers or clients to the source server.
*D	Replica rootEntry ID	Removes the specified local entry ID from the source server's Send All Object list. The entry ID must specify a partition root object that is specific to the server's local database. This command is usually used only when a Send All Updates process is endlessly trying to show updates and failing because a server is inaccessible.

Trace Flag	Parameters	Description
!D	Time	Sets the inbound and outbound synchronization interval to the specified number of minutes. Default=24 minutes. Range=2 to 10080 minutes (168 hours)
!DI	Time	Sets the inbound synchronization interval to the specified number of minutes. Default=24 minutes Range=2 to 10080 minutes (168 hours)
!DO	Time	Sets the outbound synchronization interval to the specified number of minutes. Default=24 minutes Range=2 to 10080 minutes (168 hours)
*E	None	Reinitializes the source server's entry cache.
!E	None	Schedules the inbound and outbound synchronization processes to begin execution.
!EI	None	Schedules the inbound synchronization process to begin execution.
!EO	None	Schedules the outbound synchronization process to begin execution.
*F	None	Schedules the flat cleaner process, which is part of the janitor process, to begin execution on the source server in five seconds.
!F	Time	Sets the interval (in minutes) for the flat cleaner process. Default=240 minutes (4 hours) Range=2 to 10080 minutes (168 hours)
*G	Replica rootEntry ID	Rebuilds the change cache of the specified root partition ID.
*H	None	Schedules the replica synchronization process to begin execution immediately on the source server.
!H	Time	Sets the interval (in minutes) for the heartbeat synchronization process. Default=30 minutes Range=2 to 1440 minutes (24 hours)
*HR	None	Clears the in-memory last-sent vector.
*I	Replica rootEntry ID	Adds the specified local entry ID to the source server's Send All Object list. The entry ID must specify a partition root object that is specific to the server's local database. The replica synchronization process checks the Send All Object list. If the entry ID of a partition's root object is in the list, eDirectory synchronizes all objects and attributes in the partition, regardless of the value of the Synchronized Up To attribute.

Trace Flag	Parameters	Description
!I	Time	Sets the interval (in minutes) for the heartbeat synchronization process. Default=30 minutes Range=2 to 1440 minutes (24 hours)
*J	None	Schedules the purge process, which is part of the replica synchronization process, to begin running on the source server.
!J	Time	Sets the interval (in minutes) for the janitor process. Default=2 minutes Range=1 to 10080 minutes (168 hours)
*L	None	Schedules the limber process to begin running on the source server in five seconds.
*M	Bytes	Changes the maximum file size used by the source server's ndstrace.log file. The command can be used regardless of the state of the debug file. The bytes specified must be a hexadecimal value between 10000 bytes and 100 MB. If the value specified is higher or lower than the specified range, no change occurs.
!M	None	Reports the maximum memory used by eDirectory.
!N	0 1	Sets the name form. 0=hex only 1=full dot form
*P	None	Displays the tunable parameters and their default settings.
*R	None	Resets the TTF file, which is the sys:system\ndstrace.dbg file by default. This command is the same as the SET parameter NDS Trace File Length Set to Zero.
*S	None	Schedules the Skulker process, which checks whether any of the replicas on the server need to be synchronized.
!SI	Time	Sets the interval (in minutes) for the inbound schema synchronization process. Default=24 minutes Range=2 to 10080 minutes (168 hours)
!SO	Time	Sets the interval (in minutes) for the outbound schema synchronization process. Default=24 minutes Range=2 to 10080 minutes (168 hours)
!SIO	Time	Disables the inbound schema synchronization process for the specified number of minutes. Default=24 minutes Range=2 to 10080 minutes (168 hours)

Trace Flag	Parameters	Description
!SO0	Time	Disables the inbound schema synchronization process for the specified number of minutes. Default=24 minutes Range=2 to 10080 minutes (168 hours)
*SS	None	Forces immediate schema synchronization.
*SSA	None	Schedules the schema synchronization process to begin immediately and forces schema synchronization with all target servers, even if they have been synchronized in the last 24 hours.
*SSD	None	Resets the source server's Target Schema Sync list. This list identifies which servers the source server should synchronize with during the schema synchronization process. A server that does not hold any replicas sends a request to be included in the target list of a server that contains a replica with its Server object.
*SSL	None	Prints the schema synchronization list of target servers.
*ST	None	Displays the status information for the background processes on the source server.
*STX	None	Displays the status information for the backlink process (external references) on the source server.
*STS	None	Displays the status information for the schema synchronization process on the source server.
*STO	None	Displays the status information for the backlink process (obituaries) on the source server.
*STL	None	Displays the status information for the limber process on the source server.
!T	Time	Sets the interval (in minutes) for checking the server's UP state. Default=30 minutes Range=1 to 720 minutes (12 hours)
*U	Optional ID of server	If the command does not include an entry ID, this changed the status of any server that has been previously labeled <i>down</i> to <i>up</i> . If the command includes a local entry ID, it changes the status of the specified server from <i>down</i> to <i>up</i> . Entry IDs are specific to the source server's database and must refer to an object that represents a server.
!V	A list	Lists the restricted eDirectory versions. If no versions are listed, there are no restrictions. Each version is separated by a comma.
*Z	None	Displays the currently scheduled tasks.

Troubleshooting ConsoleOne

For more information on troubleshooting ConsoleOne, see [Troubleshooting \(http://www.novell.com/documentation/lg/consol13/c1_enu/data/hlgnvvum.html\)](http://www.novell.com/documentation/lg/consol13/c1_enu/data/hlgnvvum.html) in the *ConsoleOne User Guide*.

Unable to Browse the eDirectory Tree

You cannot browse the eDirectory tree if any of the following are true:

- ◆ The Tree object in the eDirectory tree has been deleted, renamed, or moved.
Log out from the network and log in to the tree again to view the changes.
- ◆ The ndsd daemon is down.

Restart the daemon by entering one of the following commands:

- ◆ On Linux: `/etc/rc.d/init.d/ndsd start`
- ◆ On Solaris: `/etc/init.d/ndsd start`
- ◆ On AIX: `/etc/ndsd start`
- ◆ On HP-UX: `/sbin/init.d/ndsd start`
- ◆ The following error message appears:

```
An attempt to resolve SVC (switched virtual circuit) failed.
```

This is because the tree is the primary tree but the server is not the primary server. Set the server as the primary server.

If you perform any one of the actions listed above, open a new ConsoleOne window to browse the tree.

Running Pure IP Environments on NetWare Servers

In a pure IP environment on NetWare servers, ConsoleOne does not see the eDirectory tree of the server that ConsoleOne is running on.

To fix this, make the following changes in the `autoexec.ncf` file and then restart the server:

- ◆ Add the statement `LOAD SCMD.NLM` after the `LOAD` and `BIND` statements for TCPIP.
- ◆ Add `IPX™` to the `serverID`.

For more information on this issue, refer to [Solution #2943528 \(http://support.novell.com\)](http://support.novell.com) in the Novell Knowledgebase.

Obituaries

There has been a great deal of confusion surrounding obituaries stored in the directory and, as a result, some people have developed poor business practices to deal with them. Unlike some directory products, Novell eDirectory ensures referential integrity between objects. For example, if Group A has a member, User B, and User B is deleted, the directory automatically removes the reference to User B from Group A. Obituaries exist as operational attributes placed on objects by eDirectory as another way of ensuring referential integrity during delete, move, rename, restore, and other operations.

There are three general classifications for obituaries:

- ◆ Primary obituaries include the types Dead (0001), Restored (0000), Moved (0002), New RDN (0005), and Tree New RDN (0008).
- ◆ Secondary obituaries are generally associated with a Primary obituary and represent the agents and partitions that need to be notified of the operation specified in the Primary obituary. They include the types Back Link (0006), Used By (000C), and Move Tree (000a).
- ◆ Tracking obituaries include the types Inhibit Move (0003), Old RDN (0004), and Tree Old RDN (0007).

Obituaries, with the exception of Tracking obituaries, must move through a set of synchronizing states:

- ◆ Initial State or Issued (0)
- ◆ Notified (1)
- ◆ OK to Purge (2)
- ◆ Purgeable (4)

The states are recorded in the Flags field in the obituary attribute. Before an obituary can move to the next state, the current state must have been synchronized to all replicas of the real object. In order to determine whether all replicas in the ring have seen a given obituary state, a vector is computed from the transitive vector. In eDirectory 8.6 and later, a nonstored Obituary Vector is used. In previous versions of eDirectory, the Purge Vector is used. If the Modification Timestamp (MTS) on the obituary is older than the corrupted vector, the server responsible for that obituary can advance it to the next state.

For a Secondary obituary of type Back Link, the agent that holds the master replica of the object with the obituary is responsible for advancing the states. For a Secondary obituary of type Used By, the replica agent that created it is responsible for advancing the obituary states as long as that replica still exists. If it does not still exist, the agent holding the master of that partition takes over advancing the obituary states for the Used By obituary. For a Move Tree obituary, the master of the root partition is responsible for advancing the states.

Primary obituaries can be advanced in their states only after all Secondary obituaries have advanced through all of their states. After the Primary obituary reaches its last state, and that state synchronizes to all servers in the ring, all that remains is the object husk, which is an object without attributes—one which can subsequently be purged from the system by the Purge Process. Tracking obituaries are removed after the Primary obituary is ready to be removed or, in the case of Inhibit_move, the Tracking obituary is removed after the Primary obituary has moved to the OBF_NOTIFIED state on the master replica.

The replica responsible for processing obituaries does so on a background process (the Obituary Process), which is scheduled on a per-partition basis after a given partition finishes an inbound synchronization cycle. If there are no other replicas of the partition, the Outbound Replication Process is still scheduled on the heartbeat interval. The Outbound Replication Process then starts the Obituary Process. The Obituary Process cannot be manually scheduled, nor does it need to be. As synchronization occurs, the transitive vectors are updated, thus advancing the Purge Vector and Obit Vector. As these vectors move forward, the obituary states are allowed to move forward. This, together with the automatic scheduling done upon inbound synchronization, completes the obituary processing cycle. Therefore, the lifeblood of obituary processing is object synchronization.

For an object that is being removed, after all obituaries whose associated Primary obituary is of type Dead have been advanced to the last state (Purgeable), and that state has been synchronized

to all replicas, a new process is responsible for removing the remaining entry husk from the database. The Purge Process runs automatically to remove these husks. You can manually schedule the Purge Process and modify its automatic schedule interval by using the [Agent Configuration](#) page in iMonitor.

Examples

This section contains the following examples:

- ◆ “[Deleting an Object](#)” on page 521
- ◆ “[Moving an Object](#)” on page 522

Deleting an Object

1 Add the Primary obituary OBT_DEAD.

The Back Link attribute contains a list of servers that have an interest in this object and need to be notified of changes to this entry. For every DN listed in the Back Link attribute and all servers listed in the entry’s partition replica attribute, eDirectory adds a Back Link obituary. The creation time of the Primary obituary, OBT_DEAD, is stored in the Secondary obituary.

The Used By attribute contains a list of partitions that have an interest in this object and need to be notified of changes to this entry. For every DN listed in the Used By attribute, eDirectory adds a Used By obituary. The creation time of the Primary obituary, OBT_DEAD, is stored in the Secondary obituary.

2 Remove all attributes but the obituaries.

The Outbound Replication Process then synchronizes this change to all other servers in the replica ring.

On the next inbound synchronization of this partition, the Obituary Process is started, which does the following:

- ◆ Computes a time vector which is a minimum transitive vector, referred to as the purge vector. Later versions of eDirectory compute a second minimum vector, called the obituary vector, which does not consider replicas which are subordinate references.
- ◆ Each Obituary in this partition is now examined.

If the obituary is a Primary obituary, there are no Secondary obituaries, and the attribute’s modification time (MTS) on the obituary is older than the Purge Vector, then all servers have seen the change and this obituary will be removed.

If the obituary is a Back Link obituary and this server is the master, then this server is responsible for processing this obituary.

IMPORTANT: Perform the required operation for this state if it has not been done. Most often, this is done by notifying an external reference.

If the obituary is a Used By obituary and this server is the server where the delete occurred (determined by comparing the replica number in the obituary’s MTS to our replica number), this server is responsible for processing this obituary.

- ◆ If this server is responsible for processing a particular Secondary obituary type (Back Link or Used By), all Secondary obituaries of that type on an entry are in the same state, the required operation for that state has been completed on all obituaries (for example, servers have been notified), and the obituary’s MTSs for that obituary type are older than the Obituary Vector, then all Secondary obituaries of that type can be advanced to the next state.

Moving an Object

Move acts much like **Delete**, but with the following changes:

- ◆ Before the Primary obituary is placed on the move source, a partial entry is created in the destination container and a Tracking obituary (OBT_INHIBIT_MOVE) is placed on that partial entry. This Tracking obituary is placed to prevent the entry from being moved or taking part in a partition operation before the full entry is transferred from the source.
- ◆ On the source entry, the Primary obituary is OBT_MOVED.
- ◆ After the Primary obituary (OBT_MOVED) is moved to the Notified state (meaning that all replicas of the source know the entry is being moved) and all external references have been notified, the Tracking obituary (OBT_INHIBIT_MOVE) is removed from the destination entry.

Impact of Stuck and Orphaned Obituaries

Objects with obituaries are considered every time an agent outbound synchronizes, and by the obituary process, which is scheduled to run at the end of an inbound synchronization cycle.

Prevention

On a regular basis, run the iMonitor Server Information report. This report walks the entire tree, communicates with every NCP server it can find, and reports any errors it finds. You can use this report to diagnose time synchronization and limber problems, or to find out if the current server is able to communicate with all other servers from this server's perspective. If selected in the configuration page, the server can also generate NDS Agent Health information for every server in the tree. See **“Configuring and Viewing Reports” on page 179** for more information on running the Server Information report.

If you are using iMonitor 2.0 or later, make sure that the Errors and Health Sub-report report options are enabled. The following items will be verified. You should browse the report and make sure that there are no errors.

- ◆ Based on the information in the ndsimonhealth configuration file stored with iMonitor (see **“Configuration Files” on page 168**), this report will check the eDirectory agent version to ensure you are running the correct directory patches tree-wide.
- ◆ All servers are within Timesync tolerances.
- ◆ This server can communicate with all other servers.
- ◆ There have not been any servers improperly or incompletely removed from the tree.
- ◆ The Health subreport will indicate if any partitions are not within tolerance for the replication sync times.

If you are using iMonitor 1.5, select the Errors report option. The following items will be verified. You should browse the report and make sure that there are no errors.

- ◆ The agent version is displayed. Make sure all servers tree-wide are running the most current eDirectory Support Pack available from the [Novell Support Web site \(http://support.novell.com\)](http://support.novell.com).
- ◆ All servers are within Timesync tolerances.
- ◆ This server can communicate with all other servers.
- ◆ There have not been any servers improperly or incompletely removed from the tree.

Using the iMonitor Obituary Listing report or the iMonitor Object Statistics report, you can find any obituaries on your system. If you find any obituaries that you don't believe are being processed, see [“Troubleshooting Tips” on page 523](#).

Troubleshooting Tips

There are two general reasons that obituaries don't process: either the obituary has been orphaned (that is, the obituary exists on some servers but not all servers) or the obituary is stuck (that is, it exists on all servers but its states are not advancing for some reason).

Do the following to troubleshoot orphaned or stuck obituaries:

- Don't panic!
- If the obituary is for an object not stored on this server (that is, the object is an External Reference):
 - ◆ Check to see if the real object has a matching obituary. If not, this obituary has been orphaned. See [“Resolving Orphaned Obituaries on Extrefs” on page 524](#) for more information.
 - ◆ If the real object has a matching obituary, troubleshoot and resolve obituary problems on the real object before attempting to address any issues with the obit on the ExtRef partition.
- Make sure that the obituaries are correctly synchronized.
 - ◆ Use the iMonitor [Agent Synchronization page](#) to check for and resolve any synchronization errors.
 - ◆ Obituaries can change states only after all agents holding a copy of the replica ring have seen the state change. There are several ways to ensure that every replica has seen the data:

While browsing the entry with obituaries, click the Entry Synchronization link. The page displayed will show all attributes that have not been synchronized to all replicas.

Find the oldest time stamp on any of the obituary attribute values. The difference between that time and the current time should be greater than the interval shown in the Max Ring Delta field on the Partition Synchronization page.

Evaluate the transitive vector.
- Run the iMonitor [Server Information Report](#) to ensure that all server communication is functioning.
- Examine the [Agent Process Status: Obituaries](#) to look for any errors.
 - ◆ Common problems in Agent Process Status: Obituaries include
 - 625, -622, -634, and -635 communication problems. See [Server Information Report](#) for more details.
 - 601, and -603, indicating servers that have been improperly removed, or that the Server object might have a base class of Unknown.
 - ◆ Errors shown on this page are not fatal. The next time the obituary process runs for that partition, it will retry the operation. Resolve any issues shown in this page, then wait for the retry.
- While looking at obituary objects, walk around the replica ring, comparing the obituary around the ring.

- ◆ If not all replicas have a copy of the obituary and all attribute values are not purgeable, this object is inconsistent around the replica ring—and this is a case of an orphaned obituary. See [“Resolving Orphaned Obituaries” on page 524](#) for more information.
- ◆ If the object exists on all replicas and is consistent, then it might not be advancing because of synchronization errors, or the obituary process might be getting errors.
- ❑ As needed, use **Trace** with the Obituary option enabled to examine the obituary process in detail.
- ❑ To prevent obituary problems in the future, upgrade to the latest Support Pack (for eDirectory 8.6 servers). There have been fixes for all known obituary issues.

Solutions

Use the proper solution referred to in [“Troubleshooting Tips” on page 523](#).

Before using any of these solutions, you must make sure that your data is safe. You might need to back up the directory database files, server configuration, and trustees. To increase the probability of success and to minimize future problems, upgrade to the latest eDirectory Support Packs.

Resolving Orphaned Obituaries

- ◆ **Preferred method:** If eDirectory 8.6 or later is on any of the servers in the replica ring, browse to the object in iMonitor, then select Send Single Entry. This will perform a nonauthoritative send to all other replicas.
- ◆ **Far less desirable method:** If all servers in the replica ring that have a copy of the orphaned obituary are older than eDirectory 8.6, load DSBrowse with the `-a` option, browse to the object, then time-stamp the entry. This will make the object as it exists on this server the authoritative copy. We do not recommend making objects authoritative as a matter of practice.

Resolving Orphaned Obituaries on Extrefs

- ◆ **Less desirable method:** Run DSRepair with the time stamp option selected.
- ◆ **Less desirable method:** Move a real replica to the server, wait for it to turn on, then wait for the obituary to be processed. If the obituary is not processed, use the information in [“Troubleshooting Tips” on page 523](#) to resolve the issue now that the object is on a real replica. After the obituary has processed, the replica can be removed if desired.

Previous Practices

In the past, several different strategies have been employed to resolve stuck obituaries. Some of these strategies involve expensive partitioning operations, or the use of undocumented features that might cause problems in the future.

The first strategy was to switch which replica held the master. This would work in some cases because the master is the agent responsible for moving the Back Link obituaries through their various states. In the case where the replica was inconsistent and the master didn't hold the deleted object, switching masters to an agent that held the deleted entry with its obituaries would give the new agent the license to push the obituaries through their states and eventually purge it out. Send Single Entry is a much cleaner and less dangerous way to resolve obituaries that are stuck because the replica is inconsistent.

The second strategy used was to run DSRepair with certain switches to delete all obituaries. (There is a third-party application which resolves stuck obituaries by launching DSRepair.) We do not recommend this strategy. Using those switches will delete all obituaries on this agent, which means

that obituaries that are not stuck might also be removed, creating further replica inconsistencies and more stuck obituaries. Because this is not a distributed operation, you must run DSRepair on all of the servers with stuck obituaries, which increases the odds that one of those servers has obituaries for another partition which will be prematurely deleted. The premature deletion of obituaries can cause additional orphaned obituaries and, in turn, cause problems which can be found years later when you change replicas types, add new replicas, or perform other partitioning operations.

The third strategy used was to make objects authoritative, either using DSBrowse with the advanced mode operation and time stamping the entry, or running DSRepair with the `-OT` switch. This forces the entry to become authoritative and synchronize out to all other replicas. This should be done with great care because you might lose data changed on other servers. We recommend that this be a rarely employed method of obituary cleanup.

Accessing HTTPSTK When DS Is Not Loaded

You can set up a preconfigured admin user that allows access to the HTTP Protocol Stack (HTTPSTK) when DS is not loaded. The preconfigured admin user, SAdmin, has rights that are equivalent to the eDirectory Admin User object. If the server is in a state where eDirectory is not functioning correctly, you can log in to the server as this user and perform all the diagnostic and debugging tasks necessary that do not require eDirectory.

- ◆ “Setting the SAdmin Password on NetWare” on page 525
- ◆ “Setting the SAdmin Password on Windows” on page 526
- ◆ “Setting the SAdmin Password on Linux, Solaris, AIX, and HP-UX” on page 526

Setting the SAdmin Password on NetWare

Use NetWare Remote Manager to enable the SAdmin User object and set or change the password for this object. HTTPSTK.NLM must be running on the eDirectory server in order for you to set or change the SAdmin password.

- 1 Open a Web browser.
- 2 In the address (URL) field, enter the following:


`http://server's TCP/IP address:port`

For example:

`http://137.65.123.11:8008`

NOTE: The default alternate port number is 8008. If you have changed this value on the Configuration page in NetWare Remote Manager, make sure you enter the new port number.

If you have Domain Name Services (DNS) installed on your network for server name-to-IP address resolution, you can also enter the server's DNS name instead of the IP address.

- 3 Specify a username, context, and password.
- 4 Click the Configure button  > Enable Emergency Account (SADMIN User) and Set Password.
- 5 Specify an SAdmin password, then verify the password you just entered.
- 6 Click Set.

Setting the SAdmin Password on Windows

Use the DHOST remote manager page (accessible through the /dhost URL or from the root page) to set the SAdmin password. dhost.exe must be running on the eDirectory server in order for you to set or change the SAdmin password.

- 1 Open a Web browser.
- 2 In the address (URL) field, enter the following:
`http://server.name:port/dhost`
for example:
`http://MyServer:80/dhost`
You can also use the server IP address to access the DHost iConsole. For example:
`http://137.65.135.150:80/dhost`
- 3 Specify a username, context, and password.
- 4 Click HTTP Server, then specify an SAdmin password.
- 5 Verify the password you just specified, then click Submit.

Setting the SAdmin Password on Linux, Solaris, AIX, and HP-UX

You can use either the DHOST remote management page or the ndsconfig utility.

DHOST remote management page

Use the DHOST remote manager page (accessible through the /dhost URL or from the root page) to set the SAdmin password. Novell eDirectory server must be running on the eDirectory server in order for you to set or change the SAdmin password.

- 1 Open a Web browser.
- 2 In the address (URL) field, enter the following:
`http://server.name:port/dhost`
for example:
`http://MyServer:80/dhost`
You can also use the server IP address to access the DHost iConsole. For example:
`http://137.65.135.150:80/dhost`
- 3 Specify a username, context, and password.
- 4 Click HTTP Server, then specify an SAdmin password.
- 5 Verify the password you just specified, then click Submit.

ndsconfig

Use the ndsconfig utility to set the SAdmin password. ndsd must be running on the eDirectory server in order for you to set or change the SAdmin password.

Enter the following at the server console

```
ndsconfig set http.server.sadmin-pwd=password
```

where *password* is the new SAdmin password.

For more information on using the ndsconfig utility, see “[ndsconfig Utility Parameters](#)” in the *Novell eDirectory 8.7.3 Installation Guide*.

Troubleshooting SNMP

This section includes information for troubleshooting SNMP on all platforms.

- ♦ “Traps Might Not Get Generated As Expected” on page 527
- ♦ “SNMP on Linux” on page 527
- ♦ “SNMP on HP-UX” on page 528

Traps Might Not Get Generated As Expected

Traps are sent only if the corresponding verb request is received by the server. They are not sent in any other cases. For example, `ndsDeleteAttribute` is sent only when the `ndsRemoveEntry` (trap number 108) request is sent. But an application can always read the ACLs and decide to check whether the user has sufficient rights to perform the delete operation. In this case, the `ndsDeleteAttribute` trap will not be generated. However, you can use `iMonitor` to view the verb statistics on a particular server.

To get the traps for all occurrences, set the time interval to zero.

You can enable traps to send only on failure conditions. You can enable traps to get them under all conditions.

SNMP on Linux

ndssnmppsa: error while loading the shared libraries: libucdmibs-0.4.2.1.so: cannot open shared object file: No such file or directory

Or

ndssnmppsa: error while loading the shared libraries: libucdagent-0.4.2.1.so: cannot open shared object file: No such file or directory

Or

ndssnmppsa: error while loading the shared libraries: libsnmp-0.4.2.1.so: cannot open shared object file: No such file or directory

An incorrect version of `ucd-snmp` might be present on the system, or the links to required libraries might be missing.

To resolve this, install `ucd-snmp-4.2.1`, `ucd-snmp-4.2.2`, or `ucd-snmp-4.2.3` and create links to the missing `ucd-snmp` version libraries.

Complete one of the following workarounds to install and create links to the missing `ucd-snmp` version libraries:

Installing `ucd-snmp-4.2.1`, `ucd-4.2.2` or `ucd-snmp-4.2.3`:

NOTE: We recommend you to use the first workaround as uninstalling `ucd-snmp` may require to uninstall all the dependent rpms.

- ♦ First workaround:
 - ♦ Download `ucd-snmp` from the following locations:
SuSe SLES-8(i386): download `ucdsnmpp-4.2.3-109.i386.rpm` from `ftp://ftp.suse.com/pub/susei389/8.0/suse/n2/`

RedHat Linux: ucd-snmp-4.2.3-1: download ucd-snmp-4.2.3-1 from <http://sourceforge.net/projects/net-snmp>

- ◆ Install any of the above mentioned ucd-snmp in a custom location using this rpm2cpio command.

For example, if you want to install ucd-snmp-4.2.3 in /home/edir/snmp, go to `cd /home/edir/snmp` and run the following commands:

```
rpm2cpio ucd-snmp-4.2.3-109.i386.rpm | cpio -ivd
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/edir/snmp/
usr/lib
```

- ◆ Create links to the missing libraries. For details, refer [Creating Links to the Missing Libraries](#).
- ◆ Second workaround:
 - ◆ If any other version of ucd-snmp is installed, other than the above mentioned versions, uninstall it.
 - ◆ Download ucd-snmp-4.2.1 or ucd-4.2.2 or ucd-snmp-4.2.3 from the following locations:
 - SuSe SLES-8(i386): download ucdsnmp-4.2.3-109.i386.rpm from <ftp://ftp.suse.com/pub/susei389/8.0/suse/n2/>
 - RedHat Linux : ucd-snmp-4.2.3-1: download ucd-snmp-4.2.3-1 from <http://sourceforge.net/projects/net-snmp>
 - ◆ Install ucd-snmp
 - ◆ Create links to the missing libraries. For details, refer [Creating Links to the Missing Libraries](#).

Creating Links to the Missing Libraries:

To find which libraries are missing, enter the following:

```
ldd /usr/bin/ndssnmpsa
```

To create links to the missing libraries, refer the following example:

If your system had ucd version 4.2 earlier, create the following links to ucd version 4.2.1:

```
ln -s /usr/lib/libucdagent-0-4.2.so /usr/lib/libucdagent-0.4.2.1.so
```

```
ln -s /usr/lib/libsnmp.so-0-4.2.so /usr/lib/libsnmp-0.4.2.1.so
```

```
ln -s /usr/lib/libucdmibs-0-4.2.so /usr/lib/libucdmibs-0.4.2.1.so
```

NOTE: If ucd snmp is installed in custom location, you should create the link by using the custom location as a prefix.

SNMP on HP-UX

Error while Contacting the SNMP Master Agent from the MIB Browser

If you get an error (for example, a timeout error) while contacting the SNMP master agent from the MIB browser, do the following:

- ◆ Ensure that the SNMP master agent is up and running.

You can check this using the ps command as follows:

```
ps -ef | grep snmpdm
```

- ◆ See the error and warning messages in /var/adm/snmpd.log file.

Problems Configuring NET-SNMP-5.0.8

See the net-snmp-5.0.8 master agent related error and warning messages in the /usr/adm/snmpd.log file.

NOTE: If the net-snmp master agent is down and is restarted, then ndssnmpsa should also be restarted.

Problems Configuring the NAA Agent

See the NAA agent related error and warning messages in the /var/adm/snmpd.log file.

Ensure that you have started the NAA agent with log messages enabled. Start the naaagt as follows:

```
/usr/sbin/naaagt -m APALL
```

NOTE: By default, naaagt terminates automatically when snmpdm terminates (unless naaagt is started with the -K option). See the naaagt man page for more details.

Unable to Get the SNMP Query Result from the MIB Browser

Ensure that net-snmp-5.0.8 is configured, up, and running.

For any problem configuring the eDirectory SNMP subagent (ndssnmpsa), see the /var/nds/ndssnmpsa.log file. To get the debug messages, start ndssnmpsa in verbose mode as follows:

```
/usr/bin/ndssnmpsa -v 3 -l 3
```

Where *v* is verbose mode and *l* indicates the log mode.

Traps are not received at the SNMP Console or the MIB Browser

Ensure that the trap destination is entered in net-snmp configuration.

For more details on how to configure see section [“Starting/Configuring the NET-SNMP Master Agent” on page 335](#).

eDirectory Install Fails for Container Administrators

The eDirectory 8.7.3 installation program supports installations by administrators who have supervisor rights to the container that the server resides in. In order to handle this, the first server that eDirectory 8.7.3 is installed into must have supervisor rights to [Root] to extend the schema. From that point on, subsequent servers do not have to have rights to [Root]. However, with eDirectory 8.7.3, depending on the platform that eDirectory 8.7.3 is installed in to first, all schema might not be extended, requiring supervisor rights to [Root] for subsequent server installations on different platforms.

If eDirectory 8.7.3 will be installed on multiple platforms, make sure that you have supervisor rights to [Root] for the first server eDirectory will be installed on for EACH platform. For example, if the first server that eDirectory 8.7.3 is going to be installed on is running NetWare, and eDirectory 8.7.3 will also be installed on Solaris, the first server for each platform must have

supervisor rights to [Root]. Subsequent servers on each platform will only have to have container administrator rights to the container where the server is being installed.

For additional information, see solution [NOVL81742 \(http://support.novell.com/cgi-bin/search/searchtid.cgi?/10073723.htm\)](http://support.novell.com/cgi-bin/search/searchtid.cgi?/10073723.htm) in the Novell eDirectory 8.7.x Readme Addendum.

Migrating the Sun ONE Schema to Novell eDirectory

To migrate the Sun ONE schema to Novell eDirectory, complete the following steps:

“Step 1: Perform the Schema Cache Update operation” on page 530

“Step 2: Rectify the error LDIF file to eliminate the errors” on page 530

“Step 3: Import the LDIF File” on page 533

Step 1: Perform the Schema Cache Update operation

You can write the errors encountered while comparing the schema to an error file using the following command:

```
ice -e LDIF error file name -C -a -SLDAP -s Sun ONE server -p Sun ONE port -
DLDAAP -s eDirectory server -p eDirectory port
```

For example:

```
ice -e err.ldf -C -a -SLDAP -s sun_srv1 -p sun_port1 -DLDAAP -s edir_srv2 -p
edir_port2
```

Any errors encountered while comparing the schema is written to the error file (err.ldf in the example). You do not need to login to perform this operation unless one of the servers require authentication in order to read the Root DSE. Microsoft Active Directory requires authentication to read the Root DSE.

Step 2: Rectify the error LDIF file to eliminate the errors

- ◆ Sun ONE defines some schema definitions publicly that eDirectory does not. This includes attributes like, “objectClasses”, “attributeTypes”, “ldapSyntaxes” and “subschemSubentry”. These definitions exist internally and are very important to the schema, and therefore, they cannot be modified. Operations that try to modify these definitions results in the following error:

```
LDAP error : 53 (DSA is unwilling to perform)
```

Any records that contain references to these definitions cause the following error:

```
LDAP error : 16 : ( No such attribute )
```

Thus, records that contain any reference to these objects or that try to modify these definitions need to be commented in the LDIF error file (err.ldf in the example).

- ◆ Some objectClasses definitions in Sun ONE do not have naming attributes. Adding these objectClasses would result in the following error in eDirectory:

```
LDAP error : 80 (NDS error: ambiguous naming (-651))
```

This error occurs because Sun ONE does not use the same method for determining naming rules as eDirectory.

To solve this, you can use any *one* of the three following options:

Option 1:

Go through each of the offending objectClasses and add a valid naming attribute to each of them.

For example:

To add the naming attribute [cn] to the objectclass “netscapeMachineData” modify the entry (that is *emphasized* in the example below) in the err.ldf file to include the X-NDS_NAMING flag as shown below:

```
dn: cn=schema
changetype: modify
add: objectClasses
objectClasses: ( 2.16.840.1.113730.3.2.32 NAME 'netscapeMachineData'
DESC 'iPlanet defined objectclass' SUP top STRUCTURAL MAY 'cn'
X-NDS_NAMING 'cn' )
-
```

Option 2:

Go through each of the offending objectClasses and make them AUXILIARY or ABSTRACT.

For example:

To modify the objectclass definition of objectclass “netscapeMachineData” from “STRUCTURAL” to “AUXILIARY”, modify the err.ldf file entry (that is *emphasized* in the example below) as shown below:

```
dn: cn=schema
changetype: modify
add: objectClasses
objectClasses: ( 2.16.840.1.113730.3.2.32 NAME 'netscapeMachineData'
DESC 'iPlanet defined objectclass' SUP top AUXILIARY )
-
```

To modify the objectclass definition of objectclass “netscapeMachineData” from “STRUCTURAL” to “ABSTRACT”, modify the err.ldf file entry (that is *emphasized* in the example below) as shown below:

```
dn: cn=schema
changetype: modify
add: objectClasses
objectClasses: ( 2.16.840.1.113730.3.2.32 NAME 'netscapeMachineData'
DESC 'iPlanet defined objectclass' SUP top ABSTRACT )
-
```

Option 3:

Add cn to the definition of Top in eDirectory, which causes a potential naming attribute for all objectClasses.

There are two ways of adding cn to Top:

◆ **Method 1:**

Create a file as shown below and name it topsch.ldf.

```
version : 1
dn:cn=schema
changetype :modify
delete : objectclasses
objectclasses : ( 2.5.6.0 NAME 'top' STRUCTURAL )
-
add:objectclasses
objectclasses : (2.5.6.0 NAME 'top' STRUCTURAL MAY cn)
```


Use the following Novell Import Conversion Export command line:

```
ice -SLDIF -f LDIF_file_name -DLDAP -s eDirectory_server -p
eDirectory_port -d eDirectory_Admin_DN -w eDirectory_password
```

For example:

```
ice -SLDIF -f topsch.ldf -DLDAP -s edir_srv2 -p edir_port2 -d
cn=admin,o=org -w pwd1
```

◆ **Method 2:**

1. In Novell iManager, click the Roles and Tasks button .
2. Click Schema > Add Attribute.
3. In the Available Classes list, select Top, then click OK.
4. Double-click CN in the Available Optional Attributes list.
5. Click OK.

- ◆ Some objectClass definitions contain *userPassword* as part of their mandatory attributes list. Adding such objectClasses to eDirectory cause the following error:

```
LDAP error : 16 (No such attribute)
```

To resolve this error, modify the objectClass definition to inherit the new objectClass from *ndsLoginProperties* and remove the *userPassword* attribute from the mandatory attribute list.

For example:

An objectClass containing *userPassword* in the mandatory attributes list:

```
version : 1
dn: cn=schemaz
changetype: modify
add: objectClasses
objectClasses: ( 0.9.2342.19200300.100.4.19 NAME 'simpleSecurityObject'
DESC '
Standard LDAP objectClass' SUP top STRUCTURAL MUST userPassword )
```

Needs to be modified as following (notice the change to the last line):

```
version : 1
dn: cn=schema
changetype: modify
add: objectClasses
```

```
objectClasses: ( 0.9.2342.19200300.100.4.19 NAME 'simpleSecurityObject'
DESC '
Standard LDAP objectClass' SUP (ndsLoginProperties $ top) STRUCTURAL )
```

Step 3: Import the LDIF File

Use the following Novell Import Conversion Export command to import the modified schema compare LDIF file (err.ldf in our example):

```
ice -e error_file -SLDIF -f modified_LDIF_file -DLdap -s eDirectory_server -
p eDirectory_port -d eDirectory_Admin_DN -w eDirectory_password
```

For example:

```
ice -e errors.ldf -SLDIF -f err.ldf -DLdap -s edir_srv2 -p edir_port2 -d
cn=admin,o=org -w pwd1
```

Migrating the Active Directory Schema to Novell eDirectory Using ICE

While migrating schema from Active Directory to Novell eDirectory using ICE, schema migration for the “Computer” objectclass fails with an ambiguous naming error (-651) error.

To resolve this, complete the following steps:

“[Step 1: Perform the Schema Cache Update operation](#)” on page 530

“[Step 2: Rectify the error LDIF file to eliminate the errors](#)” on page 530

“[Step 3: Import the LDIF File](#)” on page 533

Step 1: Perform the Schema Cache Update operation

While migrating schema from Active Directory to Novell eDirectory using ICE, ensure that you have provided the error log option (-e) of ICE as following:

```
ice -e error_file -S ldap -s Active_Directory_server -p Active_Directory_port
-d Active_Directory_full_admin_context -w Active_Directory_password -D ldap
-s eDirectory_server -p eDirectory_port -d eDirectory_full_admin_context -w
eDirectory_password
```

For example:

```
ice -e err.ldf -S ldap -s activesrv1 -p activeport1 -d cn=admin,o=company -w
activepwd -D ldap -s edirsrv2 -p edirport2 -d cn=admin,o=company -w edirpwd
```

Step 2: Rectify the error LDIF file to eliminate the errors

The failed entry would be present in the err.ldf file as shown below:

```
dn: cn=schema
changetype: modify
delete: objectclasses
objectclasses: ( 2.16.840.1.113719.1.1.6.1.4 NAME 'computer' )
-
```

```

add: objectclasses

objectclasses: ( 2.16.840.1.113719.1.1.6.1.4 NAME 'computer' SUP (device $
user ) STRUCTURAL MAY (operator $ server $ status $ cn $ networkAddress $
local PolicyFlags $ defaultLocalPolicyObject $ machineRole $ location $
netbootInitialization $ netbootGUID $ netbootMachineFilePath $ siteGUID $
operatingSystem $ operatingSystemVersion $ operatingSystemServicePack $
operatingSystemHotfix $ volumeCount $ physicalLocationObject $ dnsHostName
$ policyReplicationFlags $ managedBy $ rIDSetReferences $ catalogs $
netbootSIFFile $ netboot MirrorDataFile ) X-NDS_NOT_CONTAINER '1' X
-NDS_NONREMOVABLE '1' X-NDS_NAME 'Computer' )

```

-

Modify this entry in the error file (err.ldf in the example) to remove the “user” objectclass from the list of superior objectclasses in the definition of the “Computer” objectclass, as shown below: [in emphasis]

```

dn: cn=schema

changetype: modify

delete: objectclasses

objectclasses: ( 2.16.840.1.113719.1.1.6.1.4 NAME 'computer' )

-

add: objectclasses

objectclasses: ( 2.16.840.1.113719.1.1.6.1.4 NAME 'computer' SUP device
STRUCTURAL MAY (operator $ server $ status $ cn $ networkAddress $ local
PolicyFlags $ defaultLocalPolicyObject $ machineRole $ location $
netbootInitialization $ netbootGUID $ netbootMachineFilePath $ siteGUID $
operatingSystem $ operatingSystemVersion $ operatingSystemServicePack $
operatingSystemHotfix $ volumeCount $ physicalLocationObject $ dnsHostName
$ policyReplicationFlags $ managedBy $ rIDSetReferences $ catalogs $
netbootSIFFile $ netbootMirrorDataFile ) X-NDS_NOT_CONTAINER '1' X
-NDS_NONREMOVABLE '1' X-NDS_NAME 'Computer' )

```

-

Step 3: Import the LDIF file

Now, import the modified entry using the following ICE command:

```
ice -S ldif -f LDIF_file -D ldap -s Novell_eDirectory_server -p port_number
-d full_admin_context -w password
```

For example:

```
ice -S ldif -f err.ldf -D ldap -s edirsrv1 -p edirport1 -d cn=admin,o=company
-w pwd1
```

A

NMAS Considerations

This appendix contains the following topics:

- ♦ “Setting Up a Security Container As a Separate Partition” on page 535
- ♦ “Merging Trees with Multiple Security Containers” on page 535

Setting Up a Security Container As a Separate Partition

Novell® Modular Authentication Services (NMAS™) relies on the storage of policies that are global to the Novell eDirectory™ tree. The eDirectory tree is effectively the security domain. The security policies must be available to all servers in the tree.

NMAS places the authentication policies and login method configuration data in the Security container that is created off of the [Root] in NetWare® 5.1 or later eDirectory trees. This information must be readily accessible to all servers that are enabled for NMAS. The purpose of the Security container is to hold global policies that relate to security properties such as login, authentication, and key management.

With NMAS, we recommend that you create the Security container as a separate partition, and that the container be widely replicated. This partition should be replicated as a Read/Write partition only on those servers in your tree that are highly trusted.

NOTE: Because the Security container contains global policies, be careful where writable replicas are placed, because these servers can modify the overall security policies specified in the eDirectory tree. In order for users to log in with NMAS, replicas of the User objects must be on the NMAS server.

Merging Trees with Multiple Security Containers

Special considerations need to be made when merging eDirectory trees where a Security container has been installed in one or both of the trees. Make sure that this is something you really want to do because this procedure has the potential to be a very time-consuming and laborious task.

IMPORTANT: These instructions are complete for trees with Novell Certificate Server™ 2.21 and earlier, Novell Single Sign-on 2.x, and NMAS 2.x.

To merge trees with multiple Security containers:

- 1** In iManager, identify the trees that will be merged.
- 2** Identify which tree will be the source tree and which tree will be the target tree.

Keep in mind these security considerations for the source and target trees:

- ♦ Any certificates signed by the source tree’s Organizational CA must be deleted.
- ♦ The source tree’s Organizational CA must be deleted.
- ♦ All user secrets stored in Novell SecretStore® on the source tree must be deleted.

- ◆ All NMAS login methods in the source tree must be deleted and reinstalled in the target tree.
- ◆ All NMAS users that were in the source tree must be re-enrolled when the trees are merged.
- ◆ All users and servers that were in the source tree must have new certificates created for them when the trees are merged.
- ◆ All users that were in the source tree must have their secrets reinstalled into SecretStore.

If neither the source tree nor the target tree has a container named Security under the root of the tree, or if only one of the trees has the Security container, no further action is required. Otherwise, continue with the remaining procedures in this section.

Product-Specific Operations to Perform prior to Tree Merge

This section contains the following information:

- ◆ “Novell Certificate Server” on page 536
- ◆ “Novell Single Sign-on” on page 537
- ◆ “NMAS” on page 537
- ◆ “Novell Security Domain Infrastructure” on page 538
- ◆ “Other Security-Specific Operations” on page 538

Novell Certificate Server

If Novell Certificate Server (previously known as Public Key Infrastructure Services, or PKIS) has been installed on any server in the source tree, you should complete the following steps.

NOTE: Depending on how the product was used, the objects and items referred to might or might not be present. If the objects and items referred to in a given step are not present in the source tree, you can skip the step.

- 1** Any Trusted Root certificates in the source tree should be installed in the target tree.

Trusted Root certificates are stored in Trusted Root objects, which are contained by Trusted Root containers. Trusted Root containers can be created anywhere within the tree; however, only the Trusted Root certificates that are in the Trusted Root containers within the Security container must be moved manually from the source tree to the target tree.

- 2** Install the Trusted Root certificates in the target tree.

2a Pick a Trusted Root container in the Security container in the source tree.

2b Create a Trusted Root container in the Security container of the target tree with the exact name used in the source tree (Step 2a).

2c In the source tree, open a Trusted Root object in the selected Trusted Root container and export the certificate.

IMPORTANT: Remember the location and filename you choose; you will use them in the next step.

2d In the target tree, create a Trusted Root object in the container that you created in Step 2b. Specify the same name as the source tree and, when prompted for the certificate, specify the file that you created in Step 2c.

2e Delete the Trusted Root object in the source tree.

- 2f** Repeat Step 2c through Step 2e until all Trusted Root objects in the selected Trust Root container have been installed into the target tree.
 - 2g** Delete the Trusted Root container in the source tree.
 - 2h** Continue Step 2a through Step 2f until all Trusted Root containers have been deleted in the source tree.
- 3** Delete the Organizational CA in the source tree.

The Organizational CA object is in the Security container.

IMPORTANT: Any certificates signed by the Organizational CA of the source tree will become unusable following this step. This includes server certificates and user certificates that have been signed by the Organizational CA of the source tree.

- 4** Delete every Key Material object (KMO) in the source tree that has a certificate signed by the Organizational CA of the source tree.

Key Material objects in the source tree with certificates signed by other CAs will continue to be valid and do not need to be deleted.

If you are uncertain about the identity of the signing CA for any Key Material object, look at the Trusted Root Certificate section of the Certificates tab in the Key Material object property page.

- 5** Delete all user certificates in the source tree that have been signed by the Organizational CA of the source tree.

If users in the source tree have already exported their certificates and private keys, those exported certificates and keys will continue to be usable. Private keys and certificates that are still in eDirectory will no longer be usable after you perform Step 3.

For each user with certificates, open the properties of the User object. Under the Certificates section of the Security tab, a table lists all the certificates for the user. All of those certificates with the Organizational CA as the issuer must be deleted.

User certificates will be present in the source tree only if Novell Certificate Server 2.0 or later has been installed on the server that hosts the Organizational CA in the source tree.

Novell Single Sign-on

If Novell Single Sign-on has been installed on any server in the source tree, you should delete all Novell Single Sign-on secrets for users in the source tree.

For every user using Novell Single Sign-on in the source tree, open the properties of the User object. All of the user's secrets will be listed under the SecretStore section of the Security tab. Delete all listed secrets.

NOTE: Depending on how the product was used, the objects and items referred to might or might not be present. If the objects and items referred to are not present in the source tree, you can skip this step.

NMAS

If NMAS has been installed on any server in the source tree, you should complete the following steps.

NOTE: Depending on how the product was used, the objects and items referred to might or might not be present. If the objects and items referred to are not present in the source tree, you can skip the step.

- 1** In the target tree, install any NMAS login methods that were in the source tree but not in the target tree.

To ensure that all of the necessary client and server login components are properly installed in the target tree, we recommend that you install all new login methods using original Novell or vendor-supplied sources.

Although methods *can* be reinstalled from existing server files, establishing a clean installation from Novell or vendor-supplied packages is typically simpler and more reliable.

- 2** To ensure that the previously established login sequences in the source tree are available in the target tree, migrate the desired login sequences.
 - 2a** In ConsoleOne, select the Security container in the source tree.
 - 2b** Right-click the Login Policy object > select Properties.
 - 2c** For each login sequence listed in the Defined Login Sequences drop-down list, note the Login Methods used (listed in the right pane).
 - 2d** Select the Security container in the target tree and replicate the login sequences using the same login methods note in Step 2c.
 - 2e** Click OK when you are finished.
- 3** Delete NMAS login security attributes in the source tree.
 - 3a** In the Security container of the source tree, delete the Login Policy object.
 - 3b** In the Authorized Login Methods container of the source tree, delete all login methods.
 - 3c** Delete the Authorized Login Methods container in the source tree.
 - 3d** In the Authorized Post-Login Methods container of the source tree, delete all login methods.
 - 3e** Delete the Authorized Post-Login Methods container in the source tree.

Novell Security Domain Infrastructure

If Novell Certificate Server 2.x or later, Novell Single Sign-on, NMAS, NetWare 5.1 or later, or eDirectory 8.5 or later has been installed on any server in the source tree, the Novell Security Domain Infrastructure (SDI) will be installed. If SDI has been installed, you should complete the following steps.

NOTE: Depending on how the product was used, the objects and items referred to might or might not be present. If the objects and items referred to are not present in the source tree, you can skip the step.

- 1** Delete the W0 object and the KAP container in the source tree.
The KAP container is in the Security container. The W0 object is in the KAP container.
- 2** On all servers in the source tree, delete the Security Domain Infrastructure (SDI) keys by deleting the `sys:\system\nici\nicisdi.key` file.

IMPORTANT: Make sure that you delete this file on *all* servers in the source tree.

Other Security-Specific Operations

If a Security container exists in the source tree, delete the Security container before you merge the trees.

Performing the Tree Merge

eDirectory trees are merged using the `ndsmerge` utility. For more information, see [Chapter 8, “Merging Novell eDirectory Trees,”](#) on page 189 and [Appendix B, “Novell eDirectory UNIX Commands and Usage,”](#) on page 541.

Product-Specific Operations to Perform after the Tree Merge

This section contains the following information:

- ♦ [“Novell Security Domain Infrastructure”](#) on page 539
- ♦ [“Novell Certificate Server”](#) on page 539
- ♦ [“Novell Single Sign-On”](#) on page 539
- ♦ [“NMAS”](#) on page 540

Novell Security Domain Infrastructure

If the `W0` object existed in the target tree before the merge, the Security Domain Infrastructure (SDI) keys used by the servers that formerly resided in the target tree must be installed in the servers that formerly resided in the source tree.

The easiest way to accomplish this is to install Novell Certificate Server 2.52 or later on all servers formerly in the source tree that held SDI keys (the `sys:\system\nici\nicisdi.key` file). This should be done even if the Novell Certificate Server has already been installed on the server.

If the `W0` object did not exist in the target tree before the merge but did exist in the source tree, the SDI must be reinstalled in the resulting tree.

The easiest way to accomplish this is to install Novell Certificate Server 2.52 or later on the servers in the resulting tree. Novell Certificate Server must be installed on the servers formerly in the source tree that held SDI keys (the `sys:\system\nici\nicisdi.key` file). It can also be installed on other servers in the resulting tree.

For more information on installing Novell Certificate Server, see the [*Novell Certificate Server Administration Guide*](http://www.novell.com/documentation/lg/crt27/index.html) (<http://www.novell.com/documentation/lg/crt27/index.html>).

Novell Certificate Server

If you are using Novell Certificate Server, then after the tree merge reissue certificates for servers and users that were formerly in the source tree, as necessary.

We recommend that you install Novell Certificate Server 2.52 or later on all servers that hold a replica of the partition containing a User object.

In order to issue a certificate for a server, Novell Certificate Server 2.52 or later must be installed.

Novell Certificate Server 2.52 or later must be installed on the server that hosts the Organizational CA. For more information, see the [*Novell Certificate Server Administration Guide*](http://www.novell.com/documentation/lg/crt27/index.html) (<http://www.novell.com/documentation/lg/crt27/index.html>).

Novell Single Sign-On

If you are using Novell Single Sign-on, after the tree merge you should re-create SecretStore secrets for users who were formerly in the source tree, as necessary.

NMAS

If you are using NMAS, after the tree merge you should re-enroll NMAS users who were formerly in the source tree, as necessary.

For more information, see the *Novell Modular Authentication Service Administration Guide* (<http://www.novell.com/documentation/lg/nmas23/index.html>).

B

Novell eDirectory UNIX Commands and Usage

This chapter lists the following Novell® eDirectory™ 8.7.3 for Linux, Solaris, AIX, and HP-UX commands and their usage:

- ♦ “General Commands” on page 541
- ♦ “LDAP-Specific Commands” on page 543

General Commands

This section gives a list of the eDirectory commands on UNIX and their usage.

NOTE: For more information on the usage of commands, see the command man pages.

Command	Description	Usage
ndsbackup	Creates eDirectory object archives and adds or extracts eDirectory objects	<pre>ndsbackup c [fevwXR] [ndsbackupfile] [exclude_file] [Replica_servername] [-a admin_user] [-I include_file]... [eDirectoryobject] ndsbackup r [fevwXR] [ndsbackupfile] [exclude_file] [Replica_servername] [-a admin_user] [-I include_file]... [eDirectoryobject] ndsbackup t [fevXR] [ndsbackupfile] [exclude_file] [Replica_servername] [-a admin_user] [-I include_file]... [eDirectoryobject] ndsbackup x [fevwXR] [ndsbackupfile] [exclude_file] [Replica_servername] [-a admin_user] [-I include_file]... [eDirectoryobject] ndsbackup s [evXR] [exclude_file] [Replica_servername] [-a admin_user] [-I include_file]... [eDirectoryobject]</pre>

Command	Description	Usage
ndsconfig	Configures Novell eDirectory	<pre>ndsconfig new [-m module_name] [-i] [-S server_name] [-t tree_name] [-n context] [-d path_for_dib] [-L ldap_port] [-l ssl_port] [-o http_port] [-O https_port] [-e] -a admin_name ndsconfig def [-m module_name] [-i] [-S server_name] [-t tree_name] [-n context] [-d path_for_dib] [-L ldap_port] [-l ssl_port] [-o http_port] [-O https_port] [-e] -a admin_name ndsconfig add [-m module_name] [-S server_name] [-t tree_name] [-p IP_address] [-n context] [-d path_for_dib] [-L ldap_port] [-l ssl_port] [-o http_port] [-O https_port] [-e] -a admin_name ndsconfig rm [-a admin_name] ndsconfig upgrade [-a admin_name] ndsconfig {set valuelist get [paramlist] get help [paramlist]}</pre>
ndsd	NDS [®] daemon	<pre>/usr/sbin/ndsd [-f config_file]</pre> <p>NOTE: Before rebooting Solaris, ndsd needs to be stopped. Enter <code>/etc/init.d/ndsd stop</code>.</p>
ndsmonitor	Monitors and diagnoses the servers in the Novell eDirectory tree using HTTP	<pre>/usr/bin/ndsmonitor [-l -u]</pre>
ndslogin	Diagnostic utility to verify Novell eDirectory authentication	<pre>ndslogin [-t treename] [-h hostname[:port]] [-p password] [-s] userFDN</pre>
ndsmerge	Utility to merge two Novell eDirectory trees	<pre>ndsmerge [-m target_tree target_admin source_admin [target_container]] [-c] [-t] [-r target_tree source_admin]</pre>
ndsrepair	Utility to repair and correct problems with the Novell eDirectory database, such as records, schema, bindery objects, and external references.	<pre>ndsrepair { -U -E -C -P [-Ad] -S [- Ad] -N -T -J entry_id --version} [- F filename] [-A yes no] [-O yes no] ndsrepair -R [-l yes no] [-u yes no] [-m yes no] [-i yes no] [-f yes no] [-d yes no] [- t yes no] [-o yes no] [-r yes no] [-v yes no] [-c yes no] [-F filename] [-A yes no] [-O yes no]</pre>
ndssch	Novell eDirectory schema extension utility	<pre>ndssch [-h hostname[:port]][-t treename] adminFDN schemafile ... ndssch [-h hostname[:port]][-t treename] [-d adminFDN schemafile [schema_description] ...</pre>
ndssnmp	SNMP services module for Novell eDirectory.	<pre>/usr/bin/ndssnmp</pre>

Command	Description	Usage
ndssnmpconfig	SNMP trap configuration utility	<code>ndssnmpconfig -h [hostname[:port]] -p password -a userFDN -c command</code>
ndssnmpsa	eDirectory SNMP subagent daemon	<code>/usr/bin/ndssnmpsa</code>
ndsstat	Utility that displays the server information	<code>ndsstat [-h hostname[:port]] {-r -s}</code>
ndstrace	Utility that displays the server debug messages	<code>/usr/bin/ndstrace</code> <code>ndstrace [-l -u -c 'command1;.....']</code>
nds-uninstall	Utility to uninstall Novell eDirectory	<code>nds-uninstall [-c component1 [-c component2]...] [-h]</code>
nldap	LDAP services for NDS daemon	<code>/usr/sbin/nldap</code>
nmasinst	NMAS™ configuration utility	<code>nmasinst -i admin.context treename</code> <code>nmasinst -addmethod admin.context_treename config.txt_path</code>
npki	Novell Public Key Infrastructure Services	<code>/usr/sbin/npki</code>

LDAP-Specific Commands

Command	Description	Usage
ldapconfig	Utility to configure LDAP Server and LDAP Group objects	<code>ldapconfig get [...] set attribute-value-list [-t treename -p hostname[:port]] [-w password] [-a user FDN] [-f]</code> <code>ldapconfig [-t treename -p hostname[:port]] [-w password] [-a user FDN] [-V] [-R] [-H] [-f] -v attribute,attribute2...</code> <code>ldapconfig [-t treename -p hostname[:port]] [-w password] [-a admin FDN] [-V] [-R] [-H] [-f] -s attribute=value,...</code>
ldapadd	Add or modify entries from an LDAP server	<code>ldapmodify [-a] [-c] [-C] [-M] [-P] [-r] [-n] [-v] [-F] [-l limit] [-M[M]] [-d debuglevel] [-e key filename] [-D binddn] [[-W] [-w passwd]] [-h ldaphost] [-p ldap-port] [-P version] [-Z[Z]] [-f file]</code> <code>ldapadd [-c] [-C] [-l] [-M] [-P] [-r] [-n] [-v] [-F] [-l limit] [-M[M]] [-d debuglevel] [-e key filename] [-D binddn] [[-W] [-w passwd]] [-h ldaphost] [-p ldapport] [-P version] [-Z[Z]] [-f file]</code>
ldapdelete	Delete entries from an LDAP server	<code>ldapdelete [-n] [-v] [-c] [-r] [-l] [-C] [-M] [-d debuglevel] [-e key filename] [-f file] [-D binddn] [[-W] [-w passwd]] [-h ldaphost] [-p ldapport] [-Z[Z]] [dn]...</code>

Command	Description	Usage
ldapmodify	Add or modify entries from an LDAP server.	<pre> ldapmodify [-a] [-c] [-C] [-M] [-P] [-r] [-n] [-v] [-F] [-l limit] [-M[M]] [-d debuglevel] [-e key filename] [-D binddn] [[-W] [-w passwd]] [-h ldaphost] [-p ldap-port] [-P version] [-Z[Z]] [-f file] ldapadd [-c] [-C] [-l] [-M] [-P] [-r] [-n] [- v] [-F] [-l limit] [-M[M]] [-d debuglevel] [-e key filename] [-D binddn] [[-W] [-w passwd]] [-h ldaphost] [-p ldapport] [-P version] [-Z[Z]] [-f file] </pre>
ldapmodrdn	LDAP modify entry Relative Distinguished Name (RDN) tool.	<pre> ldapmodrdn [-r] [-n] [-v] [-c] [-C] [-l] [-M] [-s newsuperior] [-d debuglevel] [-e key filename] [-D binddn] [[-W] [-w passwd]] [- h ldaphost] [-p ldapport] [-Z[Z]] [-f file] [dn newrdn] </pre>
ldapsearch	The LDAP search tool	<pre> ldapsearch [-n] [-u] [-v] [-t] [-A] [-T] [-C] [-V] [-M] [-P] [-L] [-d debuglevel] [-e key filename] [-f file] [-D binddn] [[-W] [-w bindpasswd]] [-h ldaphost] [-p ldapport] [-b searchbase] [-s scope] [-a deref] [-l time limit] [-z size limit] [-Z[Z]] filter [attrs....] </pre>
ndsindex	Utility to create, list, suspend, resume, or delete Novell eDirectory database indexes.	<pre> ndsindex list [-h <hostname>] [-p <port>] -D <bind DN> -W [-w <password>] [-l limit] -s <eDirectory Server DN> [-Z[Z]] [<indexName1>, <indexName2>.....] ndsindex add [-h <hostname>] [-p <port>] -D <bind DN> -W [-w <password>] [-l limit] -s <eDirectory Server DN> [-Z[Z]] <indexDefinintion1> [<indexDefinintion2>.....] ndsindex delete [-h <hostname>] [-p <port>] - D <bind DN> -W [-w <password>] [-l limit] -s <eDirectory Server DN> [-Z[Z]] <indexName1> [<indexName2>.....] ndsindex resume [-h <hostname>] [-p <port>] - D <bind DN> -W [-w <password>] [-l limit] -s <eDirectory Server DN> [-Z[Z]] <indexName1> [<indexName2>.....] ndsindex suspend [-h <hostname>] [-p <port>] -D <bind DN> -W [-w <password>] [-l limit] - s <eDirectory Server DN> [-Z[Z]] <indexName1> [<indexName2>.....] </pre>

C

Configuring OpenSLP for eDirectory

This appendix provides information for network administrators on the proper configuration of OpenSLP for Novell® eDirectory™ installations without the Novell Client™.

- ◆ “Service Location Protocol” on page 545
- ◆ “SLP Fundamentals” on page 545
- ◆ “Configuration Parameters” on page 547
- ◆ “Open Enterprise Server SLP Configuration (Linux)” on page 548

Service Location Protocol

OpenSLP is an open-source implementation of the IETF Service Location Protocol Version 2.0 standard, which is documented in [IETF Request-For-Comments \(RFC\) 2608](http://www.ietf.org/rfc/rfc2608.txt?number=2608) (<http://www.ietf.org/rfc/rfc2608.txt?number=2608>).

In addition to implementing the SLP v2 protocol, the interface provided by OpenSLP source code is an implementation of another IETF standard for programmatically accessing SLP functionality, documented in [RFC 2614](http://www.ietf.org/rfc/rfc2614.txt?number=2614) (<http://www.ietf.org/rfc/rfc2614.txt?number=2614>).

To fully understand the workings of SLP, we recommend that you read these two documents and internalize them. They are not necessarily light reading, but they are essential to the proper configuration of SLP on an intranet.

For more information on the OpenSLP project, see the [OpenSLP](http://www.OpenSLP.org) (<http://www.OpenSLP.org>) Web site and the [SourceForge](http://sourceforge.net/projects/openslp) (<http://sourceforge.net/projects/openslp>) Web site. The OpenSLP Web site provides several documents that contain valuable configuration tips. Many of these are incomplete at the time of this writing.

SLP Fundamentals

Service Location Protocol specifies three components:

- ◆ The user agent (UA)
- ◆ The service agent (SA)
- ◆ The directory agent (DA)

The user agent's job is to provide a programmatic interface for clients to query for services, and for services to advertise themselves. A user agent contacts a directory agent to query for registered services of a specified service class and within a specified scope.

The service agent's job is to provide persistent storage and maintenance points for local services that have registered themselves with SLP. The service agent essentially maintains an in-memory database of registered local services. In fact, a service cannot register with SLP unless a local SA

is present. Clients can discover services with only a UA library, but registration requires an SA, primarily because an SA must reassert the existence of registered services periodically in order to maintain the registration with listening directory agents.

The directory agent's job is to provide a long-term persistent cache for advertised services, and to provide a point of access for user agents to look up services. As a cache, the DA listens for SAs to advertise new services, and caches those notifications. Over a short time, a DA's cache will become more complete. Directory agents use an expiration algorithm to expire cache entries. When a directory agent comes up, it reads its cache from persistent storage (generally a hard drive), and then begins to expire entries according to the algorithm. When a new DA comes up, or when a cache has been deleted, the DA detects this condition and sends out a special notification to all listening SAs to dump their local databases so the DA can quickly build its cache.

In the absence of any directory agents, the UA will resort to a general multicast query that SAs can respond to, building a list of the requested services in much the same manner that DAs use to build their cache. The list of services returned by such a query is an incomplete and much more localized list than that provided by a DA, especially in the presence of multicast filtering, which is done by many network administrators, limiting broadcasts and multicasts to only the local subnet.

In summary, everything hinges on the directory agent that a user agent finds for a given scope.

Novell Service Location Providers

The Novell version of SLP takes certain liberties with the SLP standard in order to provide a more robust service advertising environment, but it does so at the expense of some scalability.

For example, in order to improve scalability for a service advertising framework, we want to limit the number of packets that are broadcast or multicast on a subnet. The SLP specification manages this by imposing restrictions on service agents and user agents regarding directory agent queries. The first directory agent discovered that services the desired scope is the one that a service agent (and consequently, local user agents) will use for all future requests on that scope.

The Novell SLP implementation actually scans all of the directory agents it knows about looking for query information. It assumes a 300-millisecond round trip time is too long, so it can scan 10 servers in about 3 to 5 seconds. This doesn't need to be done if SLP is configured correctly on the network, and OpenSLP assumes the network is in fact configured correctly for SLP traffic. OpenSLP's response timeout values are greater than that of Novell's SLP service provider, and it limits the number of directory agents to the first one that responds, whether or not that agent's information is accurate and complete.

IMPORTANT: Novell SLP directory agents share information between directory agents in context through eDirectory. With OpenSLP, each directory agent is completely separate. An OpenSLP directory agent only knows about the services it is told about, and directory agents are not synchronized.

You can run both OpenSLP and Novell SLP directory agents in the same network, but OpenSLP directory agents won't synchronize with Novell SLP directory agents, and Novell SLP directory agents won't synchronize with OpenSLP directory agents.

User Agents

A user agent takes the physical form of a static or dynamic library that is linked into an application. It allows the application to query for SLP services.

User agents follow an algorithm to obtain the address of a directory agent to which queries will be sent. Once they obtain a DA address for a specified scope, they continue to use that address for that scope until it no longer responds, at which time they obtain another DA address for that scope. User agents locate a directory agent address for a specified scope by:

1. Checking to see if the socket handle on the current request is connected to a DA for the specified scope. (If the request happens to be a multipart request, there may already be a cached connection present on the request.)
2. Checking its local known DA cache for a DA matching the specified scope.
3. Checking with the local SA for a DA with the specified scope (and adding new addresses to the cache).
4. Querying DHCP for network-configured DA addresses that match the specified scope (and adding new addresses to the cache).
5. Multicasting a DA discovery request on a well-known port (and adding new addresses to the cache).

The specified scope is “default” if not specified. That is, if no scope is statically defined in the SLP configuration file, and no scope is specified in the query, then the scope used is the word “default”. It should also be noted that eDirectory never specifies a scope in its registrations. That's not to say the scope always used with eDirectory is “default.” In fact, if there is a statically configured scope, that scope becomes the default scope for all local UA requests and SA registrations in the absence of a specified scope.

Service Agents

Service agents take the physical form of a separate process on the host machine. In the case of Win32, `slpd.exe` runs as a service on the local machine. User agents query the local service agent by sending messages to the loop-back address on a well-known port.

A service agent locates and caches directory agents and their supported scope list by sending a DA discovery request directly to potential DA addresses by:

1. Checking all statically configured DA addresses (and adding new ones to the SA's known DA cache).
2. Requesting a list of DA's and scopes from DHCP (and adding new ones to the SA's known DA cache).
3. Multicasting a DA discovery request on a well-known port (and adding new ones to the SA's known DA cache).
4. Receiving DA advertising packets that are periodically broadcast by DAs (and adding new ones to the SA's known DA cache).

Since a user agent always queries the local service agent first, this is important, as the local service agent's response will determine whether or not the user agent continues to the next stage of discovery (in this case DHCP-- see steps 3 and 4 in [“User Agents” on page 546.](#)).

Configuration Parameters

Certain configuration parameters in the `%systemroot%/slp.conf` file control DA discovery as well:

```
net.slp.useScopes = <comma delimited scope list>
net.slp.DAAddresses = <comma delimited address list>
net.slp.passiveDADetection = <"true" or "false">
net.slp.activeDADetection = <"true" or "false">
net.slp.DAActiveDiscoveryInterval = <0, 1, or a number of seconds>
```

The `useScopes` option indicates which scopes the SA will advertise into, and which scopes queries will be made to in the absence of a specific scope on the registration or query made by the service or client application. Because eDirectory always advertises into and queries from the default scope, this list will become the default scope list for all eDirectory registrations and queries.

The DAAddresses option is a comma-delimited list of dotted decimal IP addresses of DAs that should be preferred to all others. If this list of configured DAs does not support the scope of a registration or query, then SAs and UAs will resort to multicast DA discovery, unless such discovery is disabled.

The passiveDADetection option is True by default. Directory agents will periodically broadcast their existence on the subnet on a well-known port if configured to do so. These packets are termed DAAdvert packets. If this option is set to False, all broadcast DAAdvert packets are ignored by the SA.

The activeDADetection option is also True by default. This allows the SA to periodically broadcast a request for all DAs to respond with a directed DAAdvert packet. A directed packet is not broadcast, but sent directly to the SA in response to these requests. If this option is set to False, no periodic DA discovery request is broadcast by the SA.

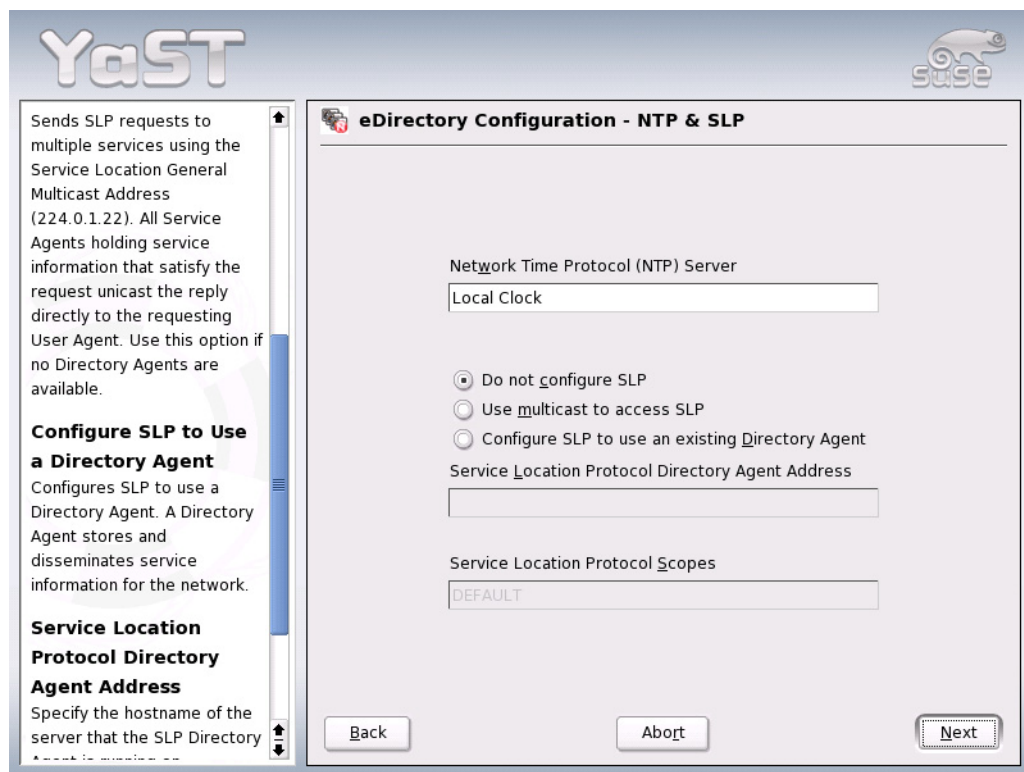
The DAActiveDiscoveryInterval option is a try-state parameter. The default value is 1, which is a special value meaning that the SA should only send out one DA discovery request upon initialization. Setting this option to 0 has the same effect as setting the activeDADetection option to “false.” Any other value is a number of seconds between discovery broadcasts.

These options, when used properly, can ensure an appropriate use of network bandwidth for service advertising. In fact, the default settings are designed to optimize scalability on an average network.

Open Enterprise Server SLP Configuration (Linux)

During the eDirectory configuration portion of the Open Enterprise Server for Linux installation, you will see the following screen:

Figure 37 eDirectory Configuration for Open Enterprise Server



You can choose Do Not Configure SLP if you have three or fewer eDirectory servers in your tree.

Choose Use Multicast to Access SLP to request SLP information using multicast packets. This option sends SLP requests to multiple services using the Service Location General Multicast Address (224.0.1.22). All Service Agents holding service information that satisfies the request will unicast the reply directly to the requesting User Agent. Use this option in environments that have no established directory agents.

Choose Configure SLP to Use an Existing Directory Agent if your environment has established directory agents. In the Service Location Protocol Directory Agent Address field, specify the hostname of an external server that an SLP directory agent is running on. Do not specify the local host.

If you want to use existing scopes (sets of services), specify them in the Service Location Protocol Scopes field. Separate multiple scopes with commas (for example: myScope1, myScope2, myScope3). The default value for this field is DEFAULT.

D

How Novell eDirectory Works with DNS

If a client asks a server to resolve a fully qualified name (for example, `admin.novell.novell_inc`) that does not exist in the Novell® eDirectory™ tree, or if you use a standalone application such as Novell iManager for UNIX or the eDirectory install application to resolve a name in the tree and you don't have a server to talk to yet, eDirectory uses service discovery protocols to resolve the name. Service discovery protocols are a class of network applications that allow distributed components to find and use needed services within a network.

eDirectory has traditionally used SAP and SLP to search for and advertise network services. DNS was added as a discovery protocol in eDirectory 8.7.1. This added functionality means that if you ask for a tree name that eDirectory doesn't understand (either because you are talking to a server that doesn't hold a copy of the tree or you are using a stand-alone application), the machine trying to do the discovery—whether it's a machine running a stand-alone application, a JClient application such as Novell iManager or ConsoleOne®, or a server—uses eDirectory's discovery protocols, in the following order:

1. Domain Name System (DNS)
2. Service Location Protocol (SLP)
3. Service Advertising Protocol (SAP)

When using the DNS protocol, eDirectory takes the name as it was passed (for example, a server name such as `prod_server4.provo.novell.novell_inc`), and tries to resolve the entire name just as it is. eDirectory then appends each name in the discovery machine's DNS search list, and asks the machine's DNS sever if it has an address for that name. For example, if the discovery machine's DNS search list included `dev.novell.com` and `test.novell.com`, eDirectory would search for `prod_server4.provo.novell.novell_inc.dev.novell.com` and `prod_server4.provo.novell.novell_inc.test.novell.com`.

Then eDirectory takes components off the name that was passed to it. For example, if trying to resolve `prod_server4.provo.novell.novell_inc`, eDirectory tries `provo.novell.novell_inc`, then `novell.novell_inc`, then `novell_inc`. eDirectory does that for each of the different search contexts until eventually it tries the single component that is the tree root. The client will attempt each of the addresses until it successfully makes a connection. It does the attempts using the ordering of records returned from the DNS server. It doesn't matter what code revision the servers in the replica ring are running as long as the machine trying to do the discovery is running eDirectory 8.7.1 or later.

We recommend putting your eDirectory tree name in DNS using an A, AAAA, or Service (SRV) resource record under the DNS domain the clients are going to use to resolve names. If you use A or AAAA records, the eDirectory servers must be running on the default 524 port. If the servers are using any other port, use an SRV record.

In the following sample resource records, `novell_inc` is the tree name and `provo.novell.com` is the DNS search context:

Record	Example
A	novell_inc.provo.novell.com. IN A 192.168.1.2
AAAA	novell_inc.provo.novell.com. IN AAAA 4321:0:1:2:3:4:567:89ab
SRV	_ldap_tcp.novell_inc.provo.novell.com. SRV 0 0 389 server1.novell_inc.provo.novell.com SRV 10 0 389 server2.novell_inc.provo.novell.com

For redundancy, or to specify multiple hosts (servers in the replica ring) to the A record, create more than one A record. eDirectory will look at all of them. For more information on A, AAAA, and SRV records, see [DNS Resource Records \(http://www.dns.net/dnsrd/rr.html\)](http://www.dns.net/dnsrd/rr.html).

You don't need to point the DNS server record entry to something that holds a corresponding partition root. As soon as the discovery machine can talk to a server that knows about the tree, it can walk up and down the tree to resolve the name. For example, if you put novell_inc in your DNS, you don't have to put in any of the servers that hold novell_inc root. All you need to do is point to any server in the novell_inc tree, because after you get to that server in the tree, that server will refer you around the tree.