Administration Guide

eDirectory™ 8.8 SP7

April 27, 2012

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. See the Novell International Trade Services Web page (http://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 © 2012 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.
1800 South Novell Place
Provo, UT 84606
U.S.A.
www.novell.com

Online Documentation: To access the latest online documentation for this and other Novell products, see the Novell Documentation Web page (http://www.novell.com/documentation).

Novell Trademarks

For Novell trademarks, see the Novell Trademark and Service Mark list (http://www.novell.com/company/legal/trademarks/tmlist.html).

Third-Party Materials

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

Contents

About This Guide 15

1 Understanding Novell eDirectory 17

1.1 Ease of Management through Novell iManager . . . 18
1.1.1 Powerful Tree Structure . . . 18
1.1.2 Web-Based Management Utility . . . 20
1.1.3 Single Login and Authentication . . . 21
1.2 Object Classes and Properties . . . 21
1.2.1 List of Objects . . . 21
1.2.2 Container Object Classes . . . 23
1.2.3 Leaf Object Classes . . . 27
1.3 Context and Naming . . . 43
1.3.1 Distinguished Name . . . 44
1.3.2 Typeful Name . . . 44
1.3.3 Name Resolution . . . 44
1.3.4 Current Workstation Context . . . 45
1.3.5 Leading Period . . . 45
1.3.6 Relative Naming . . . 45
1.3.7 Trailing Periods . . . 45
1.3.8 Context and Naming on Linux and UNIX . . . 46
1.4 Schema . . . 46
1.4.1 Schema Management . . . 47
1.4.2 Schema Classes, Attributes, and Syntaxes . . . 47
1.4.3 Understanding Mandatory and Optional Attributes . . . 51
1.4.4 Sample Schema . . . 52
1.4.5 Designing the Schema . . . 52
1.5 Partitions . . . 52
1.5.1 Partitions . . . 53
1.5.2 Distributing Replicas for Performance . . . 54
1.5.3 Partitions and WAN Links . . . 54
1.6 Replicas . . . 55
1.6.1 Replica Types . . . 56
1.6.2 Filtered Replicas . . . 59
1.7 Server Synchronization in the Replica Ring . . . 60
1.8 Access to Resources . . . 60
1.9 eDirectory Rights . . . 61
1.9.1 Trustee Assignments and Targets . . . 61
1.9.2 eDirectory Rights Concepts . . . 61
1.9.3 Default Rights for a New Server . . . 66
1.9.4 Delegated Administration . . . 67
1.9.5 Administering Rights . . . 67

2 Designing Your Novell eDirectory Network 73

2.1 eDirectory Design Basics . . . 73
2.1.1 Network Layout . . . 73
2.1.2 Organizational Structure . . . 74
2.1.3 Preparing for eDirectory Design . . . 74
2.2 Designing the eDirectory Tree . . . 74
2.2.1 Creating a Naming Standards Document . . . 74
2.2.2 Designing the Upper Layers of the Tree . . . 77

2.2.3 Designing the Lower Layers of the Tree . . . 79
2.3 Guidelines for Partitioning Your Tree . . . 80
2.3.1 Determining Partitions for the Upper Layers of the Tree . . . 80
2.3.2 Determining Partitions for the Lower Layers of the Tree . . . 81
2.3.3 Determining Partition Size . . . 81
2.3.4 Considering Network Variables . . . 81
2.4 Guidelines for Replicating Your Tree . . . 82
2.4.1 Workgroup Needs . . . 82
2.4.2 Fault Tolerance . . . 82
2.4.3 Determining the Number of Replicas . . . 83
2.4.4 Replicating the Tree Partition . . . 83
2.4.5 Replicating for Administration . . . 83
2.4.6 Managing WAN Traffic . . . 84
2.5 Planning the User Environment . . . 84
2.5.1 Reviewing Users' Needs . . . 84
2.5.2 Creating Accessibility Guidelines . . . 85
2.6 Designing eDirectory for e-Business . . . 85
2.7 Understanding the Novell Certificate Server . . . 86
2.7.1 Rights Required to Perform Tasks on Novell Certificate Server . . . 86
2.7.2 Ensuring Secure eDirectory Operations on Linux, Solaris, and AIX Systems . . . 87
2.8 Synchronizing Network Time . . . 90
2.8.1 Synchronizing Time on Windows 2000 Servers . . . 91
2.8.2 Synchronizing Time on Linux, Solaris, or AIX Systems . . . 91
2.8.3 Verifying Time Synchronization . . . 91

3 Managing Objects 93

3.1 General Object Tasks . . . 93
3.1.1 Browsing the eDirectory Tree . . . 94
3.1.2 Creating an Object . . . 96
3.1.3 Modifying an Object's Properties . . . 97
3.1.4 Copying Objects . . . 97
3.1.5 Moving Objects . . . 97
3.1.6 Deleting Objects . . . 97
3.1.7 Renaming Objects . . . 98
3.2 Managing User Accounts . . . 98
3.2.1 Creating and Modifying User Accounts . . . 98
3.2.2 Setting Up Optional Account Features . . . 99
3.2.3 Setting Up Login Scripts . . . 102
3.2.4 Login Time Restrictions for Remote Users . . . 103
3.2.5 Deleting User Accounts . . . 104
3.3 Configuring Role-Based Services . . . 105
3.3.1 Defining RBS Roles . . . 106
3.3.2 Defining Custom RBS Tasks . . . 109
3.4 Synchronization . . . 110
3.4.1 Features of Synchronization . . . 111
3.4.2 Normal or Replica Synchronization . . . 113
3.4.3 Priority Sync . . . 115

4 Managing the Schema 123

4.1 Extending the Schema . . . 124
4.1.1 Creating a Class . . . 124
4.1.2 Deleting a Class . . . 124
4.1.3 Creating an Attribute . . . 125
4.1.4 Adding an Optional Attribute to a Class . . . 125
4.1.5 Deleting an Attribute . . . 126
4.1.6 Creating an Auxiliary Class . . . 126

4.1.7 Extending an Object with the Properties of an Auxiliary Class . . . 126
4.1.8 Modifying an Object's Auxiliary Properties . . . 127
4.1.9 Deleting Auxiliary Properties from an Object . . . 127
4.2 Viewing the Schema . . . 127
4.2.1 Viewing Class Information . . . 128
4.2.2 Viewing Attribute Information . . . 128
4.3 Manually Extending the Schema . . . 128
4.3.1 Extending the Schema on Windows . . . 128
4.3.2 Extending the Schema on Linux, Solaris, or AIX Systems . . . 129
4.4 Schema Flags Added in eDirectory 8.7 . . . 130
4.5 Using the Client to Perform Schema Operations . . . 132
4.5.1 Using the DSSchema eMTool . . . 132
4.5.2 DSSchema eMTool Options . . . 133

5 Managing Partitions and Replicas 135

5.1 Creating a Partition . . . 136
5.2 Merging a Partition . . . 136
5.3 Moving Partitions . . . 137
5.4 Cancelling Create or Merge Partition Operations . . . 139
5.5 Administering Replicas . . . 139
5.5.1 Adding a Replica . . . 139
5.5.2 Deleting a Replica . . . 140
5.5.3 Changing a Replica Type . . . 141
5.6 Setting Up and Managing Filtered Replicas . . . 142
5.6.1 Using the Filtered Replica Wizard . . . 142
5.6.2 Defining a Partition Scope . . . 143
5.6.3 Setting Up a Server Filter . . . 144
5.7 Viewing Partitions and Replicas . . . 145
5.7.1 Viewing the Partitions on a Server . . . 145
5.7.2 Viewing a Partition’s Replicas . . . 145
5.7.3 Viewing Information about a Partition . . . 145
5.7.4 Viewing Partition Hierarchy . . . 146
5.7.5 Viewing Information about a Replica . . . 146

6 Novell eDirectory Management Utilities 147

6.1 Novell Import Conversion Export Utility . . . 147
6.1.1 Using the Novell iManager Import Convert Export Wizard . . . 148
6.1.2 Using the Command Line Interface . . . 155
6.1.3 Conversion Rules . . . 173
6.1.4 LDAP Bulk Update/Replication Protocol . . . 181
6.1.5 Improving the Speed of LDIF Imports . . . 182
6.2 Index Manager . . . 184
6.2.1 Creating an Index . . . 185
6.2.2 Deleting an Index . . . 185
6.2.3 Taking an Index Offline . . . 186
6.2.4 Managing Indexes on Other Servers . . . 186
6.2.5 Using the Novell Import Conversion Export Utility to Manage Indexes . . . 186
6.3 Predicate Data . . . 188
6.3.1 Managing Predicate Data . . . 189
6.4 eDirectory Service Manager . . . 189
6.4.1 Using the Client Service Manager eMTool . . . 189
6.4.2 Using the Service Manager Plug-In to Novell iManager . . . 190

7 Offline Bulkload Utility 193

7.1 Using ldif2dib for Bulkloading . . . 193
7.2 Multiple Instances . . . 195
7.3 Tuning ldif2dib . . . 195
7.3.1 Tuning the Cache . . . 195
7.3.2 Transaction Size . . . 196
7.3.3 Index . . . 196
7.3.4 Block Cache Percent . . . 196
7.3.5 Check Point Interval . . . 196
7.4 Limitations . . . 196
7.4.1 Schema . . . 196
7.4.2 ACL Templates . . . 197
7.4.3 Options . . . 197
7.4.4 Simple Password LDIF . . . 197
7.4.5 Custom Classes . . . 197
7.5 Caveats . . . 198
7.5.1 Duplicate Entries . . . 198
7.5.2 No Schema Checks . . . 198
7.5.3 Insufficient Space on Hard-Drive . . . 198
7.5.4 Forced Termination . . . 198
7.5.5 Terminal Resizing . . . 198

8 Using Novell iMonitor 199

8.1 System Requirements . . . 200
8.1.1 Platforms . . . 200
8.1.2 eDirectory Versions That Can Be Monitored . . . 201
8.2 Accessing iMonitor . . . 201
8.3 iMonitor Architecture . . . 201
8.3.1 Anatomy of an iMonitor Page . . . 201
8.3.2 Modes of Operation . . . 202
8.3.3 iMonitor Features Available on Every Page . . . 204
8.3.4 Configuration Files . . . 204
8.4 iMonitor Features . . . 206
8.4.1 Viewing eDirectory Server Health . . . 207
8.4.2 Viewing Partition Synchronization Status . . . 207
8.4.3 Viewing Server Connection Information . . . 208
8.4.4 Viewing Known Servers . . . 209
8.4.5 Viewing Replica Information . . . 209
8.4.6 Controlling and Configuring the DS Agent . . . 209
8.4.7 Configuring Trace Settings . . . 210
8.4.8 Viewing Process Status Information . . . 211
8.4.9 Viewing Agent Activity . . . 211
8.4.10 Viewing Traffic Patterns . . . 212
8.4.11 Viewing Background Processes . . . 212
8.4.12 Viewing eDirectory Server Errors . . . 213
8.4.13 Viewing DSRepair Information . . . 213
8.4.14 Viewing Agent Health Information . . . 213
8.4.15 Browsing Objects in Your Tree . . . 214
8.4.16 Viewing Entries for Synchronization or Purging . . . 214
8.4.17 Viewing Novell Identity Manager Details . . . 214
8.4.18 Viewing the Synchronization Status of a Replica . . . 215
8.4.19 Configuring and Viewing Reports . . . 215
8.4.20 Viewing Schema, Class, and Attribute Definitions . . . 217
8.4.21 Searching for Objects . . . 218
8.4.22 Using the Stream Viewer . . . 218
8.4.23 Clone DIB Set . . . 219
8.5 Ensuring Secure iMonitor Operations . . . 223

8.6 Configuring HTTP Server Object . . . 224
8.7 Setting HTTP Stack Parameters Using ndsconfig . . . 225

9 SecretStore Configuration for eDirectory Server 227

9.1 UNIX . . . 227
9.2 Windows . . . 227

10 Merging Novell eDirectory Trees 229

10.1 Merging eDirectory Trees . . . 229
10.1.1 Prerequisites . . . 230
10.1.2 Target Tree Requirements . . . 230
10.1.3 Schema Requirements . . . 230
10.1.4 Merging the Source into the Target Tree . . . 231
10.1.5 Partition Changes . . . 231
10.1.6 Preparing the Source and Target Trees . . . 232
10.1.7 Synchronizing Time before the Merge . . . 233
10.1.8 Merging Two Trees . . . 233
10.1.9 Post-Merge Tasks . . . 234
10.2 Grafting a Single Server Tree . . . 235
10.2.1 Understanding Context Name Changes . . . 237
10.2.2 Preparing the Source and Target Trees . . . 238
10.2.3 Containment Requirements for Grafting . . . 239
10.2.4 Grafting the Source and Target Tree . . . 240
10.3 Renaming a Tree . . . 240
10.4 Using the Client to Merge Trees . . . 241
10.4.1 Using the DSMerge eMTool . . . 241
10.4.2 DSMerge eMTool Options . . . 242

11 Encrypting Data In eDirectory 243

11.1 Encrypted Attributes . . . 243
11.1.1 Using Encryption Schemes . . . 245
11.1.2 Managing Encrypted Attributes Policies . . . 245
11.1.3 Accessing the Encrypted Attributes . . . 250
11.1.4 Viewing the Encrypted Attributes . . . 251
11.1.5 Encrypting and Decrypting Backup Data . . . 252
11.1.6 Cloning the DIB Fileset Containing Encrypted Attributes . . . 252
11.1.7 Adding eDirectory 8.8 Servers to Replica Rings . . . 252
11.1.8 Backward Compatibility . . . 252
11.1.9 Migrating to Encrypted Attributes . . . 252
11.1.10 Replicating the Encrypted Attributes . . . 252
11.2 Encrypted Replication . . . 253
11.2.1 Enabling Encrypted Replication . . . 253
11.2.2 Adding a New Replica to a Replica Ring . . . 258
11.2.3 Synchronization and Encrypted Replication . . . 263
11.2.4 Viewing the Encrypted Replication Status . . . 263
11.3 Achieving Complete Security While Encrypting Data . . . 264
11.3.1 Encrypting Data in an All New Setup . . . 264
11.3.2 Encrypting Data in an Existing Setup . . . 265
11.3.3 Conclusion . . . 266

12 Repairing the Novell eDirectory Database 267

12.1 Performing Basic Repair Operations . . . 268
12.1.1 Performing an Unattended Full Repair . . . 269
12.1.2 Performing a Local Database Repair . . . 270

12.1.3 Checking External References . . . 271
12.1.4 Repairing a Single Object . . . 271
12.1.5 Deleting Unknown Leaf Objects . . . 271
12.2 Viewing and Configuring the Repair Log File . . . 272
12.2.1 Opening the Log File . . . 272
12.2.2 Setting Log File Options . . . 272
12.3 Performing a Repair in Novell iMonitor . . . 273
12.4 Repairing Replicas . . . 273
12.4.1 Repairing All Replicas . . . 274
12.4.2 Repairing Selected Replicas . . . 274
12.4.3 Repairing Time Stamps . . . 274
12.4.4 Designating This Server As the New Master Replica . . . 275
12.4.5 Destroying the Selected Replica . . . 276
12.5 Repairing Replica Rings . . . 276
12.5.1 Repairing All Replica Rings . . . 276
12.5.2 Repairing the Selected Replica Ring . . . 277
12.5.3 Sending All Objects to Every Server in the Ring . . . 277
12.5.4 Receiving All Objects from the Master to the Selected Replica . . . 278
12.5.5 Removing This Server from the Replica Ring . . . 278
12.6 Maintaining the Schema . . . 278
12.6.1 Requesting Schema from the Tree . . . 279
12.6.2 Resetting the Local Schema . . . 279
12.6.3 Performing Optional Schema Enhancements . . . 280
12.6.4 Importing Remote Schema . . . 280
12.6.5 Declaring a New Schema Epoch . . . 280
12.7 Repairing Server Network Addresses . . . 281
12.7.1 Repairing All Network Addresses . . . 281
12.7.2 Repairing a Server's Network Addresses . . . 282
12.8 Performing Synchronization Operations . . . 282
12.8.1 Synchronizing the Selected Replica on This Server . . . 283
12.8.2 Reporting the Synchronization Status on This Server . . . 283
12.8.3 Reporting the Synchronization Status on All Servers . . . 283
12.8.4 Performing a Time Synchronization . . . 284
12.8.5 Scheduling an Immediate Synchronization . . . 285
12.9 DSRepair Options . . . 285
12.9.1 Running DSRepair on the eDirectory Server . . . 285
12.9.2 DSRepair Command Line Options . . . 286
12.9.3 Using Advanced DSRepair Switches . . . 288
12.10 Using the Client to Repair a Database . . . 289
12.10.1 Using the DSRepair eMTool . . . 289
12.10.2 DSRepair eMTool Options . . . 290

13 WAN Traffic Manager 293

13.1 Understanding WAN Traffic Manager . . . 293
13.1.1 LAN Area Objects . . . 295
13.1.2 WAN Traffic Policies . . . 296
13.1.3 Limiting WAN Traffic . . . 300
13.1.4 Assigning Cost Factors . . . 301
13.2 WAN Traffic Manager Policy Groups . . . 302
13.2.1 1-3am.wmg . . . 302
13.2.2 7am-6pm.wmg . . . 302
13.2.3 Costlt20.wmg . . . 303
13.2.4 Ipx.wmg . . . 303
13.2.5 Ndsttyps.wmg . . . 303
13.2.6 Onospoof.wmg . . . 314
13.2.7 Opnspoof.wmg . . . 314
13.2.8 Samearea.wmg . . . 315

13.2.9 Tcpip.wmg . . . 315
13.2.10 Timecost.wmg . . . 315
13.3 WAN Policy Structure . . . 316
13.3.1 Declaration Section . . . 316
13.3.2 Selector Section . . . 318
13.3.3 Provider Section . . . 319
13.3.4 Construction Used within Policy Sections . . . 319

14 Understanding LDAP Services for Novell eDirectory 325

14.1 Key Terms for LDAP Services . . . 326
14.1.1 Clients and Servers . . . 326
14.1.2 Objects . . . 326
14.1.3 Referrals . . . 327
14.2 Understanding How LDAP Works with eDirectory . . . 328
14.2.1 Connecting to eDirectory from LDAP . . . 328
14.2.2 Class and Attribute Mappings . . . 331
14.2.3 Enabling Nonstandard Schema Output . . . 334
14.2.4 Syntax Differences . . . 335
14.2.5 Supported Novell LDAP Controls and Extensions . . . 336
14.3 Using LDAP Tools on Linux, Solaris, or AIX . . . 337
14.3.1 LDAP Tools . . . 337
14.4 Extensible Match Search Filter . . . 347
14.5 LDAP Transactions . . . 348
14.5.1 Limitations . . . 350

15 Configuring LDAP Services for Novell eDirectory 351

15.1 Loading and Unloading LDAP Services for eDirectory . . . 351
15.2 Verifying That the LDAP Server Is Loaded . . . 352
15.3 Verifying That the LDAP Server Is Running . . . 353
15.3.1 Scenarios . . . 353
15.3.2 Verifying That The LDAP Server Is Running . . . 354
15.3.3 Verifying That A Device Is Listening . . . 355
15.4 Configuring LDAP Objects . . . 356
15.4.1 Configuring LDAP Server and LDAP Group Objects on Linux, Solaris, AIX Systems . . . 357
15.5 Refreshing the LDAP Server . . . 362
15.6 Authentication and Security . . . 363
15.6.1 Requiring TLS for Simple Binds with Passwords . . . 364
15.6.2 Starting and Stopping TLS . . . 364
15.6.3 Configuring the Server for TLS . . . 365
15.6.4 Configuring the Client for TLS . . . 366
15.6.5 Exporting the Trusted Root . . . 367
15.6.6 Authenticating with a Client Certificate . . . 367
15.6.7 Using Certificate Authorities from Third-Party Providers . . . 367
15.6.8 Creating and Using LDAP Proxy Users . . . 368
15.6.9 Using SASL . . . 369
15.7 Using the LDAP Server to Search the Directory . . . 371
15.7.1 Setting Search Limits . . . 371
15.7.2 Using Referrals . . . 372
15.7.3 Searching Filtered Replicas . . . 379
15.8 Configuring for Superior Referrals . . . 380
15.8.1 Scenario: Superior Referrals in a Federated Tree . . . 380
15.8.2 Creating a Nonauthoritative Area . . . 381
15.8.3 Specifying Reference Data . . . 383
15.8.4 Updating Reference Information through LDAP . . . 384
15.8.5 Affected Operations . . . 384
15.8.6 Discovering Support for Superior References . . . 384

15.9 Persistent Search: Configuring for eDirectory Events . . . 385
15.9.1 Managing Persistent Searches . . . 385
15.9.2 Controlling Use of the Monitor Events Extended Operation . . . 386
15.10 Getting Information about the LDAP Server . . . 387
15.11 Auditing LDAP Events . . . 389

16 Backing Up and Restoring Novell eDirectory 391

16.1 Checklist for Backing Up eDirectory . . . 392
16.2 Understanding Backup and Restore Services . . . 394
16.2.1 About the eDirectory Backup Tool . . . 395
16.2.2 What's Different between Backup and Restore in DSBK and TSA for NDS Backup . . . 395
16.2.3 Overview of How the Backup Tool Does a Restore . . . 397
16.2.4 Format of the Backup File Header . . . 398
16.2.5 Format of the Backup Log File . . . 401
16.2.6 Using DSMASTER Servers as Part of Disaster Recovery Planning . . . 402
16.2.7 Transitive Vectors and the Restore Verification Process . . . 404
16.3 Using Roll-Forward Logs . . . 404
16.3.1 Issues to Be Aware of When Turning On Roll-Forward Logging . . . 405
16.3.2 Location of the Roll-Forward Logs . . . 406
16.3.3 Backing Up and Removing Roll-Forward Logs . . . 407
16.3.4 Cautionary Note: Removing eDirectory Also Removes the Roll-Forward Logs . . . 408
16.4 Preparing for a Restore . . . 408
16.4.1 Prerequisites for Restoring . . . 408
16.4.2 Locating the Right Backup Files for a Restore . . . 409
16.5 Backing Up and Restoring NICI . . . 411
16.5.1 Backing Up NICI . . . 411
16.5.2 Restoring NICI . . . 411
16.6 Using DSBK . . . 412
16.6.1 Prerequisites . . . 413
16.6.2 Using DSBK on Various Platforms . . . 414
16.6.3 Backing Up Manually with DSBK . . . 415
16.6.4 Configuring Roll-Forward Logs with DSBK . . . 416
16.6.5 Restoring from Backup Files with DSBK . . . 417
16.6.6 Backup and Restore Command Line Options . . . 419
16.6.7 Running DSBK as a cron Job . . . 426
16.7 Recovering the Database If Restore Verification Fails . . . 426
16.7.1 Cleaning Up the Replica Ring . . . 427
16.7.2 Repair the Failed Server and Re-add Replicas to the Server . . . 428
16.8 Scenarios for Backup and Restore . . . 429
16.8.1 Scenario: Losing a Hard Drive Containing eDirectory in a Single-Server NetWork . . . 430
16.8.2 Scenario: Losing a Hard Drive Containing eDirectory in a Multiserver Environment . . . 431
16.8.3 Scenario: Losing an Entire Server in a Multiple-Server Environment . . . 433
16.8.4 Scenario: Losing Some Servers in a Multiple-Server Environment . . . 433
16.8.5 Scenario: Losing All Servers in a Multiple-Server Environment . . . 434
16.9 Disaster Recovery Plan using DSBK . . . 435
16.9.1 Disaster Recovery Plan on Linux and UNIX . . . 435
16.9.2 Disaster Recovery Plan on Windows . . . 437

17 SNMP Support for Novell eDirectory 439

17.1 Definitions and Terminology for SNMP . . . 439
17.2 Understanding SNMP Services . . . 440
17.3 eDirectory and SNMP . . . 442
17.3.1 Benefits of SNMP Instrumentation on eDirectory . . . 442
17.3.2 Understanding How SNMP Works with eDirectory . . . 443
17.4 Installing and Configuring SNMP Services for eDirectory . . . 445
17.4.1 Loading and Unloading the SNMP Server Module . . . 445

17.4.2 Subagent Configuration . . . 446
17.4.3 Setting Up SNMP Services for eDirectory . . . 448
17.5 Monitoring eDirectory Using SNMP . . . 453
17.5.1 Traps . . . 454
17.5.2 Configuring Traps . . . 467
17.5.3 Statistics . . . 474
17.6 Troubleshooting . . . 478

18 Maintaining Novell eDirectory 481

18.1 Advanced Referral Costing . . . 481
18.1.1 Improving Server-to-Server Connection . . . 482
18.1.2 Advantages of Referral Costing . . . 484
18.1.3 Deploying ARC . . . 485
18.1.4 Enabling Advanced Referral Costing . . . 486
18.1.5 Tuning Advanced Referral Costing . . . 486
18.1.6 Monitoring Advanced Referral Costing . . . 487
18.2 Improving Bulkload Performance . . . 490
18.2.1 eDirectory Cache Settings . . . 490
18.2.2 LBURP Transaction Size Setting . . . 491
18.2.3 Increasing the Number of Asynchronous Requests in ICE . . . 491
18.2.4 Increased Number of LDAP Writer Threads . . . 492
18.2.5 Disabling ACL Templates . . . 492
18.2.6 Disabling Schema Validation in ICE . . . 492
18.2.7 Disabling ACL Templates . . . 492
18.2.8 Backlinker . . . 494
18.2.9 Enabling/Disabling Inline Cache . . . 494
18.2.10 Increasing the LBURP Time Out Period . . . 494
18.3 Keeping eDirectory Healthy . . . 495
18.3.1 When to Perform Health Checks . . . 495
18.3.2 Health Check Overview . . . 495
18.3.3 Checking eDirectory Health Using iMonitor . . . 496
18.3.4 For More Information . . . 497
18.4 Resources for Monitoring . . . 497
18.5 Upgrading Hardware or Replacing a Server . . . 498
18.5.1 Planned Hardware or Storage Device Upgrade without Replacing the Server . . . 498
18.5.2 Planned Replacement of a Server . . . 500
18.5.3 Server IP Address Changes . . . 503
18.6 Restoring eDirectory after a Hardware Failure . . . 504

19 DHost iConsole Manager 505

19.1 What is DHost? . . . 506
19.2 Running DHost iConsole . . . 506
19.2.1 Running DHost iConsole on Windows . . . 507
19.2.2 Running DHost iConsole on Linux, Solaris, and AIX . . . 507
19.3 Managing eDirectory Modules . . . 507
19.3.1 Loading or Unloading Modules on Windows . . . 508
19.3.2 Loading or Unloading Modules on Linux, Solaris, and AIX . . . 508
19.4 Querying for DHost Information . . . 509
19.4.1 Viewing the Configuration Parameters . . . 509
19.4.2 Viewing Protocol Information . . . 509
19.4.3 Viewing Connection Properties . . . 510
19.4.4 Viewing the Thread Pools Statistics . . . 510
19.5 Process Stack . . . 511

20 Setting the sadmin Password 513

21 The eDirectory Management Toolbox 515

21.1 Using the Command Line Client . . . 516
21.1.1 Displaying the Command Line Help . . . 516
21.1.2 Running the Command Line Client in Interactive Mode . . . 516
21.1.3 Running the Command Line Client in Batch Mode . . . 520
21.1.4 eMBox Command Line Client Options . . . 522
21.1.5 Establishing a Secure Connection with the Client . . . 522
21.1.6 Finding Out eDirectory Port Numbers . . . 523
21.2 Using the Logger . . . 523
21.2.1 Using the Logger Command Line Client . . . 524
21.2.2 Using the Logger Feature in Novell iManager . . . 525
21.3 Using the eMBox Client for Backup and Restore . . . 525
21.3.1 Prerequisites . . . 526
21.3.2 Backing Up Manually with the eMBox Client . . . 527
21.3.3 Doing Unattended Backups, Using a Batch File with the eMBox Client . . . 528
21.3.4 Configuring Roll-Forward Logs with the eMBox Client . . . 529
21.3.5 Restoring from Backup Files with the eMBox Client . . . 531
21.4 Using Novell iManager for Backup and Restore . . . 533
21.4.1 Backing Up Manually with iManager . . . 533
21.4.2 Configuring Roll-Forward Logs with iManager . . . 536
21.4.3 Restoring from Backup Files with iManager . . . 537

A NMAS Considerations 541

A.1 Setting Up a Security Container As a Separate Partition . . . 541
A.2 Merging Trees with Multiple Security Containers . . . 541
A.2.1 Product-Specific Operations to Perform prior to Tree Merge . . . 542
A.2.2 Performing the Tree Merge . . . 545
A.2.3 Product-Specific Operations to Perform after the Tree Merge . . . 545

B Novell eDirectory Linux and UNIX Commands and Usage 547

B.1 General Utilities . . . 547
B.2 LDAP-Specific Commands . . . 552

C Configuring OpenSLP for eDirectory 555

C.1 Service Location Protocol . . . 555
C.2 SLP Fundamentals . . . 555
C.2.1 Novell Service Location Providers . . . 556
C.2.2 User Agents . . . 557
C.2.3 Service Agents . . . 557
C.3 Configuration Parameters . . . 558

D How Novell eDirectory Works with DNS 561

E Configuring GSSAPI with eDirectory 563

E.1 Prerequisites . . . 563
E.1.1 Assumptions on Network Characteristics . . . 564
E.1.2 Installing the Kerberos Plug-in for iManager . . . 564
E.1.3 Adding Kerberos LDAP Extensions . . . 565
E.1.4 Exporting the Trusted Root Certificate . . . 567
E.2 Configuring the SASL-GSSAPI Method . . . 567

E.2.1 Merging eDirectory Trees Configured with SASL-GSSAPI Method. 567
E.3 Managing the SASL-GSSAPI Method 568
E.3.1 Extending the Kerberos Schema 568
E.3.2 Managing the Kerberos Realm Object. 568
E.3.3 Managing a Service Principal 570
E.3.4 Editing Foreign Principals 574
E.3.5 Configuring SASL GSSAPI Authentication if MIT Kerberos KDC Uses eDirectory as Back End 574
E.4 Creating a Login Sequence 575
E.5 How Does LDAP Use SASL-GSSAPI? 575
E.6 Error Messages 575

F Security Considerations 577

F.1 LDAP Binds 577
F.2 Nessus Scan Results 577

About This Guide

This guide describes how to manage and configure Novell eDirectory 8.8.

- Chapter 1, “Understanding Novell eDirectory,” on page 17
- Chapter 2, “Designing Your Novell eDirectory Network,” on page 73
- Chapter 3, “Managing Objects,” on page 93
- Chapter 4, “Managing the Schema,” on page 123
- Chapter 5, “Managing Partitions and Replicas,” on page 135
- Chapter 6, “Novell eDirectory Management Utilities,” on page 147
- Chapter 7, “Offline Bulkload Utility,” on page 193
- Chapter 8, “Using Novell iMonitor,” on page 199
- Chapter 9, “SecretStore Configuration for eDirectory Server,” on page 227
- Chapter 10, “Merging Novell eDirectory Trees,” on page 229
- Chapter 11, “Encrypting Data In eDirectory,” on page 243
- Chapter 12, “Repairing the Novell eDirectory Database,” on page 267
- Chapter 13, “WAN Traffic Manager,” on page 293
- Chapter 14, “Understanding LDAP Services for Novell eDirectory,” on page 325
- Chapter 15, “Configuring LDAP Services for Novell eDirectory,” on page 351
- Chapter 16, “Backing Up and Restoring Novell eDirectory,” on page 391
- Chapter 17, “SNMP Support for Novell eDirectory,” on page 439
- Chapter 18, “Maintaining Novell eDirectory,” on page 481
- Chapter 19, “DHost iConsole Manager,” on page 505
- Chapter 20, “Setting the sadmin Password,” on page 513
- Chapter 21, “The eDirectory Management Toolbox,” on page 515
- Appendix A, “NMAS Considerations,” on page 541
- Appendix B, “Novell eDirectory Linux and UNIX Commands and Usage,” on page 547
- Appendix C, “Configuring OpenSLP for eDirectory,” on page 555
- Appendix D, “How Novell eDirectory Works with DNS,” on page 561
- Appendix E, “Configuring GSSAPI with eDirectory,” on page 563
- Appendix F, “Security Considerations,” on page 577

Audience

The guide is intended for network administrators.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation and enter your comments there.

Documentation Updates

For the most recent version of the *Novell eDirectory 8.8 SP7 Administration Guide*, see the Novell eDirectory online documentation (http://www.novell.com/documentation/edir88/) Web site.

Additional Documentation

For eDirectory installation instructions, see the *Novell eDirectory 8.8 SP7 Installation Guide* (http://www.novell.com/documentation/edir88/edirin88/data/a2iii88.html).

For documentation on iManager, the eDirectory management utility, see the *Novell iManager 2.7 Administration Guide* (http://www.novell.com/documentation/imanager27/imanager_admin_275/data/hk42s9ot.html).

1 Understanding 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. 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 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://support.novell.com/techcenter/articles/anp20020801.html).

Figure 1-1 shows a few of the objects as viewed in the Novell iManager management utility.

Figure 1-1 *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 Section 1.2, "Object Classes and Properties," on page 21.

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

This chapter includes the following information:

- Section 1.1, "Ease of Management through Novell iManager," on page 18
- Section 1.2, "Object Classes and Properties," on page 21

- Section 1.3, “Context and Naming,” on page 43
- Section 1.4, “Schema,” on page 46
- Section 1.5, “Partitions,” on page 52
- Section 1.6, “Replicas,” on page 55
- Section 1.7, “Server Synchronization in the Replica Ring,” on page 60
- Section 1.8, “Access to Resources,” on page 60
- Section 1.9, “eDirectory Rights,” on page 61

1.1 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.

For more information, see the *Novell iManager 2.7 Administration Guide* (http://www.novell.com/documentation/imanager27/imanager_admin_275/data/hk42s9ot.html).

1.1.1 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 Linux, 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.

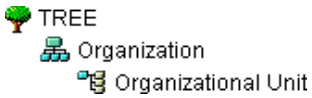
The hierarchical structure of the tree gives you great management flexibility and power. These benefits primarily result from the following two features:

- “Container Objects” on page 18
- “Inheritance” on page 19

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 1-2:

Figure 1-2 *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, which contains the Database application, the Bookkeepers group, the LaserPrinter printer, and the users Amy, Bill, and Bob.

To do this, navigate to the View Objects tab in iManager and select the parent tree of the *Accounting* object in the left pane. In the right pane, select *Accounting* and then click *Actions* > *Modify Trustees*. Click *Add Trustee* and add Amy as a trustee. Next, click *Assigned Rights* and select the rights you want Amy to have. Now Amy has rights to manage the Database application, the Bookkeepers group, the LaserPrinter printer, and the users Bill and Bob, in addition to herself.

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 1-3 on page 19.

Figure 1-3 *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 Section 1.9, “eDirectory Rights,” on page 61.

1.1.2 Web-Based Management Utility

iManager is a browser-based tool used for administering, managing, and configuring eDirectory objects. 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.

To run iManager, you will need a workstation with Microsoft Internet Explorer 6.0 SP1 or later (recommended), Mozilla 1.7 or later, or Mozilla Firefox 0.9.2 or later.

IMPORTANT: While you might be able to access iManager through a Web browser not listed, we do not guarantee full functionality.

You can use iManager 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 bundled with iManager 2.7:

- 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
- Universal Password Enforcement
- Priority Sync
- Encrypted Attributes
- Encrypted Replication
- Novell Licensing Services (NLS)

- Novell Modular Authentication Service (NMAS)
- PKI/Certificate
- Filtered Replica Configuration Wizard
- SNMP
- WAN Traffic Manager

For more information on installing, configuring, and running iManager, *Novell iManager 2.7 Administration Guide* (http://www.novell.com/documentation/imanager27/imanager_admin_275/data/hk42s9ot.html).

1.1.3 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.

1.2 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 First Name, 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 Section 1.4, "Schema," on page 46.

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

1.2.1 List of Objects

The following tables list eDirectory object classes. Added services can create new object classes in eDirectory that are not listed below.

eDirectory Container Object Classes

iManager Icon	Container Object (Abbreviation)	Description
	Tree	Represents the beginning of your tree. For more information, see “Tree” on page 23.
	Country (C)	Designates the countries where your network resides and organizes other directory objects within the country. For more information, see “Country” on page 26.
	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 24.
	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 25.
	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 26.

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 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 40.
	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 42.

iManager Icon	Leaf Object	Description
	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. The rights then transfer to each user in the group. For more information, see “Group” on page 30.
	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 42.
	Server	Represents a server running any operating system. For more information, see “Server” on page 27.
	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 28.
	Volume	Represents a physical volume on the network. For more information, see “Volume” on page 28.

1.2.2 Container Object Classes

- “Tree” on page 23
- “Organization” on page 24
- “Organizational Unit” on page 25
- “Country” on page 26
- “Domain” on page 26

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 Section 1.9, “eDirectory Rights,” on page 61. 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.
- Tree name cannot exceed 32 characters.

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.

- Organization name can be 64 characters long.

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.
- Organizational Unit name can be 64 characters long.

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.
- Country name cannot exceed 2 characters.

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:

```
DS=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

Domain name can be 64 characters long.

1.2.3 Leaf Object Classes

- "Server" on page 27
- "Volume" on page 28
- "User" on page 28
- "Group" on page 30
- "Nested Groups" on page 33
- "Alias" on page 40
- "Directory Map" on page 42
- "Profile" on page 42

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.

What a Server Object Represents

The Server object represents a server running eDirectory or a bindery-based 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`).

Linux and UNIX file system partitions cannot be managed using Volume objects. Volume objects are supported only on OES Linux.

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.

Important Properties

In addition to the required Name and Host Server 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 eDirectory version of the server hosting the 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 2.7 Administration Guide* (http://www.novell.com/documentation/imanager27/imanager_admin_275/data/hk42s9ot.html).

- Batches from database files

  For more information on using batch files, see Section 2.2, "Designing the eDirectory Tree," on page 74.

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.
- 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 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.ietf.org/rfc/rfc2255.txt).

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 distinguished names (DNs). 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 DNs 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 group management tasks by clicking the *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*.

- dgAllowDuplicates

  Specifies whether or not duplicates are allowed while printing dynamic group members. The default is TRUE.

- 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 dgldentity 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 dgldentity 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 was overcome in eDirectory 8.7 and later. Now eDirectory 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.

- 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 Dynamic Groups on Pre-eDirectory 8.6.1 Databases

Dynamic group 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

From eDirectory 8.7.3 onwards, 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.

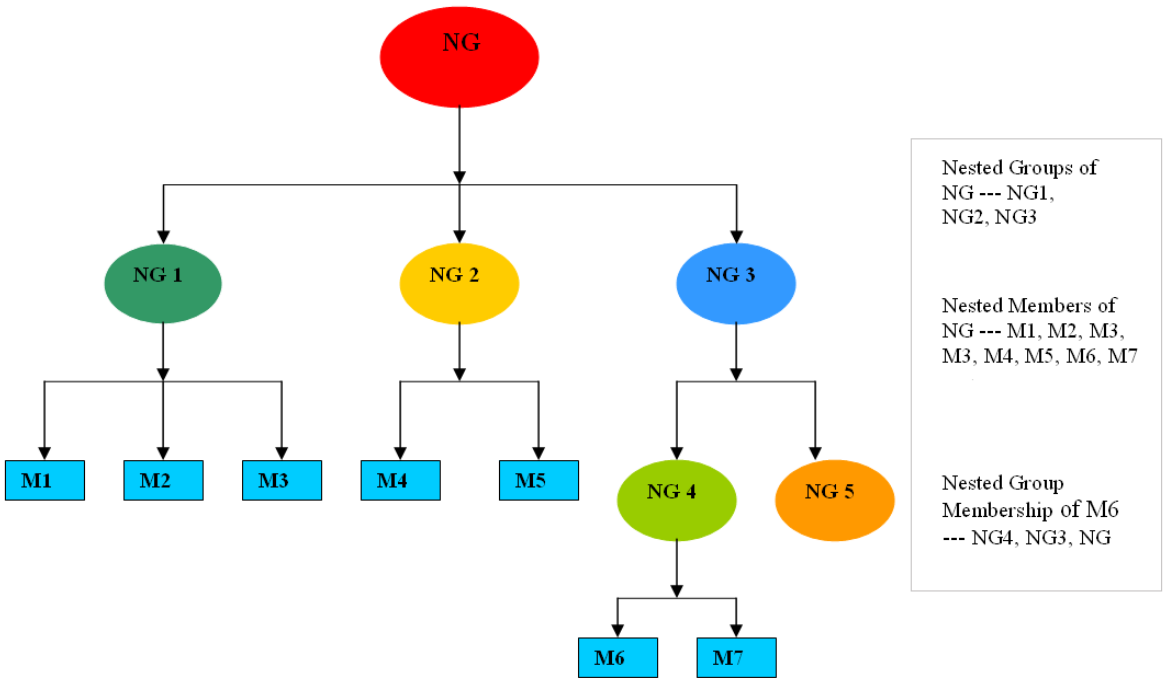
For more information, see "How to Manage and Use Dynamic Groups in Novell eDirectory" (http://support.novell.com/techcenter/articles/ana20020405.html).

Nested Groups

Nested groups allow grouping of groups and provide a more structured form of grouping. An attribute called groupMember is introduced to specify the nested groups whose members become nested members of the containing nested group object. Group objects are specified statically in the groupMember attribute. The group containing other groups is referred to as the containing group, and the groups that are part of this group are referred to as contained groups. Currently, nesting is allowed only for static groups (not dynamic groups). Nesting can have multiple levels up to 200.

IMPORTANT: Nesting is supported within the local server only. If a contained group is not found on the local server, its members are not listed as the nested members of the containing group.

Figure 1-4 *Nested Groups*



You can use iManager or LDAP tools to create the nested groups.

- "Creating Nested Groups by Using LDAP Tools" on page 34
- "Creating Nested Groups by Using iManager" on page 34

Creating Nested Groups by Using LDAP Tools

You can use LDAP tools to create the nested groups. A new auxiliary class, nestedGroupAux, along with the structural class Group represents a nested group. This auxiliary class can be added to the existing static group object to convert it into a nested group.

Both the contained and the containing group should be nested group objects. Only when the contained group is a nested group, it can populate the groupMembership attribute (groupMembership attribute not a part of static group) on it to specify the containing group. If the contained groups are found to be static group objects or dynamic group objects, only the static members of the static or dynamic group objects are listed as nested members.

You can use LDIF files and LDAP tools to manage such groups. The most useful properties associated with nested groups are groupMember and nestedConfig.

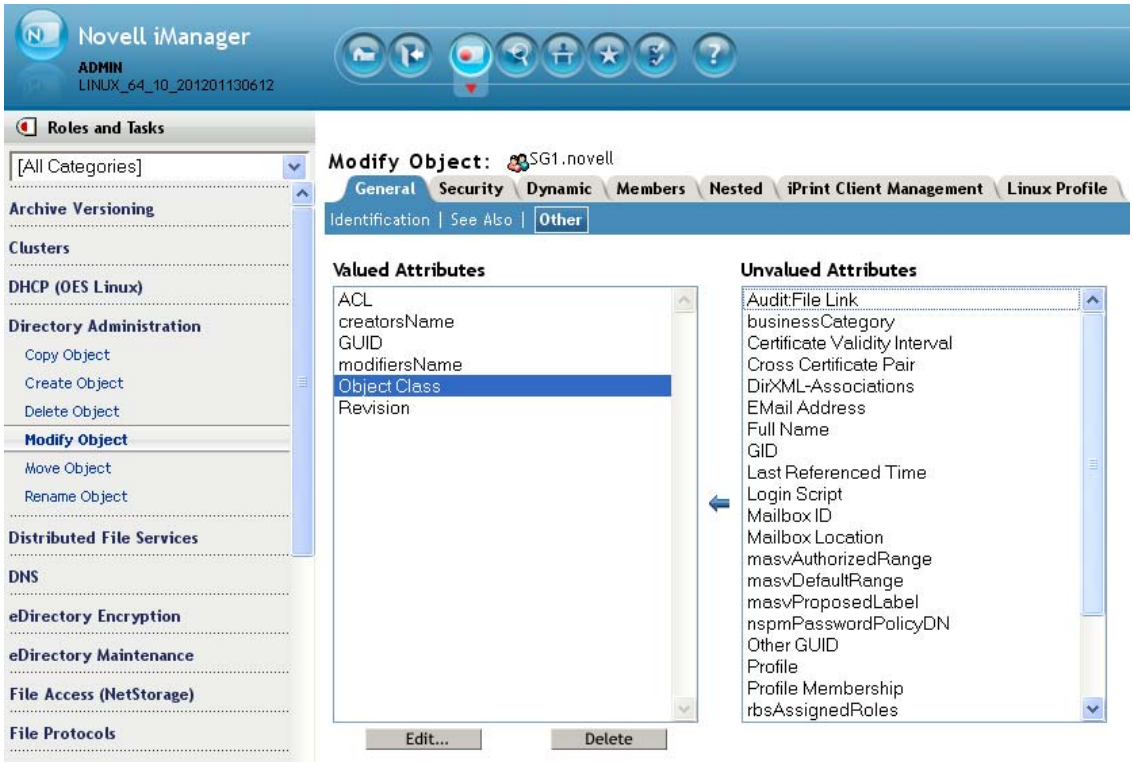
Creating Nested Groups by Using iManager

You can use iManager 2.7 SP1 or later plug-ins to create a nested group or to change a static group to a nested group in order to associate it with another group.

1 Log in to iManager 2.7 SP1 or later with administrator credentials and select *Groups* > *Create Group* from the left panel to create a static group. For example, SG1.

2 Select *Directory Administration* > *Modify Object* from the left panel, then browse to and select the *SG1.novell* object.

3 Click the *Other* tab, then select *Object Class* from the *Unvalued Arributes* list.

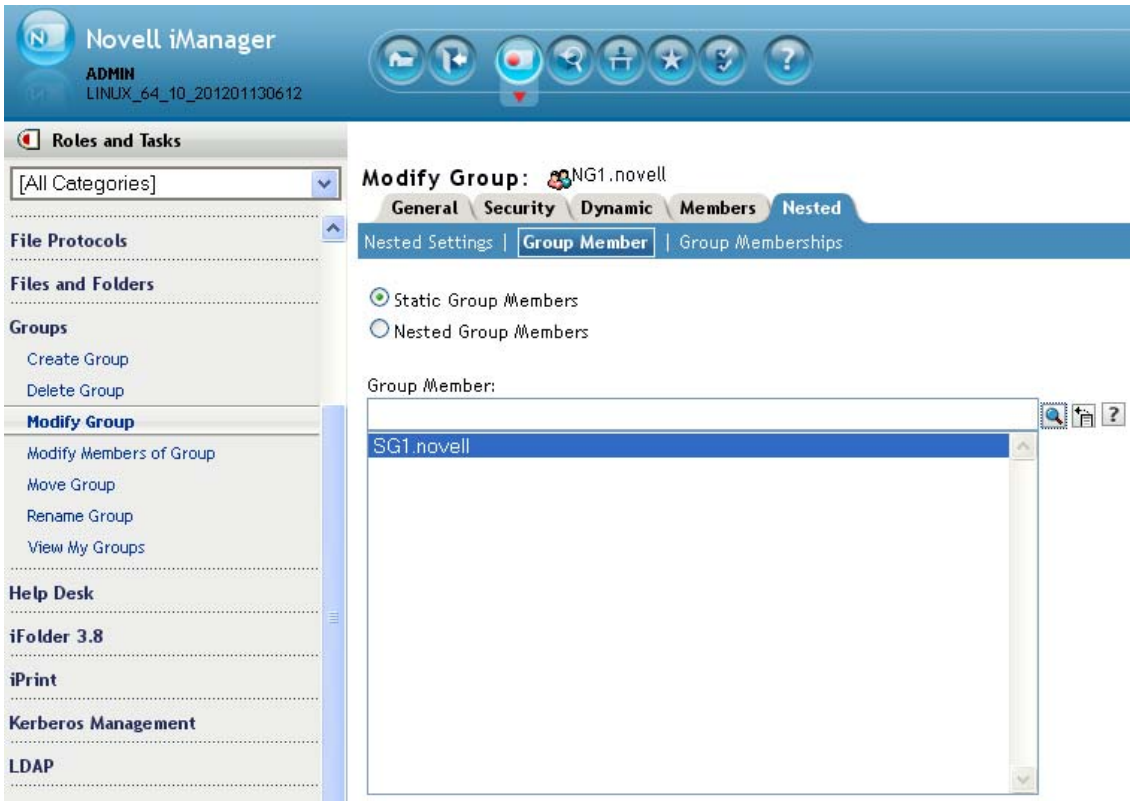


4 Add the *nestedGroupAux* value to the *Object Class*, then click *OK* and *Apply*.

5 Select *Groups* > *Create Group* from the left panel, select the *Nested Group* checkbox to create a nested group with the name NG1, then click *OK*.

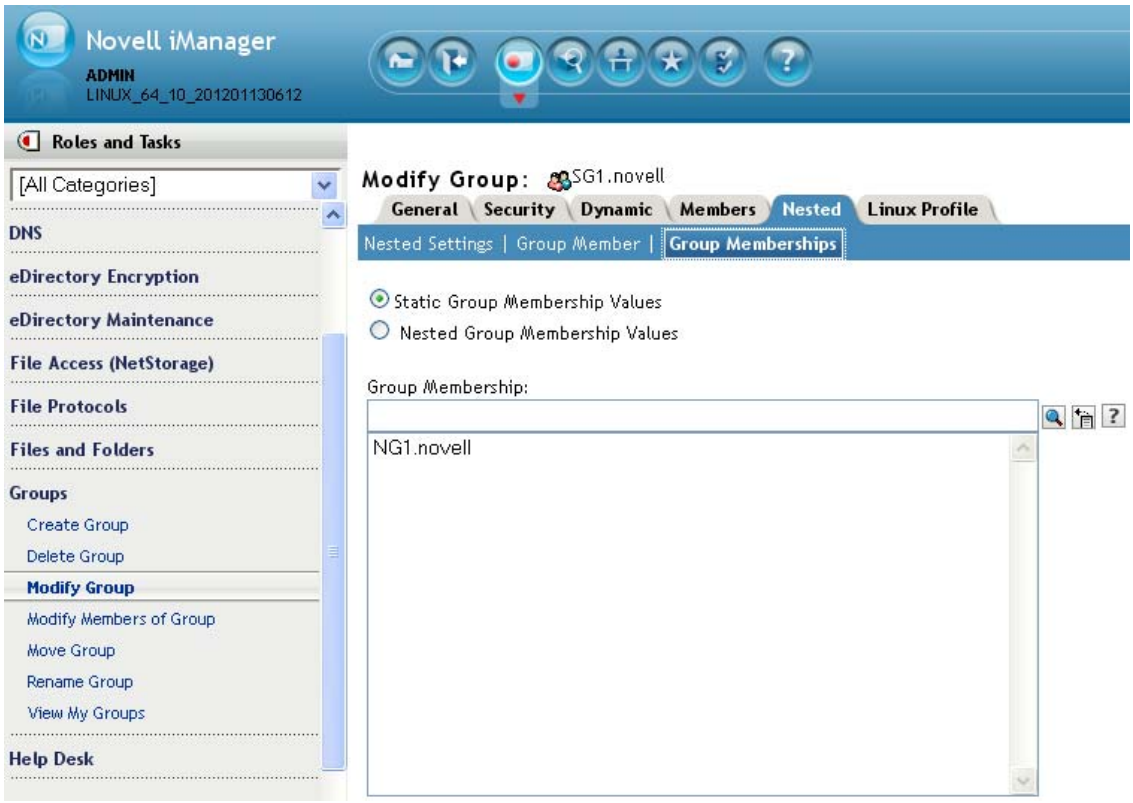
6 Select *Groups* > *Modify Group* from the left panel, select the *Nested Group* checkbox to modify the nested group with the name NG1, then click *OK*.

7 To modify the nested group, select *Groups* > *Modify Group*, then browse to and select *NG1.novell*.



8 To associate a static group to *NG1*, select the *Nested* tab > *Group Member* tab, then browse to and select the SG1 static group.

9 Click *Apply*, then click *OK* to convert the SG1 static group to the NG1 nested group.



The static group that you converted to a nested group is now a member of the static group.

To verify the membership details of SG1, select *Groups* > *Modify Group* from the left panel, then select *SG1.novell*. Select the *Nested* > *Group Membership* tab to verify the static group's membership information as *NG1.novell*.

Nested Group Properties

- groupMember

  By default, the members of a nested group include all the nested members. Therefore, the member attribute listing always returns all the nested members, and the assertion on the member attribute returns all the nested group objects. If the configuration is set to 1 (no nesting), it refers only to the direct members.

- Group Membership

  groupMembership specifies the group that this object (generally a user object) belongs to. This attribute is associated with the nestedGroupAux class, and it holds the DN of the nested group of which this group is a group member. When associated with a group object, it indicates the nested group of which this group is a member (specifically a groupMember). Similar to member and groupMember, groupMembership lists all the nested groups of which this group has a groupMembership via a nested relationship. The nestedConfig also applies to the groupMembership attribute. For non-group member objects, the nestedConfig of individual groups is used.

- nestedConfig

nestedConfig sets the configuration of the nested group object. The configuration values currently supported are 0 (nesting local server) and 1 (no nesting). By default, it always nests the local server. If only direct values such as member, groupMember, or groupMembership are to be listed for the attribute, the configuration can be set to 1.

- excludedMember

  excludedMember is included as part of the nestedGroupAux class, but it is currently not used. In future, it will indicate members that are to be excluded from nested members, analogous to dynamic groups.

Nested Group Operations

1. One group can be a member of another group via the groupMember attribute. Both groups, contained as well containing, must have the nested group auxiliary class associated with the group object.

   ```
   dn: cn=finance,o=nov

   objectclass: group

   objectclass: nestedGroupAux

   groupMember: cn=accounts,o=nov

   member: cn=jim,o=nov


   dn: cn=accounts,o=nov

   objectclass: group

   objectclass: nestedGroupAux

   member: cn=allen,o=nov

   member: cn=ESui,o=nov

   member: cn=YLi,o=nov
   ```

2. Reading the member attribute of a nested group also causes the members of the contained group to be returned if both the contained and the containing group are present locally on the server:

   ```
   dn: cn=finance,o=nov
   member: cn=jim,o=nov
   member: cn=allen,o=nov
   member: cn=ESui,o=nov
   member: cn=YLi,o=nov
   ```

   The same holds true for the groupMember attribute.

3. The reciprocal attribute to the member attribute is groupMembership. This implies that the `cn=allen,o=nov` user object needs to possess the groupMembership attribute populated with the `cn=accounts,o=nov` group DN. The groupMembership of the `cn=accounts,o=nov` group needs to be populated with `cn=finance,o=nov`. On reading the groupMembership attribute of the `cn=allen,o=nov` user object, both the groups are returned.

   ```
   dn: cn=allen,o=nov
   groupMembership: cn=accounts,o=nov
   groupMembership: cn=finance,o=nov
   ```

4. The ACLs can be assigned to a nested group and all the objects that are members of the nested group will acquire the rights. In the assigned rights field, an additional nested ACL bit (0x80000000) needs to be set in addition to the rights being assigned.

```
dn: cn=finance,o=nov
groupMember: cn=accounts,o=nov


dn: cn=accounts,o=nov
member: cn=allen,o=nov


dn: ou=MyCo,o=nov
objectclass: Organizational Unit
ACL: 2147483650#entry#cn=finance,o=nov#[All Attributes Rights]
```

The rights value – 2147483650 (0x80000002) has nested ACL (0x80000000) and read rights bit (0x00000002) set. So, the user object `cn=allen,o=nov` has been granted read rights on all attributes of the MyCo object via the nested group `cn=finance,o=nov`.

5. Applications can use filter assertions on the member, groupMember, and groupMembership attributes. In the above example, an assertion of `member=cn=allen,o=nov` would return the following:

   ```
   dn: cn=accounts,o=nov
   dn: cn=finance,o=nov
   ```

   An assertion of `groupMembership=cn=finance,o=nov` would return the following objects:

   ```
   dn: cn=allen,o=nov
   dn: cn=jim,o=nov
   dn: cn=ESui,o=nov
   dn: cn=YLi,o=nov
   dn: cn=accounts,o=nov
   ```

NOTE: There is no limit on the levels of nesting in any of the above cases. Loop detection in nested groups is done while any of the above mentioned attributes are read.

Limitations

- Nested relationships do not span beyond the local server. The objects, users, and groups involved need to be locally present on the server.
- No duplicate elimination is done in membership listing.
- Nesting of dynamic groups is not supported.
- Nested ACLs as well as the nesting semantics are not supported on older eDirectory servers (version 8.8 SP1 and earlier).

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

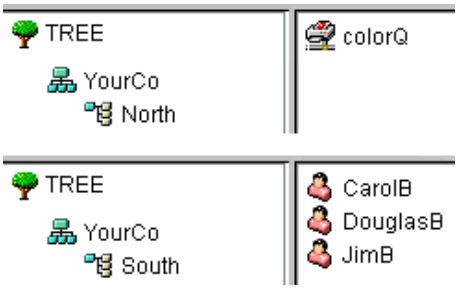
An Alias object represents another object, which can be a container, User object, or any other object in the tree. An Alias object does not carry trustee rights of its own. Any trustee authority you grant to the Alias object applies to the object it represents. The Alias can be a target of a trustee 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 1-5, but need access to the Print Queue object named ColorQ in the North container.

Figure 1-5 *Sample Containers*



You can create an Alias object in the South container, as shown in Figure 1-6.

Figure 1-6 *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 volumes, you cannot create Directory Map objects.

What a Directory Map Object Represents

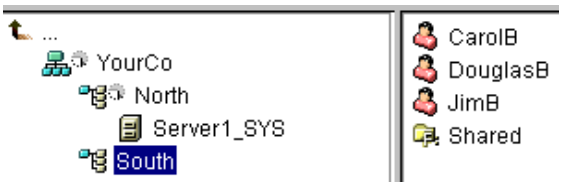
A Directory Map object represents a directory on a volume. An Alias object, on the other hand, represents an object.

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 1-7.

Figure 1-7 *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

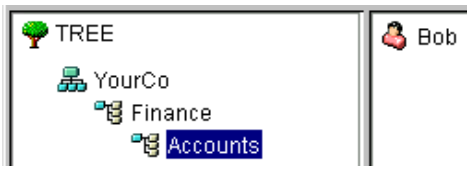
  If you have INCLUDE statements in the login script, you need to give the Profile object rights to the files included with the Rights to Files and Directories property.

1.3 Context and Naming

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

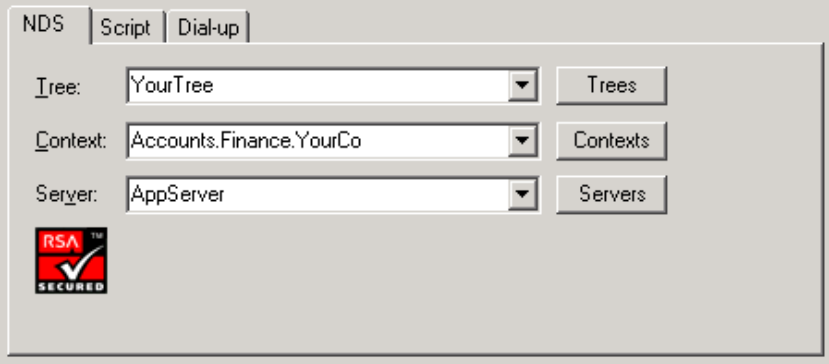
You can see in Figure 1-8 that User Bob is in Organizational Unit Accounts, which is in Organizational Unit Finance, which is in Organization YourCo.

Figure 1-8 *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 1-9.

Figure 1-9 *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.

1.3.1 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.

1.3.2 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.

1.3.3 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.

1.3.4 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.

1.3.5 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
```

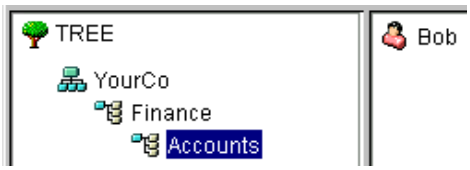
eDirectory interprets the command as "Change the context to the Finance container, which is in the YourCo container, resolved from the top of the tree."

1.3.6 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 1-10.

Figure 1-10 *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."

1.3.7 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 1-11.

Figure 1-11 *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..
```

1.3.8 Context and Naming on Linux and UNIX

When Linux and UNIX user accounts are migrated to eDirectory, the eDirectory context is not used to name users.

1.4 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, 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 add the attribute to an auxiliary class. You can then use that auxiliary class to extend User objects as needed. For more information about creating auxiliary classes, see Section 4.1.6, "Creating an Auxiliary Class," on page 126.

For more information about working with the eDirectory schema, see Chapter 4, "Managing the Schema," on page 123.

1.4.1 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.

1.4.2 Schema Classes, Attributes, and Syntaxes

- "Classes" on page 47
- "Attributes" on page 48
- "Syntaxes" on page 48

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 (*surname* or *employee number*) and given a syntax type (*string* or *number*). From then on, it is available in the attribute lists in Schema Manager.

NOTE: Due to a replication issue, attributes in eDirectory other than the stream attribute type cannot contain values larger than 60 KB or 30,000 characters. If a user or application sets the value of a string or binary attribute and exceeds that limit, eDirectory returns a -649 error indicating that the value is too long.

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:

- Back Link

  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.

- EMail 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.

- Integer 64

  Used by attributes represented as 64-bit integer values. Integer 64 attributes support the Microsoft Large Integer Syntax and can be used to store large-integer values or dates previous to 1970 or beyond 2038.

  NOTE: eDirectory uses its existing syntax and 32-bit values for internal timestamps.

- 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. 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.

1.4.3 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.

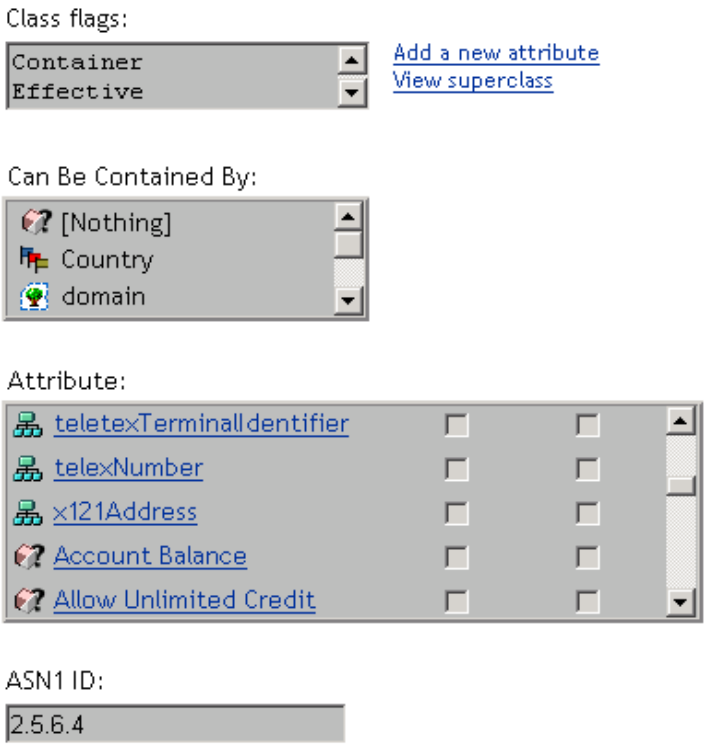
An exception to the rule is when an optional attribute is used for naming, the attribute then becomes mandatory.

1.4.4 Sample Schema

Figure 1-12 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 1-12 *Class Information Page in iManager*



1.4.5 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 Section 4.1, “Extending the Schema,” on page 124 and Section 4.2, “Viewing the Schema,” on page 127 for more information.

1.5 Partitions

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.

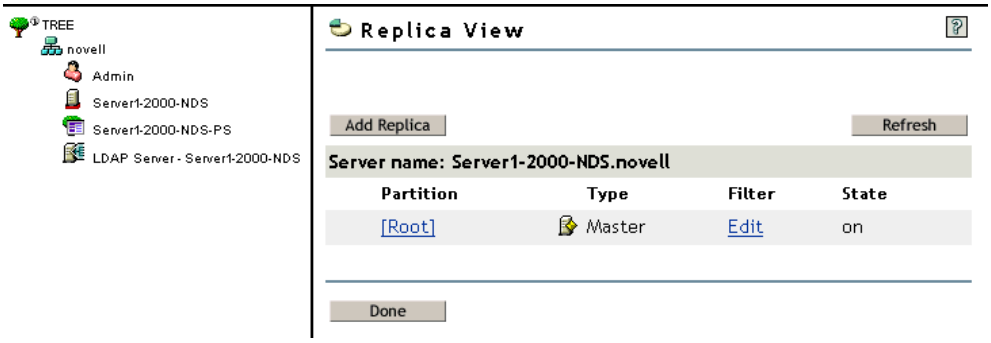
Partitioning allows you to take part of the directory off one server and put it on another server.

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 135.

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 1-13 *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.

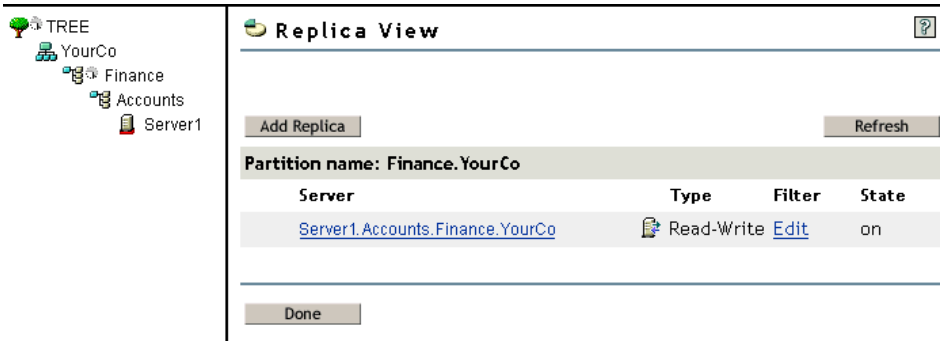
This is the default partitioning for eDirectory, keeping the entire directory together in one partition.

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 Section 1.6, "Replicas," on page 55 and "Viewing Replicas on an eDirectory Server" on page 143.

1.5.1 Partitions

Partitions are named by their topmost container. In Figure 1-14 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 1-14 *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 145.

1.5.2 Distributing Replicas for Performance

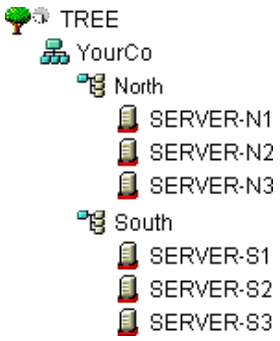
In the preceding example, suppose that Server1 holds replicas of both the Tree partition and the Finance partition. At this point, you haven't gained any performance advantage from eDirectory because Server1 still holds the entire directory (replicas of both partitions).

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.

1.5.3 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 1-15 *Sample eDirectory Containers*



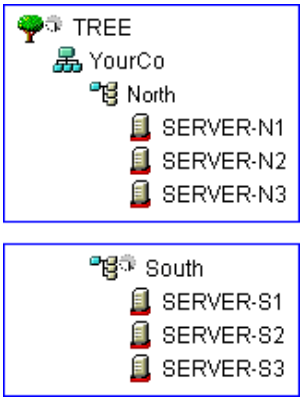
eDirectory performs faster and more reliably in this scenario if the directory is divided in two partitions.

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.

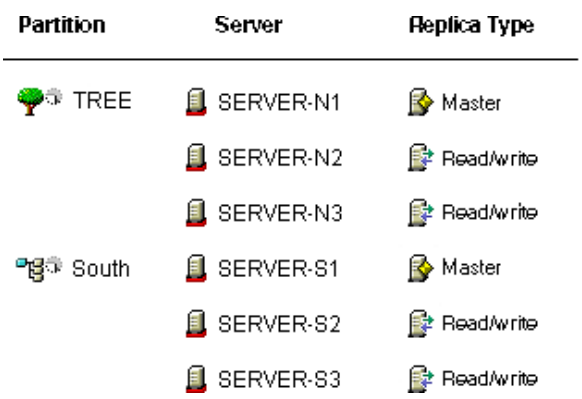
The two-partition solution shown in Figure 1-16 solves performance and reliability problems over the WAN link.

Figure 1-16 *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 1-17.

Figure 1-17 *Sample Partitions, Severs, and Replicas*



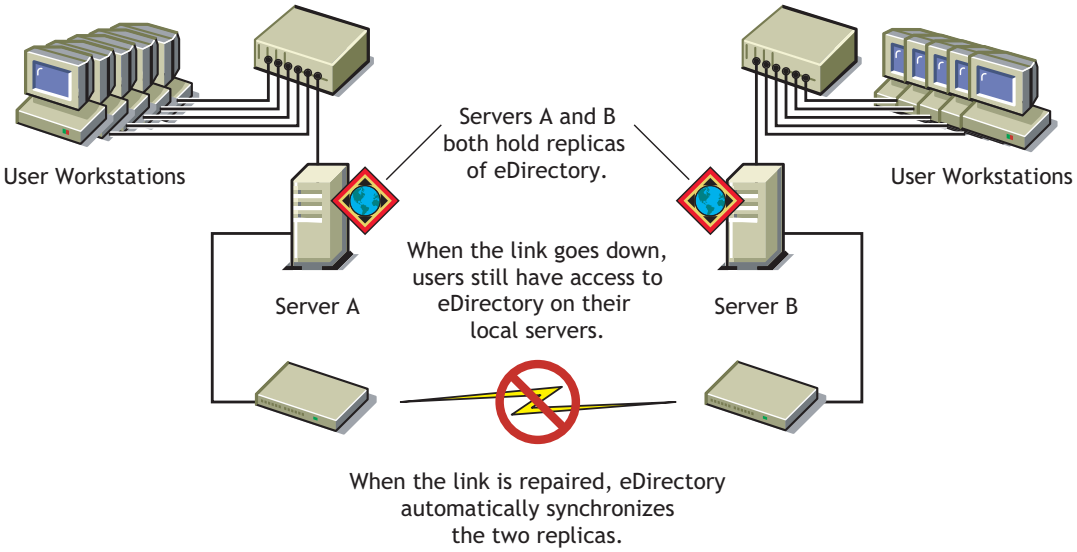
For each site, the objects that represent local resources are kept locally. Synchronization traffic among servers also happens locally over the LAN, rather than over the slow, unreliable WAN link.

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

1.6 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 1-18).

Figure 1-18 *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 135.

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 402.

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 servers that provide bindery services.

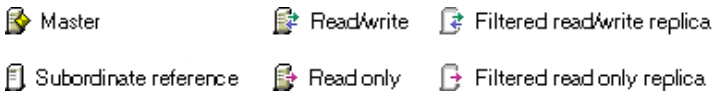
If users regularly access eDirectory information across a WAN link, you can decrease access time and WAN traffic by placing a replica containing the needed information on a server that users can access locally.

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.

1.6.1 Replica Types

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

Figure 1-19 *Replica Types*



- "Master Replica" on page 57
- "Read/Write Replica" on page 58
- "Read-Only Replica" on page 58
- "Filtered Read/Write Replica" on page 58
- "Filtered Read-Only Replica" on page 59
- "Subordinate Reference Replica" on page 59

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, like 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. If login update is enabled then login to read only replica fails as it involves attribute updates.

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 66.

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 59.

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 59.

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.

1.6.2 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 of 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 Novell Identity Manager.

  For more information on Novell Identity Manager, see the *Novell Identity Manager 4.0.1 Administration Guide* (http://www.novell.com/documentation/idm401/)
- 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 Section 5.6, "Setting Up and Managing Filtered Replicas," on page 142.

Allowing Local Logins to Filtered Replicas

In addition to selecting the *Enable local login* option in iManager, to allow local logins to a Filtered Replica, you should also add the class ndsLoginProperties to the filter.

1.7 Server Synchronization in the Replica Ring

When multiple servers hold replicas of the same partition, those servers are considered a replica ring. Synchronization is the propagation of directory information from one replica to another, so the information in each partition is consistent with the other. eDirectory automatically keeps those servers synchronized. For more information, refer Section 3.4, "Synchronization," on page 110

The following are the types of eDirectory synchronization:

- Normal Synchronization or Replica Synchronization
- Priority Sync

1.8 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/imanager27/imanager_admin_275/data/am757mw.html) in the *Novell iManager 2.7 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 Section 3.3, "Configuring Role-Based Services," on page 105 for more information.

1.9 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 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, printing upgrade, or Windows user migration receive trustee assignments appropriate for most situations.

1.9.1 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 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.

[This] is a special type of trustee, that is defined to be an authenticated object, when its name matches the entry being accessed. This helps the administrator to easily specify rights such as, every user manages his own telephone number, with a single ACL at the top of the tree with [This] as a trustee.

1.9.2 eDirectory Rights Concepts

The following concepts can help you better understand eDirectory rights.

- "Object (Entry) Rights" on page 62

- "Property Rights" on page 62
- "Effective Rights" on page 63
- "How Effective Rights Are Calculated" on page 63
- "Security Equivalence" on page 65
- "Access Control List (ACL)" on page 66
- "Inherited Rights Filter (IRF)" on page 66

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.

IMPORTANT: If you grant a trustee Read access to the [All Attributes Rights] property of a user, the trustee is granted Read access to the `Password Management` attribute for that user. The trustee can then read the user's passwords.

For more information about creating and managing password policies, see "Creating Password Policies" (https://www.netiq.com/documentation/password_management33/pwm_administration/data/an4bun5.html) in the *Novell Password Management Administration Guide* (https://www.netiq.com/documentation/password_management33/pwm_administration/data/bookinfo.html).

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:
   - **2a** 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.
   - **2b** eDirectory moves down a level in the branch of the tree that contains the target resource.
   - **2c** 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.
   - **2d** 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.
   - **2e** eDirectory repeats the filtering and adding steps (Step 2c and Step 2d above) at each level of the tree, including at the target resource.

2f 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:

3a 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.

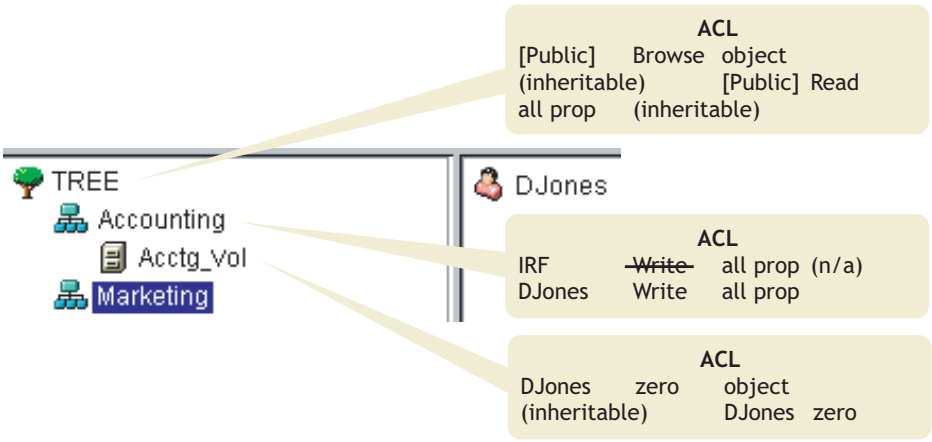
3b eDirectory adds rights that are implied by any of the current effective rights.

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 1-20.

Figure 1-20 *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 71.

1.9.3 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 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 the `sys:\public` folder. This allows User objects under the container to access utilities in `\public`. **NOTE**: These rights only apply to servers running OES Linux.
User objects	If home directories are automatically created for users, the users have the Supervisor right to those directories.

1.9.4 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 69.

To delegate the use of specific functions in role-based administration applications, see Section 3.3, "Configuring Role-Based Services," on page 105.

1.9.5 Administering Rights

- "Assigning Rights Explicitly" on page 68
- "Granting Equivalence" on page 69
- "Blocking Inherited Rights to an eDirectory Object or Property" on page 71
- "Viewing Effective Rights to an eDirectory Object or Property" on page 71

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 71.

- "Controlling Access to Novell eDirectory by Resource" on page 68
- "Controlling Access to Novell eDirectory by Trustee" on page 68

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 69
- "Granting Security Equivalence Explicitly" on page 70
- "Setting Up an Administrator For an Object's Specific eDirectory Properties" on page 70

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 96 for details.

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

See "Assigning Rights Explicitly" on page 68 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 *Modify Members of Groups* window.

  In Novell iManager, click *Roles and Tasks* > *Groups* > *Modify Members of Group*, specify the name and context of a Group object, and click *OK*. In the General tab, specify the members you want to add to the group and click *OK*.

- For a Role object, use the *Modify Object* window.

  In Novell iManager, click *Roles and Tasks* > *Directory Administration* > *Modify Object*, specify the name and context of an Organizational Role object, and click *OK*. Click *Other*, select *rbsMember*, and click *Edit*. On the Edit Attribute window, specify the members you want to add to the role and click *OK*.

4 Click *OK*.

Granting Security Equivalence Explicitly

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

2 Click *Directory Administration* > *Modify Object*.

3 Enter or browse to 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*, select or browse to the name and context of the object that you want the user to be equivalent in terms of security, then click *OK*.
- If you chose an object that you want the user to be security equivalent to, click *Security Equal To Me*, select or browse to the name and context of the user that you want the object to be equivalent to in terms of security, 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 96 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 file system, all rights assignments on folders are inheritable. In eDirectory, 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.

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	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: [All Attributes Rights]-Represents all the properties of the object. [Entry Rights]-Represents the object as a whole. Rights to this item don't imply any property rights, except in the case of Supervisor. 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).
Effective Rights	Shows the trustee's effective rights to the selected property, as calculated by eDirectory.
Show All Properties in Schema	Leave this check box deselected to show only the properties of this object class. 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.

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.

- Section 2.1, "eDirectory Design Basics," on page 73
- Section 2.2, "Designing the eDirectory Tree," on page 74
- Section 2.3, "Guidelines for Partitioning Your Tree," on page 80
- Section 2.4, "Guidelines for Replicating Your Tree," on page 82
- Section 2.5, "Planning the User Environment," on page 84
- Section 2.6, "Designing eDirectory for e-Business," on page 85
- Section 2.7, "Understanding the Novell Certificate Server," on page 86
- Section 2.8, "Synchronizing Network Time," on page 90

2.1 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 Section 2.6, "Designing eDirectory for e-Business," on page 85.

2.1.1 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

2.1.2 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

2.1.3 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 73 and "Organizational Structure" on page 74.

2.2 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 74
- "Designing the Upper Layers of the Tree" on page 77
- "Designing the Lower Layers of the Tree" on page 79

2.2.1 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 Novell Identity Manager to move data between eDirectory and other applications. For more information on Novell Identity Manager, see the *Novell Identity Manager 4.0.1 Administration Guide* (http://www.novell.com/documentation/idm401/).

Naming Conventions

- "Objects" on page 75
- "Server Objects" on page 75
- "Country Objects" on page 75

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 o, 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 to be 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 Windows 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.org/iso/country_codes/iso_3166_code_lists.htm).

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.

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.

2.2.2 Designing the Upper Layers of the Tree

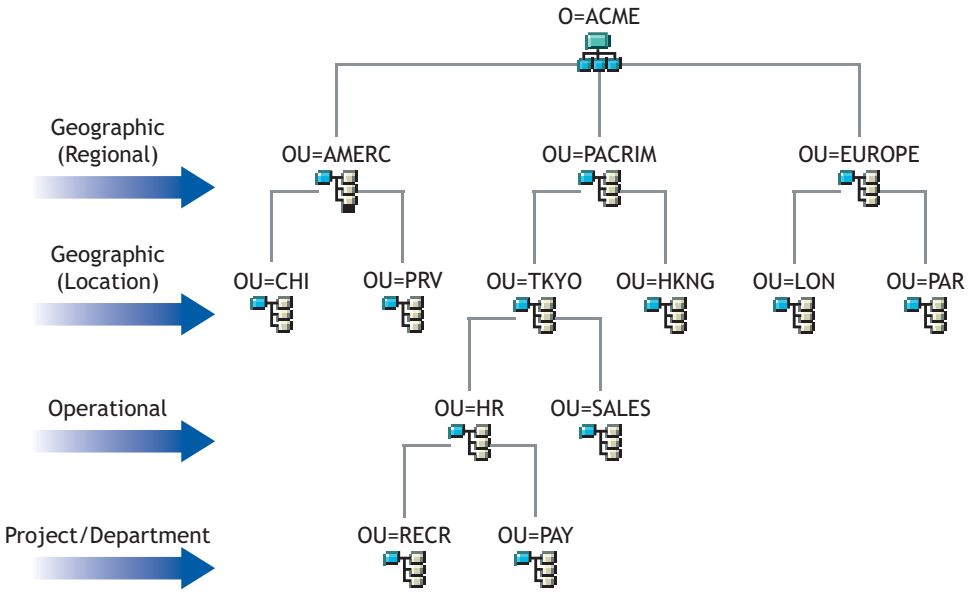
You should carefully design the upper layers of the tree because changes to the upper layers affect the rest of the tree, especially if your organization has WAN links. You want to design the top of the tree so that few changes will be necessary.

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 2-1 depicts the eDirectory design rules.

Figure 2-1 *eDirectory Design Rules*



To create the upper layers of the tree, see "Creating an Object" on page 96 and "Modifying an Object's Properties" on page 97.

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 Novell Identity Manager to simplify management. For more information on Novell Identity Manager, see the *Novell Identity Manager 4.0.1 Administration Guide (http://www.novell.com/documentation/idm401/)*.

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.

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, or AIX 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.

2.2.3 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 96 and "Modifying an Object's Properties" on page 97.

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 Windows, look at the DIB Set at `\novell\nds\dibfiles`.
- For Linux, Solaris, or AIX, 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.

2.3 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 Section 1.5, “Partitions,” on page 52. For information on creating partitions, refer to Chapter 5, “Managing Partitions and Replicas,” on page 135.

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.

2.3.1 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.

NOTE: 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 13, "WAN Traffic Manager," on page 293.

2.3.2 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.

2.3.3 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 ITB
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 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 59.

2.3.4 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

2.4 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 135.

The following guidelines will help determine your replica placement strategy.

- "Workgroup Needs" on page 82
- "Fault Tolerance" on page 82
- "Determining the Number of Replicas" on page 83
- "Replicating the Tree Partition" on page 83
- "Replicating for Administration" on page 83
- "Managing WAN Traffic" on page 84

2.4.1 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.

2.4.2 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 SP4 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. Consider purchasing another server for fault tolerance replication.

NOTE

- Some backup software does not back up eDirectory automatically.
- We recommend you exclude the DIB directory on your eDirectory server from any antivirus or backup software processes. Use the eDirectory Backup Tool to back up your DIB directory. For more information about backing up eDirectory, see "Backing Up and Restoring Novell eDirectory" on page 391.

2.4.3 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.

2.4.4 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.

2.4.5 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.

2.4.6 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 13, “WAN Traffic Manager,” on page 293.

2.5 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.

2.5.1 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 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.

2.5.2 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.

2.6 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 Novell Identity Manager to link this user tree to your other trees that contain network information. For more information, see the *Novell Identity Manager 4.0.1* (http://www.novell.com/documentation/idm401/).

- 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.

2.7 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. The first eDirectory SP4 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.

2.7.1 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/crt33/) online documentation.

2.7.2 Ensuring Secure eDirectory Operations on Linux, Solaris, and AIX 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 88
- "Initializing the NICI Module on the Server" on page 88
- "Starting the Certificate Server (PKI Services)" on page 89
- "Stopping the Certificate Server (PKI Services)" on page 89
- "Creating an Organizational Certificate Authority Object" on page 89
- "Creating a Server Certificate Object" on page 89
- "Exporting an Organizational CA's Self-Signed Certificate" on page 90

For information about using external certificate authority, refer to the *Novell Certificate Server 3.3 Administration Guide* (http://www.novell.com/documentation/crt33/crtadmin/data/a2ebomw.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" on page 88.

Initializing the NICI Module on the Server

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

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

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

     ```
     /etc/ndsd stop
     ```

   IMPORTANT: We recommend you to use ndsmanage to start and stop ndsd.
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

     ```
     lslpp -l | 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/init.d/ndsd start
     ```
   - On Solaris systems, enter:

     ```
     /etc/init.d/ndsd start
     ```
   - On AIX systems, enter:

     ```
     /etc/ndsd start
     ```

   IMPORTANT: We recommend you to use ndsmanage to start and stop ndsd.

Starting the Certificate Server (PKI Services)

To start PKI services, enter:

```
npki -1
```

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 Certificate Authority Object" (http://www.novell.com/documentation/crt33/crtadmin/data/fbgccghh.html) in the *Novell Certificate Server 3.3 Administration Guide*.
3. Click the *Roles and Tasks* button .
4. Click *Novell Certificate Server* > *Configure Certificate Authority*.

   If no Organizational Certificate Authority object exists, this opens the Create an Organizational Certificate Authority Object dialog box and the corresponding wizard that creates the object. Follow the prompts to create the object. For specific information on the dialog box or any of the wizard pages, click *Help*.

NOTE: You can have only one Organizational CA for your eDirectory tree. For more information about creating an Organizational CA, see "Creating an Organizational Certificate Authority Object" (http://www.novell.com/documentation/crt33/crtadmin/data/fbgccghh.html) in the *Novell Certificate Server 3.3 Administration Guide*.

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 a Server Certificate Object" (http://www.novell.com/documentation/crt33/crtadmin/data/fbgcdhec.html) in the *Novell Certificate Server 3.3 Administration Guide*.
3. Click the *Roles and Tasks* button .

4 Click *Novell Certificate Server* > *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 *Directory 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

2.8 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.

If your network uses Windows, Linux, Solaris, or AIX, you should use Network Time Protocol (NTP) to synchronize the servers, because it is a widely-used standard to provide time synchronization.

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

2.8.1 Synchronizing Time on Windows 2000 Servers

For information on time synchronization for Windows 2000 servers, see Setting Time Synchronization With Windows 2000 (http://www.netadmintools.com/art313.html) Web site.

2.8.2 Synchronizing Time on Linux, Solaris, or AIX Systems

You can use the xntpd Network Time Protocol (NTP) daemon to synchronize time on Linux, Solaris, and AIX 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 systems, see the Oracle Documentation Web site (http://www.oracle.com/technetwork/indexes/documentation/index.html).

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).

2.8.3 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.

Windows

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

Linux, Solaris, and AIX

1. Run the following command:

```
ndsrepair -T
```

3 Managing Objects

Novell eDirectory 8.8 includes Novell iManager 2.7, 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 2.7 Administration Guide* (http://www.novell.com/documentation/imanager27/imanager_admin_275/data/hk42s9ot.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 67 for more information.
- Configure role-based administration (define administrator roles for specific administrative applications through the role-based services object).

This chapter contains information on the following topics:

- Section 3.1, "General Object Tasks," on page 93
- Section 3.2, "Managing User Accounts," on page 98
- Section 3.3, "Configuring Role-Based Services," on page 105
- Section 3.4, "Synchronization," on page 110

3.1 General Object Tasks

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

- "Browsing the eDirectory Tree" on page 94
- "Creating an Object" on page 96
- "Modifying an Object's Properties" on page 97
- "Copying Objects" on page 97
- "Moving Objects" on page 97
- "Deleting Objects" on page 97
- "Renaming Objects" on page 98

3.1.1 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 94
- "Using the Object Selector Button" on page 95

Using the View Object Button

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

- "Using Browse" on page 94
- "Using Search" on page 95

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	Lets you specify the name of the container whose contents you want to view. To use this option, specify the name of the container you want, then click *Apply*.
Name	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*.
Type	Lets you specify the type of object you want to search for. The default is All Available Types. To use this option, select an object type from the drop-down list, then click *Apply*.

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 96
- “Using Search” on page 96

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*.

3.1.2 Creating an Object

1. In Novell iManager, click the *Roles and Tasks* button .
2. Click *Directory 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*.

3.1.3 Modifying an Object's Properties

1. In Novell iManager, click the *Roles and Tasks* button .
2. Click *Directory 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*.

3.1.4 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 *Directory 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*.

3.1.5 Moving Objects

1. In Novell iManager, click the *Roles and Tasks* button .
2. Click *Directory 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*.

3.1.6 Deleting Objects

1. In Novell iManager, click the *Roles and Tasks* button .
2. Click *Directory Administration* > *Delete Object*.

3 Specify the name and context of the object or objects you want to delete.

4 Click *OK*.

3.1.7 Renaming Objects

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

2 Click *Directory 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*.

3.2 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 98
- "Setting Up Optional Account Features" on page 99
- "Setting Up Login Scripts" on page 102
- "Login Time Restrictions for Remote Users" on page 103
- "Deleting User Accounts" on page 104

3.2.1 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 99
- "Modifying a User Account" on page 99

- "Enabling a User Account" on page 99
- "Disabling a User Account" on page 99

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*.

3.2.2 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	LDAP Attribute
Password Restrictions	Sets up a login password.	passwordRequired
Login Restrictions	◆ Enable or disable the account. ◆ Limit the number of concurrent login sessions. ◆ Set a login expiration and lockout date.	loginDisabled loginMaximumSimultaneous loginExpirationTime or loginGraceLimit
Time Restrictions	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.If the user will log in remotely, see "Login Time Restrictions for Remote Users" on page 103.	loginAllowedTimeMap
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.	networkAddressRestriction
Account Balance	Sets up an accounting of this user's server usage.	accountBalance
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.	lockedByIntruder

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 *Directory 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.
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. If the values of these three fields are specified as zero then the user account is locked indefinitely.

6 Click *OK*.

3.2.3 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 *Directory 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 *Novell Login Scripts Guide* (http://www.novell.com/documentation/linux_client/login/data/front.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 71 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*.

3.2.4 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.
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*.

3.2.5 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*.

3.3 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.

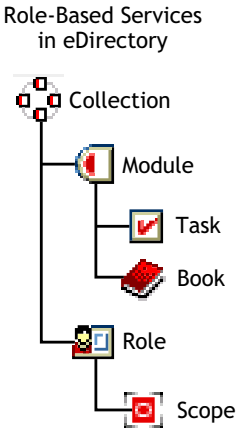
Role-Based Services is represented by objects defined in eDirectory. The base eDirectory schema gets extended during the iManager installation. The RBS object types are listed in the following table.

Object	Description
rbsCollection	A container object that holds all RBS Role and Module objects. 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. rbsCollection objects can be created in any of the following containers: ◆ Country ◆ Domain ◆ Locality ◆ Organization ◆ Organizational Unit
rbsRole	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. 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. rbsRole objects can be created only in rbsCollection containers.
rbsModule	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, NMAS, or Novell Certificate Access). rbsModule objects can be created only in rbsCollection containers.
rbsTask	A leaf object that represents a specific function, such as resetting login passwords. rbsTask objects are located only in rbsModule containers.

Object	Description
rbsBook	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. rbsBook objects are located only in rbsModule containers.
rbsScope	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. This object is dynamically created when needed, then automatically deleted when no longer needed. They are located only in rbsRole containers. **WARNING**: Never change the configuration of a Scope object. Doing so will have serious consequences and could possibly break the system.

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

Figure 3-1 *RBS Objects in the eDirectory Tree*



3.3.1 Defining RBS Roles

RBS roles specify the tasks that users are authorized to perform. Defining an RBS role includes creating an rbsRole 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 rbsTask objects in your eDirectory tree. These objects are added automatically during the installation of product packages. They are organized into one or more rbsModules, 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 108.

- "Creating a Role Object" on page 107
- "Modifying the Tasks Associated with a Role" on page 107

- "Assigning RBS Role Membership and Scope" on page 108
- "Deleting a Role-Based Services Object" on page 108

Creating a Role Object

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

1. In Novell iManager, click the *Configure* button .
2. Click *Role Based Services* > *RBS Configuration*.
3. Click the collection in which you want to create a new role.
4. Click the *Role* tab.
5. Click *New* > *iManager Role*.
6. Follow the instructions in the Create iManager Role Wizard.

See "Defining Custom RBS Tasks" on page 109 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 Based Services* > *RBS Configuration*.
3. Click the collection in which you want to modify a role.
4. Click the *Role* tab.
5. Click the role you want to modify.
6. (Optional) If you want to add tasks to a role, complete the following steps:
   - **6a** Click *Add*.
   - **6b** Use the arrow buttons to move tasks from the *All Tasks* list to the *Assigned Tasks* list, as necessary.
   - **6c** Click *OK*, then click *OK* again.
7. (Optional) If you want to remove tasks from a role, complete the following steps:
   - **7a** Select the tasks you want to remove and click *Remove*.
   - **7b** Click *OK*, then click *OK* again.
8. When finished, click *Close*.

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 Based Services* > *RBS Configuration*.
3. Click the collection in which you want to modify a role.
4. Click the *Role* tab.
5. Select the role you want to modify.
6. Click *Actions* > *Member Associations*.
7. (Optional) If you want to add a member to the role, complete the following steps:
   - **7a** In the *Name* field, specify the name of the object you want to add (a User, Group, or Container object) and context.
   - **7b** In the *Scope* field, specify an Organization or Organizational Unit object name and context.
   - **7c** Click *Add*.
8. (Optional) If you want to remove a member from the role, complete the following steps:
   - **8a** In the list of current role members, select the member you want to remove.
   - **8b** Click *Remove*.
9. When finished, click *OK*, then click *OK* again.
10. Click *Close*.

Deleting a Role-Based Services Object

1. In Novell iManager, click the *Configure* button .
2. Click *Role Based Services* > *RBS Configuration*.
3. Click the collection in which you want to delete an RBS role.
4. Click the *Role* tab.
5. Select the role you want to modify.

6 Click *Delete*.

7 Click *OK*.

8 When finished, click *OK*.

9 Click *Close*.

3.3.2 Defining Custom RBS Tasks

- "Creating an iManager Task" on page 109
- "Modify Role Assignment" on page 109
- "Deleting a Task" on page 109

Creating an iManager Task

1 In Novell iManager, click the *Configure* button .

2 Click *Role Based Services* > *RBS Configuration*.

3 Click the collection in which you want to create a new task.

4 Click the *Task* tab.

5 Click *New* > *iManager Task*.

6 Follow the instructions in the Task Builder to create a custom task.

Modify Role Assignment

1 In Novell iManager, click the *Configure* button .

2 Click *Role Based Services* > *RBS Configuration*.

3 Click the collection in which you want to modify a task.

4 Click the *Task* tab.

5 Select the task you want to modify.

6 Click *Actions* > *Role Assignment*.

7 Move the roles you want from the *Available Roles* column to the *Assigned Roles* column.

8 Click *OK*, then click *OK* again.

9 Click *Close*.

Deleting a Task

1 In Novell iManager, click the *Configure* button .

2 Click *Role Based Services* > *RBS Configuration*.

3 Click the collection in which you want to delete a task.

4 Click the *Task* tab.

5 Select the task you want to delete.

6 Click *Delete*.

7 Click *OK*.

8 When finished, click *OK*.

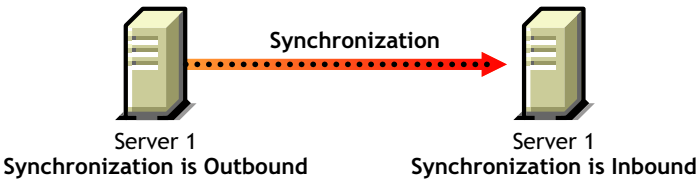
9 Click *Close*.

3.4 Synchronization

Synchronization is the transfer of directory information from one replica to another, so the information in each partition is consistent with the other. eDirectory automatically keeps the servers in the replica ring synchronized.

Synchronization consists of inbound and outbound synchronization. For example, if the modifications to data have to be synchronized from server1 and server2, the term *outbound* refers to the synchronization process that is sent from server1 to server2. The term *inbound* refers to the synchronization process that is received by server2 from server1.

Figure 3-2 *Outbound and Inbound Synchronization*



There are two types of synchronization:

- Normal Synchronization or Replica Synchronization
- Priority Sync (in eDirectory 8.8 or later).

The following table gives you a comparison between normal synchronization and priority sync:

Table 3-1 *Comparison between Normal or Replica Synchronization and Priority Sync*

Normal Synchronization or Replica Synchronization	Priority Sync
Triggered when there are modifications to data in any of the servers in the replica ring.	Triggered when there are modifications only to the data that you identify as critical.
For more information, refer to Section 3.4.2, “Normal or Replica Synchronization,” on page 113.	For more information, refer to Section 3.4.3, “Priority Sync,” on page 115.
After the data is modified, the changes are buffered. Normal synchronization starts after approximately 30 seconds from the time the modifications are saved.	The changes to the critical data are not buffered. Priority sync starts immediately after the data is modified.
The most important synchronization in eDirectory. It happens irrespective of whether the modifications are synchronized by priority sync or not.	Complementary to normal synchronization. Though the critical attributes are synchronized through priority sync, they are synchronized again through normal synchronization.
Can happen between eDirectory 8.8 servers or across servers hosting earlier versions of eDirectory.	Happens only between eDirectory 8.8 servers, holding the same partition.
Never fails due to its feature. For more information, refer to Section 3.4.1, “Features of Synchronization,” on page 111.	If priority sync fails, the modifications to the critical data are synchronized through normal synchronization. For more information, refer to “When Can Priority Sync Fail?” on page 120.

3.4.1 Features of Synchronization

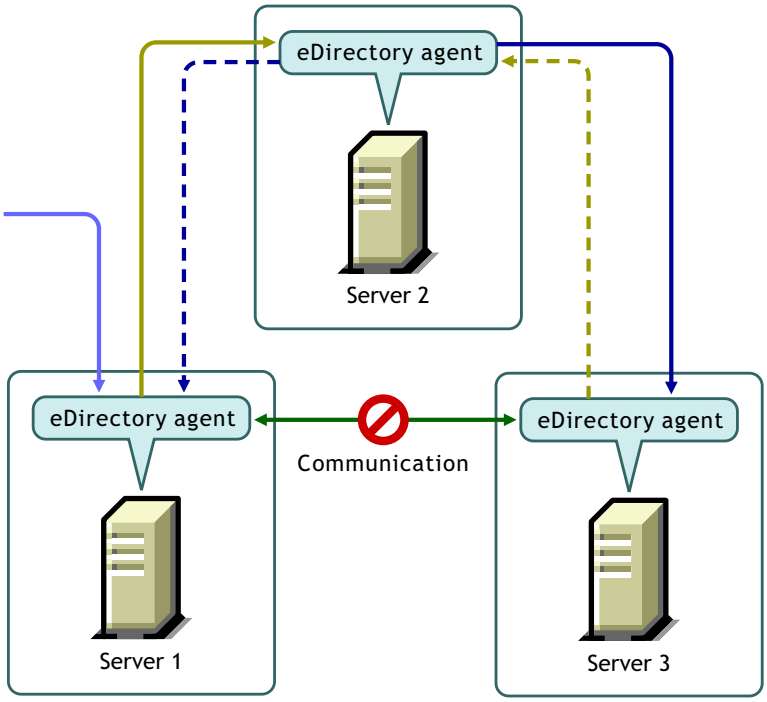
Synchronization in eDirectory

- Is transitive.
- Maintains object transaction model.
- Has timestamps like transitive vector, local received up to and remote received up to.

Transitive Synchronization

Synchronization in eDirectory is transitive. This means that eDirectory synchronizes the changes to the data without requiring the eDirectory agent to directly contact and synchronize those changes with every other agent in the replica ring.

Figure 3-3 *Transitive Synchronization*



For example, if you make a change to data on Server 1, the change is synchronized from Server 1 to Server 2 and from Server 2 to Server 3. Even if Server 1 could not come into direct contact with Server 3, because of a problem in communication, it still receives the latest change to the data, through Server 2. Server 3 lets Server 2 know that it has received the changes. Server 2 in turn tells Server 1 that Server 3 and itself are synchronized.

Object Transaction Model

Synchronization in eDirectory maintains the object transaction model, a standard for LDAP and X.500-compliant directories. Object transaction model means that all the previous transactions should be synchronized before synchronizing the new ones.

For example, you have modifications D1, D2, and D3 to the data on a server. Due to network failure, these modifications are not synchronized across other servers. If you make another modification D4 on the server, D4 will be synchronized only after D1, D2, and D3 are synchronized across all the servers in the replica ring.

Transitive Vector

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

For more information, refer to "Transitive Vectors and the Restore Verification Process" on page 404.

Local Received Up To

Local Received Up To (LRUT) is the time before which the local replica has received the changes.

For more information, refer to "Browsing Objects in Your Tree" on page 214.

Remote Received Up To

Remote Received Up To (RRUT) is the LRUT of the remote replica.

For more information, refer to "Browsing Objects in Your Tree" on page 214.

3.4.2 Normal or Replica Synchronization

Normal or Replica Synchronization is one of the two synchronization processes in eDirectory. Normal synchronization synchronizes all the modifications to data on a server with other servers in the replica ring.

Normal synchronization happens across all servers having any version of eDirectory, having the same partition.

For more information, refer to Section 5.5, "Administering Replicas," on page 139.

You can enable or disable normal synchronization by enabling or disabling outbound and inbound synchronization in Novell iMonitor. Both inbound and outbound synchronizations are enabled by default. To sync the modifications to data across the other servers through normal synchronization, you need to configure the synchronization parameters in iMonitor. Refer to "Controlling and Configuring the DS Agent" on page 209 for more information.

In normal synchronization, when you make any modifications to the data, the changes you make are buffered before synchronizing them across the servers. You can view the synchronization status in the servers of your setup in iMonitor. Refer to "Browsing Objects in Your Tree" on page 214 for more information.

Normal synchronization maintains the object transaction model and is transitive. Refer to "Transitive Synchronization" and "Object Transaction Model" on page 101 for more information.

Configuring Normal Synchronization

You can configure normal synchronization using Agent Configuration under Agent Synchronization in iMonitor.

This section provides the following information:

- "Enabling/Disabling Normal Synchronization" on page 114
- "Enabling/Disabling Inline Cache" on page 114
- "Synchronization Threads" on page 114
- "Synchronization Method" on page 114

Enabling/Disabling Normal Synchronization

You can enable or disable normal synchronization by enabling or disabling the outbound and inbound synchronization in iMonitor. Refer to "Controlling and Configuring the DS Agent" on page 209 for more information.

Outbound synchronization is enabled by default. When you disable this option on a server, the modifications to the data on this server are not synchronized with other servers. You can specify the amount of time, in hours, for which you want the outbound synchronization disabled. The default which is also the maximum time is 24 hours. After the specified time, the modifications to the data on this server are synchronized with other servers.

Inbound synchronization is enabled by default. When you disable this option for a server, the modifications to the data on other servers are not synchronized with this server.

Enabling/Disabling Inline Cache

You can enable or disable the Inline Change Cache for a server. You can disable Inline Change Cache only when Outbound Synchronization is disabled. Enabling Outbound Synchronization also enables Inline Change Cache.

Disabling Inline Change Cache marks the change cache as invalid for this replica and tags it with an invalid flag in *Agent Configuration* > *Partitions*. Enabling Inline Change Cache removes the invalid change cache flag when the change cache is rebuilt.

Synchronization Threads

For outbound synchronization, you need to configure the synchronization threads. Using iMonitor, you can specify the number of synchronization threads using *Agent Configuration* under *Agent Synchronization*.The supported values are 1 to 16.

See "Controlling and Configuring the DS Agent" on page 209 for more information.

Synchronization Method

Normally, eDirectory automatically chooses the method based on the number of replicas and replication partners. The following are the synchronization methods:

- **By Partition:** The modifications to data are synchronized simultaneously with other replicas.Several threads are used to synchronize the modifications. For example, D1, D2, and D3 are modifications to data on replica R1, and these have to be synchronized across replicas R2 and R3, D1, D2, and D3 are simultaneously synchronized with R2 and R3.
- **By Server:** Modifications to data are synchronized sequentially. Only one thread is used to sync the modifications. For example, D1, D2, and D3 are modifications to data on replica R1. These have to be synchronized across replicas R2 and R3, D1 is first synchronized with R2 and R3. Then D2 is synchronized with R2 and R3.
- **By Dynamic Adjust:** Based on the system resources you have allotted, eDirectory automatically chooses the synchronization method.

Using iMonitor, you can specify the method of synchronization using *Agent Configuration* under *Agent Synchronization*. For more information, refer to "Controlling and Configuring the DS Agent" on page 209.

3.4.3 Priority Sync

Priority Sync is one of the two synchronization processes in eDirectory. In eDirectory 8.8 and later, you can use priority sync when you need to sync your critical data immediately and cannot wait for normal synchronization.

Priority sync is complimentary to the normal synchronization process in eDirectory. Unlike normal synchronization, in priority sync, the changes are not buffered before synchronizing them across the servers. This makes priority sync faster than normal synchronization.

Priority sync is enabled by default. Refer to "Enabling and Disabling Inbound and Outbound Priority Sync" on page 115 for more information.

To sync the modifications to the critical data through priority sync:

1. Specify the number of threads for priority sync.

   See "Priority Sync Threads" on page 116 for more information.

2. Specify the priority sync queue size.

   See "Priority Sync Queue Size" on page 116 for more information.

3. Create and define a priority sync policy by identifying the critical attributes that you want to sync as priority.

   See "Creating and Defining a Priority Sync Policy" on page 118 for more information.

4. Apply the priority sync policy to one or more partitions.

   See "Applying a Priority Sync Policy" on page 119 for more information.

The priority sync process is to sync only the modifications to the critical attributes. Priority sync maintains the object transaction model. So, if noncritical data is modified and is not yet synchronized, and if the critical data is changed for the same entry, the noncritical data along with critical data is synchronized.

For example, a user has the following attributes: Income, Employee No, Address, and Cube No. You identify Income and Address as critical attributes. Employee No and Cube No are modified but these modifications are not yet synchronized. When the modifications to Income and Address are synchronized through priority sync, Employee No and Cube No also get synchronized, though they are not identified as critical data.

This section provides you the following information:

- "Enabling and Disabling Inbound and Outbound Priority Sync" on page 115
- "Priority Sync Threads" on page 116
- "Priority Sync Queue Size" on page 116
- "Managing Priority Sync Policies" on page 117
- "When Can Priority Sync Fail?" on page 120

Enabling and Disabling Inbound and Outbound Priority Sync

You can enable or disable the inbound and outbound priority sync in eDirectory 8.8 or later using iMonitor. Refer to "Controlling and Configuring the DS Agent" on page 209 for more information.

Inbound priority sync is enabled by default. By disabling the inbound priority sync on a server, the modifications to the critical data on other servers are not synchronized with this server through priority sync. However, the modifications are synchronized by the normal synchronization process.

Outbound priority sync is enabled by default. By disabling this option on a server, the modifications to the critical data on this server are not synchronized with other servers through priority sync. However, the modifications are synchronized by the normal synchronization process.

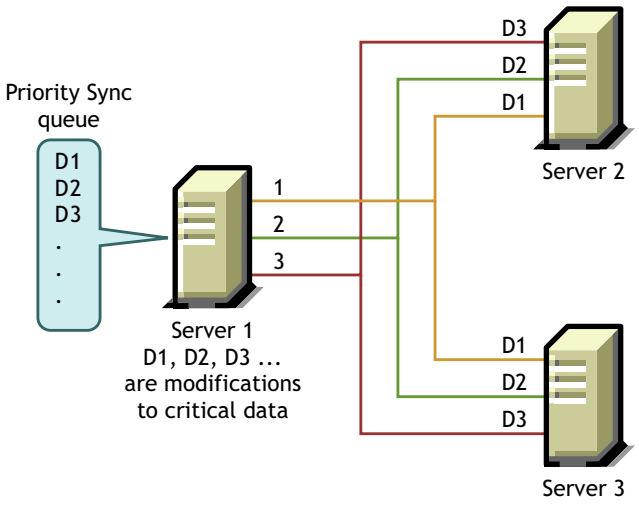
Priority Sync Threads

You need to configure the number of threads to be used for outbound priority sync. In iMonitor, you can specify the number of priority sync threads using *Agent Configuration* under *Agent Synchronization*. For more information, refer to "Controlling and Configuring the DS Agent" on page 209. The supported values are 1 to 32. The default is 4.

Priority Sync Queue Size

This indicates the maximum number of modified critical entries the queue can hold before synchronizing them. As soon as you modify the critical entries, they go into the priority sync queue and are synchronized one after the other. For example, if D1, D2, and D3 are the critical entries that are modified on server1 and these entries have to be synchronized across server2 and server3 through priority sync, then D1 is first synchronized with server2 and server3. Then D2 is synchronized with server2 and server3, and later D3 is synchronized with server2 and server3. If an earlier entry in the queue is not successfully synchronized with one of the servers, it does not affect the synchronization of the rest of the entries.

Figure 3-4 *Priority Sync Queue*



You can specify the priority sync queue size in iMonitor using *Agent Configuration* under *Agent Synchronization*. For more information, refer to "Controlling and Configuring the DS Agent" on page 209.

During a priority sync process, if a number of modifications happen at short intervals, the queue reaches its maximum size then, the queue expires and a new queue is formed. The modifications in the older queue that are not yet synchronized, will be synchronized by normal synchronization.

The queue size for priority sync can vary from 0 to 2^{32} - 1. By default, this value is 2^{32} - 1. If the Priority Sync queue size is set to 0, no modifications are synchronized through priority sync. These modifications are synchronized by normal synchronization.

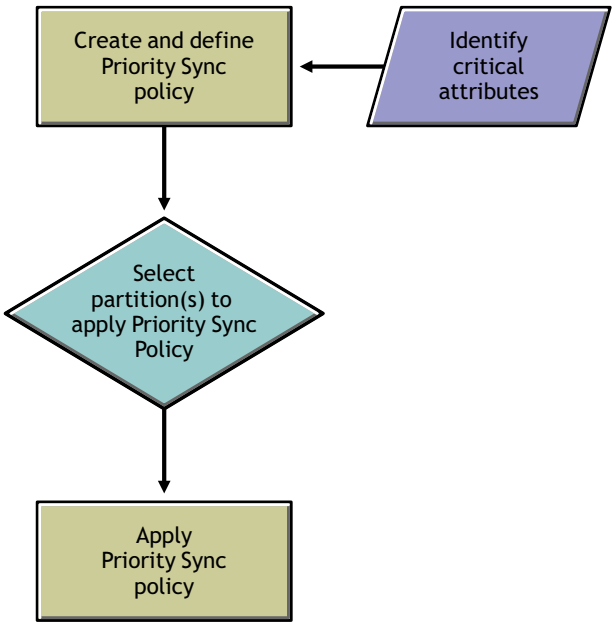
The value -1 implies unlimited queue size. -1 is 2^{32} - 1. When a negative value is specified, for example, -3, it means -3 = -1-2, which is 2^{32} - 1-2.

Managing Priority Sync Policies

You can manage priority sync by creating and defining policies and applying them to partitions through iManager or LDAP. You define a priority sync policy by identifying the attributes that are critical.

NOTE: Plug-ins are available only in Novell iManager 2.6 and later.

Figure 3-5 *Priority Sync process*



For example, if the attributes Password and Account Number are critical, you can create a priority sync policy PS1 that contains these attributes. You can then apply the policy PS1 to a partition P1. If you change the password or the account number of an entry on a server, the changes are immediately synchronized with other servers having partition P1.

For priority sync to happen, you need to check if outbound and inbound priority sync are enabled in iMonitor. Inbound and outbound priority sync are enabled by default. If you disable inbound and outbound priority sync, the modifications to the data are synchronized by normal synchronization.

For more information, see "Controlling and Configuring the DS Agent" on page 209.

This section provides the following information:

- "Creating and Defining a Priority Sync Policy" on page 118
- "Editing a Priority Sync Policy" on page 118
- "Applying a Priority Sync Policy" on page 119
- "Deleting a Priority Sync Policy" on page 120

When you create a child partition, the priority sync policy that is applied to the parent is inherited by the child partition. When you merge partitions, the priority sync policy of the parent is retained.

Creating and Defining a Priority Sync Policy

You can define a priority sync policy by selecting the attributes either directly or through an object class. When you select attributes through an object class, all the attributes under the object class are selected for priority sync. You can choose to select the mandatory or optional attributes for priority sync.

The priority sync policy can be created anywhere in the eDirectory tree using either iManager or LDAP.

Using iManager:

1. Click the *Roles and Tasks* button .
2. Click *Partition and Replica Management* > *Priority Sync Policies*.
3. In the Priority Sync Policies Management Wizard, select *Create, Edit and Apply policy*.
4. Click *Next*.
5. Follow the instructions in the Create Priority Sync Policy Wizard to create the policy.

   Help is available throughout the wizard.

Using LDAP:

To create an empty priority sync policy:

```
dn:cn=policy1,o=policies
changetype:add
objectclass:prsyncpolicy
```

To define the priority sync policy by marking the attributes for priority sync:

```
dn:cn=policy2,o=policies
changetype:add
objectclass:prsyncpolicy
prsyncattributes:description
```

In the above example, Description is the attribute marked for priority sync.

Editing a Priority Sync Policy

You can edit a Priority Sync Policy object using iManager or LDAP.

Using iManager

1. Click the *Roles and Tasks* button .
2. Click *Partition and Replica Management* > *Priority Sync Policies*.
3. In the Priority Sync Policies Management Wizard, select *Edit policy*.
4. Click *Next*.
5. Follow the instructions in the Edit Priority Sync Policy Wizard to edit the policy.

   Help is available throughout the wizard.

Using LDAP

In the following example, the priority sync policy is modified by marking Surname for priority sync instead of Description.

```
dn:cn=policy2,o=policies

changetype:modify

add:prsyncattributes

prsyncattributes:surname
```

To remove an attribute that is marked priority sync from the priority sync policy:

```
dn:cn=policy2,o=policies

changetype:modify

add:prsyncattributes

prsyncattributes:description
```

In the above example, the attribute Description is removed from the priority sync policy.

Applying a Priority Sync Policy

You can apply one priority sync policy to many partitions, but not more than one policy to a partition.

You can apply a priority sync policy to a partition using either iManager or LDAP.

Using iManager:

1. Click the *Roles and Tasks* button .
2. Click *Partition and Replica Management* > *Priority Sync Policies*.
3. In the Priority Sync Policies Management Wizard, select *Create, Edit and Apply policy*.
4. Follow the instructions in the Apply Priority Sync policy Wizard to apply the policy.

   Help is available throughout the wizard.

Using LDAP:

To apply a priority sync policy to a root partition:

```
dn:

changetype:modify

add:prsyncpolicydn

prsyncpolicydn:cn=policy2,o=policies
```

In the above example, policy2 is applied to the root partition.

To apply a priority sync policy to a nonroot partition:

```
dn:o=org

changetype:modify

add:prsyncpolicydn

prsyncpolicydn:cn=policy2,o=policies
```

In the above example, policy2 is applied to the nonroot partition.

To replace a priority sync policy for a nonroot partition:

```
dn:o=org

changetype:modify

replace:prsyncpolicydn

prsyncpolicydn:cn=policy1,o=policies
```

In the above example, policy2 is replaced by policy1.

To disassociate a priority sync policy with a nonroot partition:

```
dn:o=org

changetype:modify

delete:prsyncpolicydn
```

In the above example, the priority sync policy is disassociated from the nonroot partition O=Org.

Deleting a Priority Sync Policy

You can delete a priority sync policy using either iManager or LDAP.

Using iManager:

1. Click the *Roles and Tasks* button .
2. Click *Partition and Replica Management* > *Priority Sync Policies*.
3. In the Priority Sync Policies Management Wizard, select *Delete policies*.
4. Follow the instructions in the Delete Priority Sync policy Wizard to delete the policy.

   Help is available throughout the wizard.

Using LDAP:

```
dn:cn=policy1,o=policies

changetype:delete
```

NOTE: For more information on creating and managing priority sync policies, see Section 14.3, “Using LDAP Tools on Linux, Solaris, or AIX,” on page 337 and Section 6.1, “Novell Import Conversion Export Utility,” on page 147.

When Can Priority Sync Fail?

Priority sync can fail under any of the following circumstances:

- Network failure: Priority sync will not store modifications if it is unable to send them to the remote server in the case of network failure.
- Priority sync queue size reaches its maximum: Priority sync will ignore the changes in the priority sync queue if the number of entries exceeds the priority sync queue size.
- Failure in schema synchronization: If the schema is not synchronized, priority sync process will fail.

- Object does not exist on other servers: If the creation of the object is itself not synchronized, priority sync fails.
- Mixed servers in the replica ring: If you have both eDirectory 8.8 and pre-eDirectory 8.8 servers, priority sync fails.

When priority sync fails because of any of the above reasons, the changes to the critical data are synchronized through normal synchronization.

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.

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:

- Section 4.1, "Extending the Schema," on page 124
- Section 4.2, "Viewing the Schema," on page 127
- Section 4.3, "Manually Extending the Schema," on page 128
- Section 4.4, "Schema Flags Added in eDirectory 8.7," on page 130
- Section 4.5, "Using the Client to Perform Schema Operations," on page 132

For more detailed schema information, see the *Novell eDirectory Schema Reference* (http://developer.novell.com/documentation/ndslib/schm_enu/data/h4q1mn1i.html).

4.1 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

4.1.1 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 125 for more information.

4.1.2 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*.

4.1.3 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.

NOTE: Due to a replication issue, attributes in eDirectory other than the stream attribute type cannot contain values larger than 60 KB or 30,000 characters. If a user or application sets the value of a string or binary attribute and exceeds that limit, eDirectory returns a -649 error indicating that the value is too long.

To create a new attribute:

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.

4.1.4 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.

4.1.5 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*.

4.1.6 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.

4.1.7 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 127 instead.
No	Click *Add*, select the auxiliary class, then click *OK*.

5 Click *Close*.

4.1.8 Modifying an Object's Auxiliary Properties

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

2 Click *Directory 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 the *Novell eDirectory Schema Reference* (http://developer.novell.com/documentation/ndslib/schm_enu/data/h4q1mn1i.html).

6 Click *Apply*, then click *OK*.

4.1.9 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*.

4.2 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.

4.2.1 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.

4.2.2 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.

4.3 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 Windows" on page 128
- "Extending the Schema on Linux, Solaris, or AIX Systems" on page 129

4.3.1 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*.

4.3.2 Extending the Schema on Linux, Solaris, or AIX Systems

The following sections provide information about extending the schema on Linux, Solaris, and AIX systems:

- "Using the ndssch Utility to Extend the Schema on Linux, Solaris, or AIX" on page 129
- "Extending the RFC 2307 Schema" on page 129

Using the ndssch Utility to Extend the Schema on Linux, Solaris, or AIX

In addition to Novell iManager, you can use ndssch, the eDirectory schema extension utility, to extend the schema on Linux, Solaris, or AIX 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.

Use the following syntax:

```
ndssch [-h hostname[:port]] [-t tree_name] [-F <logfile>] admin-FDN schemafile...
```

```
ndssch [-h hostname[:port]] [-t tree_name] [-d] admin_FDN schemafile
[schema_description]...
```

ndssch Parameter	Description
`-h hostname`	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.
port	The server port.
`-t tree_name`	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/opt/novell/eDirectory/conf/nds.conf` file. For more information, see "Configuration Parameters" in the *Novell eDirectory 8.8 SP7 Installation Guide*.
`-F logfile`	Specifies the path name to the ndssch log file.
admin-FDN	Name with the full context of the user with eDirectory administrator rights to the tree.
schemafile	Filename that contains information about the schema to be extended.
`-d`, *schema_description*	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) are user or group related and NIS related. The user- or group-related definitions are compiled into the `/opt/novell/eDirectory/lib/nds-modules/schema/rfc2307-usergroup.sch` file. The NIS-related definitions are compiled into the `/opt/novell/eDirectory/lib/nds-modules/schema/rfc2307-`

nis.sch file. The corresponding files in the LDIF format are also provided (/opt/novell/eDirectory/lib/nds-modules/schema/rfc2307-usergroup.ldif and /opt/novell/eDirectory/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 130
- "Using the ldapmodify Utility" on page 130

Using the ndssch Utility

Enter one of the following commands:

```
ndssch -t tree_name admin-FDN /opt/novell/eDirectory/lib/nds-schema/rfc2307-usergroup.sch
```

or

```
ndssch -t tree_name admin-FDN /opt/novell/eDirectory/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 /etc/opt/novell/eDirectory/conf/nds.conf file.

Using the ldapmodify Utility

Enter one of the following commands:

```
ldapmodify -h -D -w -f /opt/novell/eDirectory/lib/nds-schema/rfc2307-usergroup.ldif
```

or

```
ldapmodify -h -D -w -f /opt/novell/eDirectory/lib/nds-schema/rfc2307-nis.ldif
```

Parameter	Description
-h *ldaphost*	Specifies an alternate host on which the LDAP server is running.
-D *binddn*	Uses *binddn* to bind to the X.500 directory. It should be a string-represented DN as defined in RFC 1779.
-w *passwd*	Uses *passwd* as the password for simple authentication.
-f *file*	Reads the entry modification information from file instead of from standard input.

4.4 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 or later versions, so for consistent behavior relating to this flag, the entire tree must be upgraded to eDirectory 8.7 or later versions of eDirectory.

Because only an eDirectory 8.7.*x* (or later versions) server will recognize these new flags, they can be set only on a schema definition by an eDirectory 8.7.*x* (or later versions) 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* (or later version) 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*.
Linux, Solaris, and AIX	Use the ndssch utility. See "Using the ndssch Utility to Extend the Schema on Linux, Solaris, or AIX" on page 129 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.

4.5 Using the 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 Section 12.6, "Maintaining the Schema," on page 278.).

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 Client, see Section 21.1, "Using the Command Line Client," on page 516.

4.5.1 Using the DSSchema eMTool

1. Run the Client in interactive mode by entering the following at the command line:

   ```
   java -cp path_to_the_file/emboxclient.jar  -i
   ```

   (If you have already put the `emboxclient.jar` file in your class path, you only need to enter `java -i`.)

   The Client prompt appears:

   ```
   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 8028, unless you have a Web server that is already using the port. The `-n` option opens a nonsecure connection.

   The 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 -n MyTree` imports remote schema from the tree named MyTree.

   A space must be between each switch. The order of the switches is not important.

   The Client will indicate whether the repair is successful.

   See "DSSchema eMTool Options" on page 133 for more information on the DSSchema eMTool options.

4. Log out from the Client by entering the following command:

   ```
   logout
   ```

5. Exit the Client by entering the following command:

   ```
   exit
   ```

4.5.2 DSSchema eMTool Options

The following tables lists the DSSchema eMTool options. You can also use the `list -t dsschema` command in the Client to list the DSSchema options with details. See "Listing eMTools and Their Services" on page 519 for more information.

Option	Description
`rst`	Synchronizes the schema of the master replica of the root of the tree to this server.
`irs -n`*`tree_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.
`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	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. Filtered replicas allow administrators to create sparse and fractional replicas. ◆ Sparse replicas contain only the object classes that you specify. ◆ Fractional replicas contain only the attributes you specify. 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.
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.

- Section 5.1, “Creating a Partition,” on page 136
- Section 5.2, “Merging a Partition,” on page 136

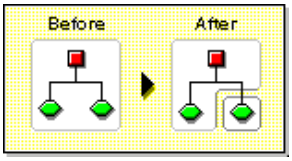
- Section 5.3, “Moving Partitions,” on page 137
- Section 5.4, “Cancelling Create or Merge Partition Operations,” on page 139
- Section 5.5, “Administering Replicas,” on page 139
- Section 5.6, “Setting Up and Managing Filtered Replicas,” on page 142
- Section 5.7, “Viewing Partitions and Replicas,” on page 145

5.1 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 5-1 *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.

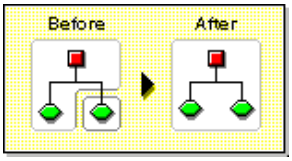
To create a partition:

1. In Novell iManager, click the *Roles and Tasks* button .
2. Click *Partition and Replica Management* > *Create Partition*.
3. Specify the name and context of the container you want to create a new partition from, then click *OK*.

5.2 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 5-2 *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 and 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.

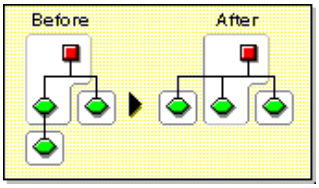
To merge a child partition with its parent partition:

1. In Novell iManager, click the *Roles and Tasks* button .
2. Click *Partition and Replica Management* > *Merge Partition*.
3. Specify the name and context of the partition you want to merge with its parent partition, then click *OK*.

5.3 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 5-3 *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 Replica Management* > *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*.

5.4 Cancelling 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.

5.5 Administering Replicas

Before you add or delete a replica, or change replica type, carefully plan target replica locations. See Section 2.4, "Guidelines for Replicating Your Tree," on page 82.

5.5.1 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 Replica Management* > *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 56.

5.5.2 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 Replica Management* > *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*.

5.5.3 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 Replica Management* > *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.
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 56.

5.6 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 142
- "Defining a Partition Scope" on page 143
- "Setting Up a Server Filter" on page 144

5.6.1 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 Replica Management* > *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 144.

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 143.

7 Click *Next*, then click *Finish*.

5.6.2 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 59.

Viewing Replicas on an eDirectory Server

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

2 Click *Partition and Replica Management* > *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 Replica Management* > *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 Replica Management* > *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*.

5.6.3 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 man 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 144
- "Using the Server Object" on page 144

Using the Replica View

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

2 Click *Partition and Replica Management* > *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 *Directory 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.

5.7 Viewing Partitions and Replicas

This section contains the following information:

- "Viewing the Partitions on a Server" on page 145
- "Viewing a Partition's Replicas" on page 145
- "Viewing Information about a Partition" on page 145
- "Viewing Partition Hierarchy" on page 146
- "Viewing Information about a Replica" on page 146

5.7.1 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 Replica Management* > *Replica View*.
3. Enter the name and context of a Server object, then click *OK*.

5.7.2 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 Replica Management* > *Replica View*.
3. Enter the name and context of a partition, then click *OK*.

5.7.3 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 Replica Management* > *View Partition Information*.
3. Enter the name and context of a partition, then click *OK*.

5.7.4 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: .

5.7.5 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 Replica Management* > *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:

- Section 6.1, “Novell Import Conversion Export Utility,” on page 147
- Section 6.2, “Index Manager,” on page 184
- Section 6.3, “Predicate Data,” on page 188
- Section 6.4, “eDirectory Service Manager,” on page 189

6.1 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, and 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” 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 148
- “Using the Command Line Interface” on page 155

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.

6.1.1 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.
- Add data from an LDIF or schema file to a server.
- Add data from one server to another server.
- Compare data between an LDIF or schema file and another LDIF file.
- Compare data between a server and an LDIF file.
- Generate an order file.

For information on using and accessing Novell iManager, see the *Novell iManager 2.7 Administration Guide* (http://www.novell.com/documentation/imanager27/imanager_admin_275/data/hk42s9ot.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.

Updating Schema from a File

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

2 Click *eDirectory Maintenance* > *Import Convert Export Wizard*.

3 Click *Add Schema from a File* > *Next*.

4 Select the type of file you want to add.

You can choose between LDIF and schema file types.

5 Specify the name of the file containing the schema you want to add, specify the appropriate options, then click *Next*.

Select *Do Not Add but Compare Schema* if you want to only compare the schema and not add the additional schema to the destination server. The additional schema is not added to the destination server, but the differences between the schema is available to you in a link at the end of the operation.

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 schema is to be imported.

7 Add the appropriate options, 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* > *Finish*.

Adding Schema from a Server

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

2 Click *eDirectory Maintenance* > *Import Convert Export Wizard*.

3 Click *Add Schema from a Server* > *Next*.

4 Specify the LDAP server that the schema is to be added from.

5 Add the appropriate options, 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

Select *Do Not Add but Compare Schema* if you want to only compare the schema and not add the additional schema to the destination server. The additional schema is not added to the destination server, but the differences between the schema is available to you in a link at the end of the operation.

6 Specify the LDAP server where the schema is to be added.

7 Add the appropriate options, 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* > *Finish*.

Comparing Schema Files

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

2 Click *eDirectory Maintenance* > *Import Convert Export Wizard*.

3 Click *Compare Schema Files* > *Next*.

4 Select the type of file you want to compare.

You can choose between LDIF and schema file formats.

5 Specify the name of the file containing the schema you want to compare, 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 schema file you want to compare it with.

You can select only an LDIF file.

7 Click *Next* > *Finish*.

The differences between the two schema files are available to you in a link at the end of the operation.

Comparing Schema from Server and File

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

2 Click *eDirectory Maintenance* > *Import Convert Export Wizard*.

3 Click *Compare Schema between Server and File* > *Next*.

4 Specify the LDAP server that the schema is to be compared from.

5 Add the appropriate options, 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

6 Select the type of file you want to compare with.

7 Specify the name of the file containing the data you want to compare, 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.

8 Click *Next* > *Finish*.

The differences between the server's schema and the schema file are available to you in a link at the end of the operation.

Generating an Order File

This option creates an order file for use with the DELIM handler to import data from a delimited data file. The wizard helps you to create this order file that contains a list of attributes for a specific object class.

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

2 Click *eDirectory Maintenance* > *Import Convert Export Wizard*.

3 Click *Generate Order File*, then click *Next*.

4 Select the class for which you want to generate the order file and click *View*.

Select the attributes you want add it to the *Sequenced Attributes* list.

Select the auxiliary class and add it to the *Select Auxiliary Classes* list.

For more information on Sequenced Attributes list and Auxiliary Classes list, refer to the iMonitor online help.

Click Next.

5 Add the appropriate options, as described in the following table:

Option	Description
Context	Context where the objects created would be associated to
Select the Data File	Location of the data file
Select the Delimiter in the Data File	Delimiter that would be used within the data file. The default delimiter is a comma (,)
Select the Naming Attribute	Naming attributes from the list of all available attributes from the selected class

Use the *Advanced Settings* to configure additional options for the LDAP source handler. Click *Help* for more information on the available options.

Use the *Records to Process* to select the records to be processed in the data file. Click *Help* for more information on the available options.

6 Add the appropriate options, 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

Use the *Advanced Settings* to configure additional options for the LDAP source handler. Click *Help* for more information on the available options.

7 Click *Next*, then click *Finish*.

6.1.2 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. A Windows version (`ice.exe`) is included in the installation. On Linux, Solaris, and AIX systems, the Import/Export utility is included in the `NOVLice` 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 158
- "LDIF Destination Handler Options" on page 159
- "LDAP Source Handler Options" on page 160
- "LDAP Destination Handler Options" on page 162
- "DELIM Source Handler Options" on page 163
- "DELIM Destination Handler Options" on page 164
- "SCH Source Handler Options" on page 165
- "LOAD Source Handler Options" on page 165

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 log_file`	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, or AIX 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 LDIF_error_log_file`	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.

Option	Description
`-p URL`	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 173 for more information.
`-c URL`	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 173.
`-s URL`	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 173.
`-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 filename`	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 158.
-SLDAP	Specifies that the source is an LDAP server. For a list of supported LDAP options, see “LDAP Source Handler Options” on page 160
-SDELIM	Specifies that the source is a comma-delimited data file. **NOTE**: For better performance, import data by using Novell Import Conversion Export utility with LDIF file instead of DELIM. You can use a custom PERL script to generate the output into your desired format. For a list of supported DELIM options, see “DELIM Source Handler Options” on page 163.
-SSCH	Specifies that the source is a schema file. For a list of supported SCH options, see “SCH Source Handler Options” on page 165
-SLOAD	Specifies that the source is a DirLoad template. For a list of supported LOAD options, see “LOAD Source Handler Options” on page 165.

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 159.
-DLDAP	Specifies that the destination is an LDAP server. For a list of supported options, see “LDAP Destination Handler Options” on page 162.
-DDELIM	Specifies that the destination is a comma-delimited file. For a list of supported options, see “DELIM Destination Handler Options” on page 164.

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 LDIF_file`	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, or AIX 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 value`	Specifies the range of records to be processed.
`-v`	Enables the verbose mode of the handler.
`-e value`	Scheme to be used for decrypting the attribute values present in the LDIF file. [des/3des].
`-E value`	Password for decryption of attributes.

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 LDIF_file`	Specifies the filename where LDIF records can be written. If you omit this option on Linux, Solaris, or AIX systems, the output will go to stdout.
`-B`	Do not suppress printing of binary values.
`-b`	Do not base64 encode LDIF data.
`-e value`	Scheme to be used for encrypting the attribute values received from the LDAP server.[des/3des].
`-E value`	Password for encryption of attributes.

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 server_name`	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.
`-p port`	Specifies the integer port number of the LDAP server specified by *`server_name`*. The default is 389. For secure operations, the default port is 636. When ICE is communicating with an LDAP server on the SSL port (default 636) without a certificate, it chooses to accept any server certificate and assumes it to be a trusted one. This should only be used in controlled environments where encrypted communication between servers and clients is desired but server verification is not necessary.
`-d DN`	Specifies the distinguished name of the entry that should be used when binding to the server-specified bind operation.
`-w password`	Specifies the password attribute of the entry specified by *`DN`*.
`-W`	Prompts for the password of the entry specified by *`DN`*. This option is applicable only for Linux, Solaris, and AIX.
`-F filter`	Specifies an RFC 1558-compliant search filter. If you omit this option, the search filter defaults to `objectclass=*`.
`-n`	Does not actually perform a search, but shows what search would be performed.
`-a attribute_list`	Specifies a comma-separated list of attributes to retrieve as part of the search. In addition to attribute names, there are three other values: ◆ 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.
`-o attribute_list`	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 `-a` 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, `-a* -o telephoneNumber` searches for all user-level attributes and filters the telephone number from the results.
`-R`	Specifies to not automatically follow referrals. The default is to follow referrals with the name and password given with the `-d` and `-w` options.
`-e value`	Specifies which debugging flags should be enabled in the LDAP client SDK. For more information, see "Using LDAP SDK Debugging Flags".

Option	Description
-b *base_DN*	Specifies the base distinguished name for the search request. If this option is omitted, the base DN defaults to " " (empty string).
-c *search_scope*	Specifies the scope of the search request. Valid values are the following: ◆ 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.
-r *deref_aliases*	Specifies the way aliases should be dereferenced during the search operation. Values include the following: ◆ 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. If you omit this option, the alias dereferencing behavior defaults to Never.
-l *time_limit*	Specifies a time limit (in seconds) for the search.
-z *size _limit*	Specifies the maximum number of entries to be returned by the search.
-V *version*	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 *filename*	Specifies a file in DER format containing a server key used for SSL authentication. Filename is optional on Linux, with default value /etc/opt/novell/certs/SSCert.der.
-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	This option is no longer supported. To use SSL, specify a valid certificate using the -L option.
-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".

Option	Description
`-s server_name`	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.
`-p port`	Specifies the integer port number of the LDAP server specified by *server_name*. The default is 389. For secure operations, the default port is 636.
`-d DN`	Specifies the distinguished name of the entry that should be used when binding to the server-specified bind operation.
`-w password`	Specifies the password attribute of the entry specified by *DN*.
`-W`	Prompts for the password of the entry specified by *DN*. This option is applicable only for Linux, Solaris, and AIX.
`-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 181.
`-F`	Allows the creation of forward references. When 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.
`-l`	Stores password values using the simple password method of the Novell Modular Authentication Service (NMAS). 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 value`	Specifies which debugging flags should be enabled in the LDAP client SDK. For more information, see "Using LDAP SDK Debugging Flags".
`-V version`	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 filename`	Specifies a file in DER format containing a server key used for SSL authentication. Filename is optional on Linux with default value `/etc/opt/novell/certs/SSCert.der`.
`-k`	This option is no longer supported. To use SSL, specify a valid certificate using the `-L` option.
`-M`	Enables the Manage DSA IT control.
`-MM`	Enables the Manage DSA IT control, and makes it critical.

Option	Description
-P	Enables concurrent LBURP processing. This option is enabled only if all the operations in the LDIF are add. When you use the -F option, -P is enabled by default.
-Z	Specifies the number of asynchronous requests. This indicates the number of entries the ICE client can send to the LDAP server asynchronously before waiting for any result back from the server.

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 *filename*	Specifies a filename containing comma-delimited records read by the DELIM source handler and sent to the destination handler.
-F *value*	Specifies a file containing the attribute data order for the file specified by -f. By default, the number of columns for an attribute in the delimited file equals maximum number of values for the attribute. If an attribute is repeated, the number of columns equals the number of times the attribute repeats in the template. If this option is not specified, enter this information directly using -t. See "Performing a Comma-Delimited Import" on page 168 for more information.
-t *value*	The comma-delimited list of attributes specifying the attribute data order for the file specified by -f. By default, the number of columns for an attribute in the delimited file equals maximum number of values for the attribute. If an attribute is repeated, the number of columns equals the number of times the attribute repeats in the template. Either this option or -F must be specified. See "Performing a Comma-Delimited Import" on page 168 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 *value*	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 *value*	Specifies the path to append the RDN to (such as o=myCompany). If you are passing the DN, this value is not necessary.
-o *value*	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 *value*	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.

Option	Description
`-d value`	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 value`	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 `-q[t]`.
`-v`	Runs in verbose mode.
`-k value`	Specifies the first line in the delimited file is the template. If this option is used with `-t` or `-F`, the template specified is checked for consistency with that in the delimited file.

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
`-f filename`	Specifies the filename where comma-delimited records can be written.
`-F value`	Specifies a file containing the attribute data order for the file specified by `-f`. By default, the number of columns for an attribute in the delimited file equals maximum number of values for the attribute. If an attribute is repeated, the number of columns equals the number of times the attribute repeats in the template. If this option is not specified, enter this information directly using `-t`.
`-t value`	The comma-delimited list of attributes specifying the attribute data order for the file specified by `-f`. By default, the number of columns for an attribute in the delimited file equals maximum number of values for the attribute. If an attribute is repeated, the number of columns equals the number of times the attribute repeats in the template. Either this option or `-F` must be specified.
`-l value`	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.

Option	Description
`-d value`	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 value`	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 `-q[t]`.
`-n value`	Specifies a naming attribute to be appended during import, for example, `cn`.

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
`-f filename`	Specifies the full path name of the `*.sch` file.
`-c`	(Optional) Prevents the SCH handler from stopping on errors.
`-v`	(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
`-f filename`	Specifies the template file containing all attribute specification and all control information for running the program.
`-c`	Continues to the next record if an error is reported.
`-v`	Runs in verbose mode.
`-r`	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.
`-m`	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)
initial: $R(initial)
sn: $R(last)
dn:cn=$A(givenname,%.1s)$A(initial,%.1s)$A(sn),ou=dev,ou=ds,o=novell
objectclass: inetorgperson
telephonenumber: 1-800-$N(1-999,%03d)-$C(%04d)
title: $R(titles)
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(givenname)
$A(givenname,%s)
$A(givenname,%.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 givenname, initial, and sn. Therefore, these attributes must precede the DN in the settings file.

```
givenname: $R(first)
initial: $R(initial)
sn: $R(last) dn:o=novell,ou=dev,ou=ds,cn=$A(givenname,%.1s)$A(initial,%.1s)$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 Schultzcn cn: Karl Schultzcn cn:
Doug Griegercn 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 168
- "Performing an LDIF Export" on page 168
- "Performing a Comma-Delimited Import" on page 168
- "Performing a Comma-Delimited Export" on page 169
- "Performing a Data Migration between LDAP Servers" on page 170
- "Performing a Schema Import" on page 170
- "Performing a LOAD File Import" on page 170
- "Performing an LDIF Export from LDAP server having encrypted attributes" on page 173
- "Performing an LDIF Import having encrypted attributes" on page 173

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 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 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.

Comma-delimited files generated using Novell Import Conversion Export utility have the template used for generating them in the first line. To specify that first line in the delimited file is the template, use the `-k` option. If `-F` or `-t` is used with `-k`, the template specified should be consistent with that in the delimited file, where both have exactly the same attributes. However, the number of occurrences and the order of appearance of each attribute can differ. In the above example, `in.csv` contains

`dn,cn,title,title,title,sn` in the first line. The following templates are consistent and can be used with -t or -F when -k is used:

`dn,cn,title,sn` (number of repetitions of attribute title differs)

`dn,sn,title,cn` (order of attributes differ)

However, the following are not consistent with the template in `in.csv` and hence cannot be specified with -t or -F when -k is used:

`dn,cn,title,sn,objectclass` (new attribute objectclass)

`dn,cn,title` (missing attribute `sn`)

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 -F
objectClass=* -c sub -D DELIM -f /tmp/server1.csv -F order.csv
```

This command 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.

If any attribute in the `order.csv` has multiple values, `/tmp/server1.csv`, the number of columns for this attribute equals maximum number of values for the attribute. If an attribute repeats in `order.csv`, the number of columns for this attribute equals the number of times the attribute repeats.

For example, if `order.csv` contains `dn,sn,objectclass`, and `objectclass` has 4 values, whereas `dn` and `sn` have only 1 value for all the entries exported, `dn` and `sn` would have 1 column each, whereas `objectclass` would have 4 columns. If you want only 2 values for `objectclass` to be output to the delimited file, `order.csv` should contain `dn,sn,objectclass,objectclass`.

In both cases the attributes are written to the /tmp/server1.csv in the first line. In the first case, /tmp/server1.csv would have dn,sn,objectclass,objectclass,objectclass,objectclasss in the first line of /tmp/server1.csv, and in the second case, it would have dn,sn,objectclass,objectclass.

To prevent the first line to be treated as a sequence of attributes during a subsequent import, use the -k option. See "Performing a Comma-Delimited Import" on page 168 for more information.

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 command 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 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)

initials: $R(initial)

sn: $R(last)

dn: cn=$A(givenname,%.1s)$A(initial,%.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 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 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
-
```

Performing an LDIF Export from LDAP server having encrypted attributes

To perform an LDIF export from LDAP server having encrypted attributes, combine the LDAP source and LDIF destination handlers along with the scheme and password for encryption, for example:

```
ice -S LDAP -s server1.acme.com -p 636 -L cert-server1.der -d cn=admin,c=us -w
password -F objectClass=* -c sub -D LDIF -f server1.ldif -e des -E secret
```

Performing an LDIF Import having encrypted attributes

To perform an LDIF import of a file having attributes encrypted by ICE previously, combine the LDIF source with the scheme and password used previously for exporting the file and LDAP destination handlers, for example:

```
ice -S LDIF -f server1.ldif -e des -E secret -D LDAP -s server2.acme.com -p 636 -L
cert-server2.der -d cn=admin,c=us -w password
```

6.1.3 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	Changes the placement of an entry. 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. For information on the format of these rules, see “Placement Rules” on page 178.
Creation	Supplies missing information that might be needed to allow an entry to be created successfully on import. 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. For information on the format of these rules, see “Create Rules” on page 176.

Rule	Description
Schema Mapping	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 ◆ 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. For information on the format of these rules, see "Schema Mapping Rules" on page 175.

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 175.

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 156.

Option	Description
`-p` *URL*	Location of an XML placement rule to be used by the engine.
`-c` *URL*	Location of an XML creation rule to be used by the engine.
`-s` *URL*	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 Novell Identity Manager. For more information on Novell Identity Manager, see the *Novell Identity Manager 4.0.1 Administration Guide* (http://www.novell.com/documentation/idm401/).

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="inetOrgPperson">
      <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 `sr1.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, `outt1.ldf`.

```
ice -o -sfile://sr1.xml -SLDIF -f1entry.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 `1entry.ldf` file and to send the results to a destination file, `outt1.ldf`.

```
ice -o -cfile://cr1.xml -SLDIF -f1entry.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 `scr-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
```

6.1.4 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 162.

6.1.5 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 182
- "Using LBURP" on page 183
- "Configuring the Database Cache" on page 183
- "Using Simple Passwords" on page 183
- "Using Indexes Appropriately" on page 184

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 135.

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 181.

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 18, "Maintaining Novell eDirectory," on page 481.

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 onwards, you can choose to store passwords using the simple password feature of Novell Modular Authentication Service (NMAS). 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 Services 3.3 Administration Guide* (http://www.novell.com/documentation/nmas33/admin/data/a20gkue.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 Section 6.2, "Index Manager," on page 184.

6.2 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 Section 6.3, "Predicate Data," on page 188.

6.2.1 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.

6.2.2 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 Section 6.3, "Predicate Data," on page 188 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.

6.2.3 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. The index 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*.

6.2.4 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*.

6.2.5 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	Specifies the state of the index. When defining an index, this field should be set to 2 (online). eDirectory supports the following values: ◆ 0 - Online, which indicates that the index is up and working. ◆ 1 - Suspended, which indicates that the index is not used in queries and is not updated. ◆ Bringing Online, which indicates that the index is in the process of being created. It has two states, Bringing Online (low) and Bringing online (high). ◆ 2 - Bringing Online (low) indicates that the index creation process on the said attribute is pending. ◆ 3 - Bringing Online (high) indicates that the index creation is in progress. ◆ 4 - Creation, which indicates that the index has been defined and is waiting for the background process to run. The background process changes the state after the building begins.
4	Index rule	Specifies the type of matching: ◆ 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	Specifies who created the index. When defining an index, you must set this value to 0. eDirectory supports the following values: ◆ 0 - User Defined ◆ 1 - Added on Attribute Creation ◆ 2 - Required for Operation ◆ 3 - System Index
6	Index value state	Specifies the source of the index. When defining an index, set this string to 1. eDirectory supports the following values: ◆ 0 - Uninitialized ◆ 1 - Added from Server ◆ 2 - Added from Local DIB ◆ 3 - Deleted from Local DIB ◆ 4 - Modified from Local DIB
7	Attribute name	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.

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
```

6.3 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.

6.3.1 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.

6.4 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 Client Service Manager eMTool" on page 189
- "Using the Service Manager Plug-In to Novell iManager" on page 190

6.4.1 Using the 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 Client, see Section 21.1, "Using the Command Line Client," on page 516.

To use the Client Service Manager eMTool:

1. Run the Client in interactive mode by entering the following at the command line:

   ```
   java -cp path_to_the_file/emboxclient.jar  -i
   ```

   (If you have already put the `emboxclient.jar` file in your class path, you only need to enter `java -i`.)

The Client prompt appears:

```
Client>
```

2 Log in to the server that will run Service Manager by entering the following:

```
login -s server_name_or_IP_address -p port_number
-u username.context -w password -n
```

The port number is usually 80 or 8028, unless you have a Web server that is already using the port. The `-n` option opens a nonsecure connection.

The Client indicates whether the login is successful.

3 Enter one of the following Service Manager commands:

Command	Description
`service.serviceList`	Lists the available eDirectory services.
`service.serviceStart -n Module_name`	Starts the specified eDirectory service.
`service.serviceStop -n Module_name`	Stops the specified eDirectory service.
`service.serviceInfo -n Module_name`	Displays information for the specified service.

You can also use the `list -t service` command in the Client to list the Service Manager options with details. See "Listing eMTools and Their Services" on page 519 for more information.

4 Log out from the Client by entering the following command:

```
logout
```

5 Exit the Client by entering the following command:

```
exit
```

6.4.2 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.
	Starts a service.
	Stops a service.
	A service is running but you can't stop it.

7 Offline Bulkload Utility

ldif2dib utility lets you bulkload data from LDIF files to the Novell eDirectory database (DIB), when the eDirectory server is offline. It uses the existing directory and does not create a new database while importing entries from an LDIF file to the DIB.

This chapter includes the following information:

- Section 7.1, "Using ldif2dib for Bulkloading," on page 193
- Section 7.2, "Multiple Instances," on page 195
- Section 7.3, "Tuning ldif2dib," on page 195
- Section 7.4, "Limitations," on page 196
- Section 7.5, "Caveats," on page 198

7.1 Using ldif2dib for Bulkloading

You can specify the LDIF file containing the data to be imported and the path to the database files where data needs to be imported through the command line interface. Using ldif2dib to bulkload data requires the following steps:

1. Take a backup of the DIB.

   For more information on the backup and restore process, refer to in the *Novell eDirectory 8.8 SP7 Administration Guide.*

2. Stop the eDirectory server.
3. To start bulkloading from the LDIF file, enter the following at the command prompt:

   ```
   ldif2dib <LDIF File Name> [Options]
   ```

   Where

   - **LDIF File Name:** Specifies the name of LDIF file to bulkload.
   - **Options:** These are optional and specify the different parameters that you can use for tuning this utility. The options supported by the ldif2dib utility are listed below:

Options	Description	Value
`-b`	Specifies batch mode operation.	Default: 0
`-c`	Specifies the cache size in bytes.	Default: 0
`-e`	Populates errors into the specified log file.	Default: `ldif2dib.log`
`-p`	Specifies the block cache percentage.	Range: 0–100. Default: 50
`-i`	Specifies the check point interval in seconds.	

`-n`	Specifies the database name to import entries. Enables bulkloading the object entries from LDIF file to any instance of eDirectory (DIB). if `-n` is not used, the utility displays a list of all eDirectory instances configured on the system. If only a single instance is configured, the utility selects the location of the `nds.db` file for that instance.	Default: `nds.db` If you are not using the default database, you need to specify the complete path to the DIB.
`-dr`	Specifies the directory where the roll forward log (rfl) files are located.	Default: eDirectory database
`-dd`	Specifies the directory where the data files are located.	Default: eDirectory database
`-t`	Specifies the transaction size, that is, objects per transaction.	Default: 100 objects
`-md`	Specifies the maximum dirty cache in bytes.	Default: 0
`-ld`	Specifies the low dirty cache in bytes.	Default: 0
`-r`	Populates the change cache, if there is more than one replica for the partition into which the objects are being loaded. If this option is not provided, the change cache is generated when the server is brought up on the bulkloaded DIB.	
`-a`	Specifies the number of entries that should be loaded.	Entire LDIF file
`-u`	Checks for duplicate entries either in the LDIF file or the DIB.	
`-k`	Specifies the number of entries that should be skipped from the LDIF file.	Default: 0 objects
`-s`	Specifies skipping errors and continuing.	
`-w`	Generates a RSA key-pair (NDS password) that is to be imported for the userPassword attribute.	
`-v`	Verbose mode to log the DNs of the entries processed into the log file.	
`-x`	Disables indexes before loading entries using ldif2dib. At the end of the bulkload, the indexes are re-enabled.	
`-?`	Displays the help messages. You can use this option anywhere in the command line with or without the hyphen (-).	

For example, if you want to set the options for specifying batch mode, cache size and block cache percentage options, enter the following command:

```
ldif2dib 1MillionUsers.ldif -b/novell/log/logfile.txt -c314572800 -p90
```

TIP: You can temporarily suspend the bulkload by pressing the s/S key. The Escape key (Esc) can be used to stop the bulkload.

7.2 Multiple Instances

ldif2dib can be used to bulkload entries from LDIF files to a particular instance of eDirectory (DIB) by specifying the location of its `nds.db` file with the `-n` option. If the location of the `nds.db` file is not specified with the `-n` option and if there is a single instance of eDirectory configured on the system, ldif2dib automatically detects the location of its database files. However, if there are multiple instances, ldif2dib displays a menu listing all configured instances and allows you to choose an instance for bulkload.

For more information on the multiple instances of eDirectory, see "Multiple Instances" (http://www.novell.com/documentation/edir88/edir88new/data/bqebx8t.html) section in the *Novell eDirectory 8.8 SP7 What's New Guide*.

7.3 Tuning ldif2dib

This section contains information about the parameters that can be used to tune ldif2dib.

- Section 7.3.1, "Tuning the Cache," on page 195
- Section 7.3.2, "Transaction Size," on page 196
- Section 7.3.3, "Index," on page 196
- Section 7.3.4, "Block Cache Percent," on page 196
- Section 7.3.5, "Check Point Interval," on page 196

7.3.1 Tuning the Cache

The database cache setting is one of the more significant settings that affects the eDirectory performance. If it is set too low, eDirectory operations slow down because information must be retrieved from the disk more often. If it is set too high, enough memory is not available for other processes to run and the whole system slows down. For more information on cache, see "Configuring the FLAIM Subsystem" (http://www.novell.com/documentation/edir88/edir88tuning/data/bqmivb8.html) in the *Novell eDirectory 8.8 SP7 Tuning Guide for UNIX Platforms.*

Bulkload performance generally increases on increasing the cache size. However, no performance improvement has been observed by increasing the cache size beyond a value which is 3.8 times the size of the LDIF file.

7.3.2 Transaction Size

The transaction size defines the chunk size in terms of number of objects per transaction. When the transaction size is high, a small number of large chunk writes result and when it is low, a large number of small chunk writes result.

The bulkload performance increases with higher transaction sizes. A transaction size of zero results in a special case which allows unlimited objects per transaction. When the transaction size is zero, the performance is high because the commit is done at the end of the bulkload. However, we do not recommend you to set the transaction size to 0 for very large LDIF files (larger than one million objects). You can set the transaction size as high as 4000 for very large LDIF files.

7.3.3 Index

Although use of indexes leads to a higher search performance, it makes bulkload slower because indexes need to be updated for every object loaded to the DIB. This is especially true for substring indexes. Therefore when you are bulkloading large number of objects, you can suspend indexes to speed up the bulkload. The indexes are automatically resumed when eDirectory server is brought up. Use the `-x` option to disable indexes before loading entries using ldif2dib.

7.3.4 Block Cache Percent

If the sub-string indexes are enabled for attributes, it is recommended to set the block cache percent to 50%, and if the sub-string indexes are disabled for attributes, you can set the block cache percent to 90%.

7.3.5 Check Point Interval

Checkpoint interval is the time for which the database waits before it initiates the checkpoint background thread which brings the on-disk version of the database up to the same coherent state as the in-memory (cached) database. This check point thread flushes the dirty cache to the disk, followed by cleaning up the roll forward log. Since bulkload is temporarily suspended while check point thread runs, we recommend that you set the check point interval to a high value to achieve faster bulkloads.

7.4 Limitations

This section contains limitations of the ldif2dib utility:

7.4.1 Schema

- The LDIF file should mention all the object classes that an entry belongs to. An entry can belong to multiple object classes because of inheritance. For example, an entry of type inetOrgPerson should have following syntax in the LDIF file:

```
objectclass: inetorgperson
objectclass: organizationalPerson
objectclass: person
objectclass: top
```

- Currently, following syntaxes are not supported:

SYN_UNKNOWN	SYN_NET_ADDRESS
SYN_OCTET_LIST	SYN_PATH
SYN_REPLICA_POINTER	SYN_TIMESTAMP
SYN_BACK_LINK	SYN_TYPED_NAME
SYN_HOLD	SYN_TIME

7.4.2 ACL Templates

ACLs that are specified in the ACL templates for an object class, are not automatically added for objects bulkloaded using ldif2dib.

7.4.3 Options

On Linux, if the `-b` option is used, the screen that displays statistics disappears after the bulkload is complete. The final statistics, however, are written to the log file for reference.

7.4.4 Simple Password LDIF

On Windows, while uploading LDIF having simple password, ldif2dib might fail if the NICI keys in `system` and `Administrator` folder are not in sync. To work around this issue, access the keys present in the `nici/system` folder as follows:

1. Go to the `C:\Windows\system32\novell\nici\`folder.
2. Backup the files present in the `Administrator` folder.
3. Get access to the `system` folder and its files by following the below mentioned steps:
   - **3a** Go to the *Security* tab in the Properties window of the `system` folder.
   - **3b** Select *Advanced Options* and go to *Owner* tab.
   - **3c** Select *Administrator*.
   - **3d** Go back to the *Security* tab and add *Administrator* to the list.

     Repeat the similar steps to get read access to all the files present inside the `system` folder.
4. Overwrite the files in the `Administrator` folder with the ones in the `system` folder.
5. Once the upload is done, copy the backed up files to the `Administrator` folder.
6. Revert back the Administrator's access to the `system` folder and also the files within the folder.

7.4.5 Custom Classes

Bulkloading an LDIF with a large number of container objects using ldif2dib can result in a memory build up leading to a -150 error being reported.

7.5 Caveats

Behavior of ldif2dib is undefined in the following scenarios:

- Section 7.5.1, “Duplicate Entries,” on page 198
- Section 7.5.2, “No Schema Checks,” on page 198
- Section 7.5.3, “Insufficient Space on Hard-Drive,” on page 198
- Section 7.5.4, “Forced Termination,” on page 198
- Section 7.5.5, “Terminal Resizing,” on page 198

7.5.1 Duplicate Entries

Uploading LDIF files having duplicate entries or having entries already present in the DIB, without the `-u` option would cause the entry to be added more than once, leading to an inconsistent state of the DIB. So if you are not sure if entries are repeated in the LDIF or if they are present in DIB before the bulkload, use the `-u` option during bulkload.

7.5.2 No Schema Checks

ldif2dib does not perform any schema checks. As a result, you can add an attribute to an object even if the attribute does not belong to the schema of the object. This would leave the DIB in an inconsistent state. Use ldif2dib only when you are sure that the LDIF data does not need schema checks.

7.5.3 Insufficient Space on Hard-Drive

Behavior of ldif2dib is undefined when there is not enough space on the hard-drive for all the objects being loaded. You need to make sure that there is sufficient space for all the objects before starting the bulkload.

7.5.4 Forced Termination

Forcefully terminating the ldif2dib process can leave the DIB in an inconsistent state. Use the Escape key to gracefully exit the bulkload.

7.5.5 Terminal Resizing

Resizing the terminal during bulkload can distort the statistics displayed on the user interface. Terminal resizing should be avoided while bulkload is in progress.

8 Using Novell iMonitor

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 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
- Novell Identity Manager 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 section gives information on the following topics:

- Section 8.1, “System Requirements,” on page 200
- Section 8.2, “Accessing iMonitor,” on page 201
- Section 8.3, “iMonitor Architecture,” on page 201
- Section 8.4, “iMonitor Features,” on page 206
- Section 8.5, “Ensuring Secure iMonitor Operations,” on page 223
- Section 8.6, “Configuring HTTP Server Object,” on page 224
- Section 8.7, “Setting HTTP Stack Parameters Using ndsconfig,” on page 225

8.1 System Requirements

To use iMonitor you need

- Novell eDirectory 8.7.1 or later
- A supported Web browser, including Microsoft Internet Explorer or Firefox

8.1.1 Platforms

The iMonitor utility runs on the following platforms:

- Windows 2000 and 2003 Server (No SSL)
- Linux
- Solaris
- AIX

For Windows, iMonitor loads automatically when eDirectory runs. On Linux, Solaris, and AIX, iMonitor can be loaded using the `ndsimonitor -l` command. It can also be loaded automatically by adding `[ndsimonitor]` in the `/etc/opt/novell/eDirectory/conf/ndsimon.conf` file before starting the eDirectory Server.

The iMonitor utility runs on the following Web browsers:

- Microsoft IE 6
- Microsoft IE 7
- Microsoft IE 8
- Firefox* 1.5.x, 2.x, or 3.x

8.1.2 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 Windows
- All versions of NDS and eDirectory for Linux, Solaris, and AIX

8.2 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://137.65.135.150:8028/nds`

   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 eDirectory HTTPS stack is available, you can use iMonitor through HTTPS.
4. Specify a user name, context, and password. For example, `login cn=admin.o=novell`

   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*.

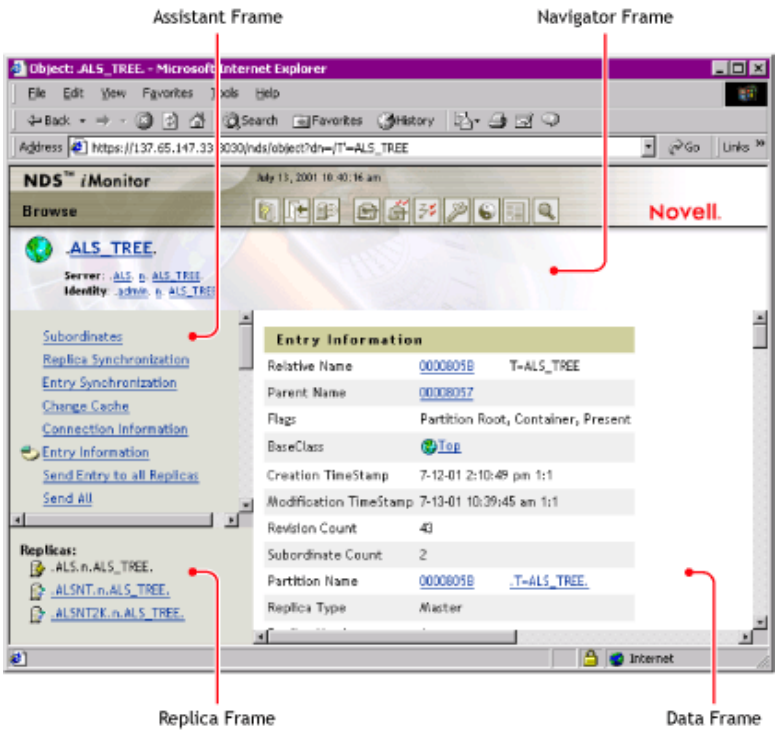
8.3 iMonitor Architecture

- "Anatomy of an iMonitor Page" on page 201
- "Modes of Operation" on page 202
- "iMonitor Features Available on Every Page" on page 204
- "Configuration Files" on page 204

8.3.1 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 8-1 *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.

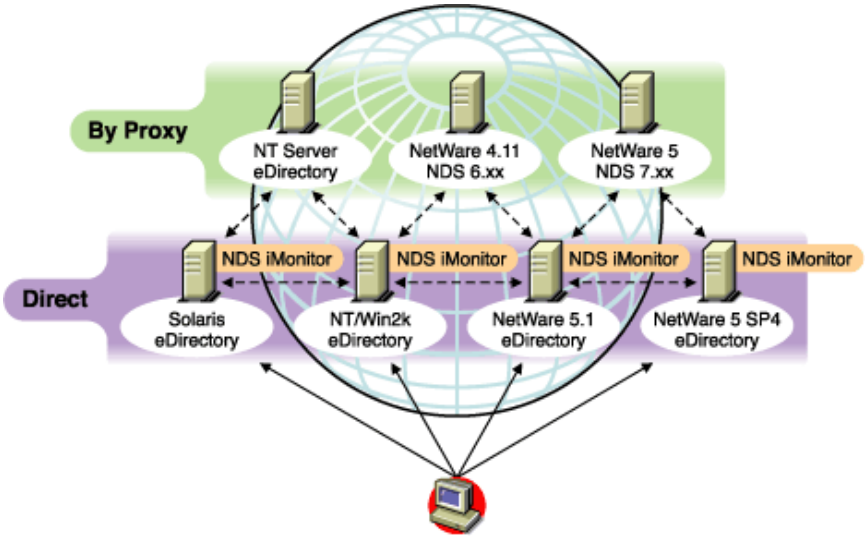
Data Frame: Shows the detailed information about your servers that you request by clicking one of the links listed above. This is the only page you will see if your Web browser does not support frames.

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.

8.3.2 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 8-2 *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 and 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, DSRepair, 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

8.3.3 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.

8.3.4 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 Windows, and in the `/etc` directory on Linux, Solaris, and AIX.

- "ndsimon" on page 204
- "ndsimonhealth" on page 205

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
Windows	*install directory*\novell\NDS\ndsimon.ini
Linux, Solaris, and AIX	/etc/opt/novell/eDirectory/conf/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

  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 8028. 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.

- 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
Windows	*install directory*\novell\NDS\ndsimonhealth.ini
Linux, Solaris, and AIX	/etc/opt/novell/eDirectory/conf/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.

8.4 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 207
- "Viewing Partition Synchronization Status" on page 207
- "Viewing Server Connection Information" on page 208
- "Viewing Known Servers" on page 209
- "Viewing Replica Information" on page 209
- "Controlling and Configuring the DS Agent" on page 209
- "Configuring Trace Settings" on page 210
- "Viewing Process Status Information" on page 211
- "Viewing Agent Activity" on page 211
- "Viewing Traffic Patterns" on page 212
- "Viewing Background Processes" on page 212

- "Viewing eDirectory Server Errors" on page 213
- "Viewing DSRepair Information" on page 213
- "Viewing Agent Health Information" on page 213
- "Browsing Objects in Your Tree" on page 214
- "Viewing Entries for Synchronization or Purging" on page 214
- "Viewing the Synchronization Status of a Replica" on page 215
- "Configuring and Viewing Reports" on page 215
- "Viewing Schema, Class, and Attribute Definitions" on page 217
- "Searching for Objects" on page 218
- "Using the Stream Viewer" on page 218
- "Clone DIB Set" on page 219

8.4.1 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.

8.4.2 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.

8.4.3 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).

8.4.4 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.

8.4.5 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.

8.4.6 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*) was the first eDirectory version to implement a feature known as Filtered Replicas. See "Filtered Replicas" on page 59 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.

8.4.7 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 user name 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.

8.4.8 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.

In iMonitor, click *Agent Process Status* in the Assistant frame. Background process statuses that are currently reported include the following:

- Schema synchronization
- Obituary processing
- External reference/DRL
- Limber
- Repair

8.4.9 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

eDirectory 8.6 or later version 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.

8.4.10 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.

8.4.11 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 or later versions, 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.

8.4.12 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.

8.4.13 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.

8.4.14 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.

8.4.15 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.

8.4.16 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 eDirectory server to view this page.

Entry Synchronization lets you determine why an entry needs to be synchronized.

NOTE: iMonitor only lists a limited number of objects in the Change Cache page. If you want to view all objects in the change cache, either for a specific partition or for all partitions on a server, you can run a Change Cache Dump Report in the Reports page. See "Configuring and Viewing Reports" on page 215 for more information about configuring and running reports in iMonitor.

8.4.17 Viewing Novell Identity Manager 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.

8.4.18 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.

8.4.19 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, using the [Public] identity. Any reports you run directly are run as your identity. All report data is stored on the server from which you run the report. iMonitor stores report data in the following directories by default, depending on the operating system:

Platform	Directory
Windows	`C:\Novell\NDS\ndsimon\dsreports\`
Linux, Solaris, and AIX	`/var/opt/novell/eDirectory/data/dsreports`

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.

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.

Report	Description
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.
Change Cache Dump	Lists all the objects in the change cache for the selected partition or for all partitions on the server. This report also generates an XML dump of the objects in the change cache, along with attributes and values that need to be synchronized across servers. The report provides information for analyzing all objects in the change cache. **NOTE** ◆ In order to run a Change Cache Dump Report, you must have eDirectory 8.8 SP7 or later installed. ◆ iMonitor stores change cache dumps in the same directory as the actual Change Cache Dump Report, as listed in the previous table.
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.
Value Count	Generates a list of objects with attribute, which have value count more than a value you specify.

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 either periodically 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. In the *Runable Report* list, click *Custom Reports*.
3. Enter a name for the report, then enter the URL for the iMonitor page you want to launch as a report.

   When running a custom report, enter the URL as follows:

```
/nds/required page
```

4. In the *Saved reports* field, specify the number of versions of the report you want to keep or retain.
5. (Optional) Click *Save* to save the report.
6. (Optional) Configure the report to run either periodically 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.

8.4.20 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 tee 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.

8.4.21 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.

     NOTE: The *Display Options* settings are only available if you click *Advanced* to view all Advanced Search options.
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.

8.4.22 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 Windows servers) or `streams.conf` (for Solaris and Linux servers), so you can also manually edit the default settings.

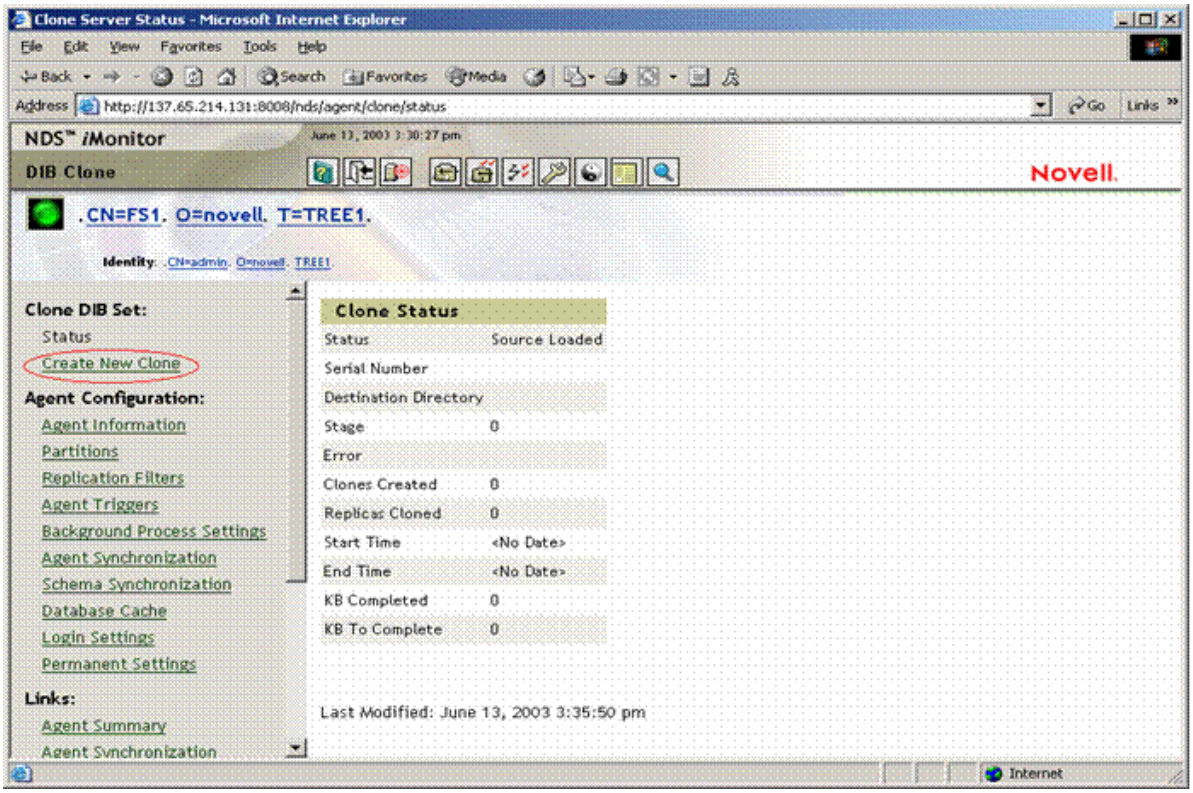
8.4.23 Clone DIB Set

This option creates a complete DIB fileset duplicate of an eDirectory database stored on a single server (the source server). The DIB Clone must be taken from the source server that holds all the master replicas in the tree. 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 Windows server.

Although the back end for this feature was shipped with eDirectory 8.7, it was not supported until eDirectory 8.7.1 running iMonitor 2.4 or later. This option does not apply to any version of Novell eDirectory or NDS prior to 8.7.

Figure 8-3 *Clone DIB Set Page in iMonitor*



This section includes the following information:

- "Clone DIB Set Use Cases" on page 220
- "Creating a Clone" on page 220

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
◆ Only need one copy of the partition to succeed. ◆ Less down time on large servers with multiple partitions.	◆ 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
◆ Quicker up time, especially on large scale databases.	◆ 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 up and not locked.

- “Online Method” on page 220
- “Offline Method” on page 221

Online Method

1 Load the dsclone module on the source server.

Platform	To Extend the Schema
Windows	In `NDSCons.exe`, select *dsclone.dll*, then click *Start*.
Linux, Solaris, and AIX	Add an `ndsclone` entry to the `ndsmodules.conf` file, then use the `http://IP address:port/dhost` page to load the Directory Clone Agent.

2 Disable the inbound sync from iMonitor agent configuration page before starting the clone DIB process on the source server.

3 Create the clone DIB fileset.

3a Run Clone DIB Configuration in iMonitor.

Click *Agent Configuration* > *Clone DIB Set* > *Create New Clone*.

3b 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.

3c Click *Submit*.

The NDS Clone object is created and the DIB fileset is copied to the specified destination.

4 Move the cloned DIB fileset onto the target server's DIB directory.

Additionally, on Linux, Solaris, and AIX systems, transfer the `/etc/opt/novell/eDirectory/conf/nds.conf` file to the target server and update all the references to the source server in the file with the target server name.

5 Install eDirectory and start the server on the target server. Bring down the target server and remove the old DIB fileset. Copy the DIB directory containing the cloned DIB fileset to the target server and bring it up.

Ensure that 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.

6 Make sure that the replica attribute value of the target server is synched with all the servers. Once the attribute changes are available on all servers, reenable the inbound sync on the source server. The inbound sync can be enabled either through the iMonitor agent configuration page or through DSTrace.

7 To complete the eDirectory configuration, see "Completing the eDirectory Configuration" on page 222.

Offline Method

1 Create the clone DIB fileset.

1a Run Clone DIB Configuration in iMonitor.

Click Agent Configuration > *Clone DIB Set* > *Create New Clone*.

1b 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.

1c Click *Submit*.

The NDS clone object is created, the DIB is locked in the source server, and an error reports that eDirectory is locked.

1d Manually copy the `*.nds`, `nds*`, and `nds.rfl/*.*` files from the source server's DIB directory to a destination or media on the target server convenient for moving the set to the target server's DIB directory. Additionally, on Linux, Solaris, and AIX systems, transfer the `/etc/opt/novell/eDirectory/conf/nds.conf` file to the target server and update all the references to the source server in the file with the target server name.

1e Export NDSD_DISABLE_INBOUND=Y environment variable, then start `ndsd` to disable the inbound sync on the source server.

1f Restart eDirectory on the source server.

If eDirectory is restarted on the source server before the files are copied, this clone is invalid. The new NCP Server object must then be deleted and the clone must be recreated.

2 Move the cloned DIB fileset onto the target server's DIB directory.

3 Make sure that the replica attribute value of the target server is synched with all the servers. Once the attribute changes are available on all servers, reenable the inbound sync on the source server. The inbound sync can be enabled either through the iMonitor agent configuration page or through DSTrace.

4 Install eDirectory and start the server on the target server, with the DIB directory containing the cloned DIB fileset.

Ensure that 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.

5 To complete the eDirectory configuration, see "Completing the eDirectory Configuration" on page 222.

Completing the eDirectory Configuration

- "SDIKEY" on page 222
- "Configuring SAS, LDAP, and SNMP Services" on page 223

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
Windows	`C:\WINDOWS\system32\novell\nici\nicisdi.key`
Linux, Solaris, and AIX	`/var/opt/novell/nici/0/nicisdi.key`

3 Start eDirectory on the target server.

Configuring SAS, LDAP, and SNMP Services

All the services listed below can be configured in one operation by entering the following command at the command line:

```
ndsconfig upgrade [-a admin FDN]
```

IMPORTANT: The above command is applicable only to Linux, Solaris, and AIX.

For configuring the services individually, refer the following tables:

SAS

Platform	Command or Tool
Windows	Create SAS Service object and Certificates by using iManager.

LDAP

Platform	Command or Tool
Windows	Create LDAP Server and Group Objects by using iManager.

SNMP

Platform	Command or Tool
Windows	rundll32 snmpinst, snmpinst *-c createobj -a userFDN -p password -h hostname_or_IP_address*

8.5 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.

8.6 Configuring HTTP Server Object

An eDirectory installation creates an HTTP server object. The default configuration for HTTP Services is located in the directory on this object. However, you can modify the default configuration by using either ConsoleOne or using the Novell iManager. The HTTP server object represents server-specific configuration data.

The following are the attributes on the HTTP server object:

- **httpDefaultTLSPort:** Indicates the secure port at which HTTP the server listens.
- **httpDefaultClearPort:** Indicates the clear text port at which HTTP the server listens.
- **httpAuthRequiresTLS:** Indicates whether the request coming through the clear text port need to be redirected to a secure port.
- **httpTraceLevel:** Indicates the debug level of HTTP server in DSTrace.
- **httpKeyMaterialObject:** Holds the DN of the certificate object which the HTTP server needs to use when handling the secure connection.
- **httpSessionTimeout:** Indicates the timeout of the HTTP sessions. The default value is 900 seconds.

- **httpKeepAliveRequestTimeout:** Indicates the keep alive timeout of each HTTP request. The default value is 15 seconds.
- **httpRequestTimeout:** Indicates the timeout of each HTTP request. The default value is 300 seconds.
- **httpIOBufferSize:** Indicates the input and output buffer size of the HTTP server. The default value is 8192 bytes.
- **httpThreadsPerCPU:** Indicates the HTTP threads that has to be spawned per CPU. The default value is 2 threads.
- **httpHostServerDN:** Holds the DN of the NCP server object to which it is associated with.
- **httpBindRestrictions:** Used to set the cipher encryption level. The four values that can be used to restrict the cipher usage are:
  - 0 - accept HIGH, MEDIUM, LOW and EXPORT ciphers
  - 1 - accept HIGH, MEDIUM, and LOW ciphers only
  - 2 - accept HIGH and MEDIUM ciphers only
  - 3 - accept HIGH ciphers only

  The default value is 2.

8.7 Setting HTTP Stack Parameters Using ndsconfig

The following are the HTTP stack parameters using ndsconfig:

- **http.server.interfaces:** Holds the clear text interface at which the HTTP server listens. This is set during a new instance configuration by ndsconfig.
- **http.server.request-io-buffer-size:** Indicates the input and output buffer size of the HTTP server. The detault value is 8192 bytes.
- **http.server.request_timeout-seconds:** Indicates the timeout of each HTTP request. The default value is 300 seconds.
- **http.server.keep-timeout-seconds:** Indicates the keep alive timeout of each HTTP request. The default value is 15 seconds.
- **http.server.threads-per-processor:** Indicates the HTTP threads that has to be spawned per CPU. The default value is 2 threads.
- **http.server.session-exp-seconds:** Indicates the time out of the HTTP sessions. The default value is 900 seconds.
- **http.server.trace-level:** Indicates the debugging level of HTTP stack in DSTrace. The default level is 2.
- **http.server.clear-port:** Indicates the clear text port at which HTTP server listens.
- **http.server.tls-port:** Indicates the secure port at which the HTTP server listens.
- **http.server.auth-req-tls:** Indicates whether the requests coming through clear text port need to be redirected to secure port.
- **https.server.interfaces:** Holds the secure interface at which the HTTP server listens. This is set during new instance configuration by ndsconfig.
- **https.server.cached-cert-dn:** Holds the DN of the certificate object, which the HTTP server needs to use while handling the secure connection.

9 SecretStore Configuration for eDirectory Server

SecretStore executables and libraries are installed by default with eDirectory installation. With eDirectory 8.8 SP4 onwards, for new installation of the eDirectory servers, SecretStore configuration is made optional. For eDirectory server upgrade, no changes are made to the existing configuration. Ensure you extend the eDirectory schema for SecretStore functionality on Linux, UNIX, and Windows platforms using the following command:

```
ice -S SCH -f /var/opt/novell/eDirectory/lib/nds-schema/sssv3.sch -D LDAP -s
<serverIP> -d <adminDN>
```

For example,`ice -S SCH -f /var/opt/novell/eDirectory/lib/nds-schema/sssv3.sch -D LDAP -s 1.2.3.4 -d cn=admin,o=administrators`

Use the procedures given in the following sections to configure and deconfigure SecretStore:

- Section 9.1, “UNIX,” on page 227
- Section 9.2, “Windows,” on page 227

9.1 UNIX

Configuring SecretStore

Use the following steps to configure the SecretStore:

1. To configure, run `ssscfg -c`.
2. Add an entry `ssncp` in the `/etc/opt/novell/eDirectory/conf/ndsmodules.conf` to load SecretStore module by default while eDirectory is being started. You can also use `nss` utility to load or unload the SecretStore module later.

Deconfiguring SecretStore

For deconfiguration, run the `ssscfg -d` command. Remove the `ssncp` entry if it exists in the `/etc/opt/novell/eDirectory/conf/ndsmodules.conf` location.

9.2 Windows

Use the following steps to configure and deconfigure the SecretStore:

1. For configuration, run `ssscfg.exe -c`.
2. For deconfiguration, run `ssscfg.exe -d`.

The `ssscfg.exe` utility exists in the `eDirectoryInstallDrive:>\Novell\NDS\` directory. To autoload the SecretStore module during eDirectory server startup, set the `ssncp.dlm` module to `auto` from the GUI interface of the `NDSCons.exe`.

10 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 270 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:

- Section 10.1, "Merging eDirectory Trees," on page 229
- Section 10.2, "Grafting a Single Server Tree," on page 235
- Section 10.3, "Renaming a Tree," on page 240
- Section 10.4, "Using the Client to Merge Trees," on page 241

10.1 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 230
- "Target Tree Requirements" on page 230
- "Schema Requirements" on page 230
- "Merging the Source into the Target Tree" on page 231
- "Partition Changes" on page 231
- "Preparing the Source and Target Trees" on page 232
- "Synchronizing Time before the Merge" on page 233
- "Merging Two Trees" on page 233
- "Post-Merge Tasks" on page 234

10.1.1 Prerequisites

- ❒ Novell eDirectory 8.8 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.

NOTE: To delete Authorized Login Methods, use the ldapdelete tool or ConsoleOne.

10.1.2 Target Tree Requirements

- ❒ Novell eDirectory 8.8 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.

10.1.3 Schema Requirements

Before attempting to perform a merge operation, the schema of both trees must match exactly. You should run DSRepair 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.

10.1.4 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.

10.1.5 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 10-1 and Figure 10-2 illustrate the effect on partitions when you merge two trees.

Figure 10-1 *eDirectory Trees before a Merge*

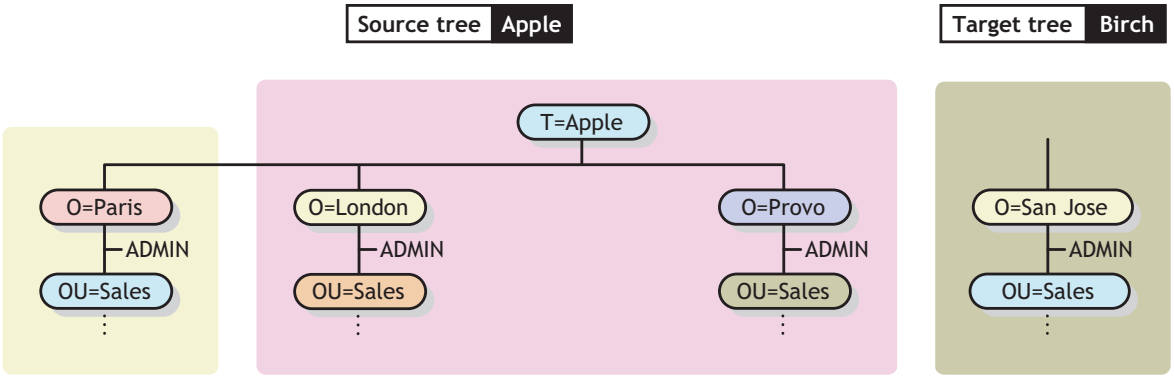
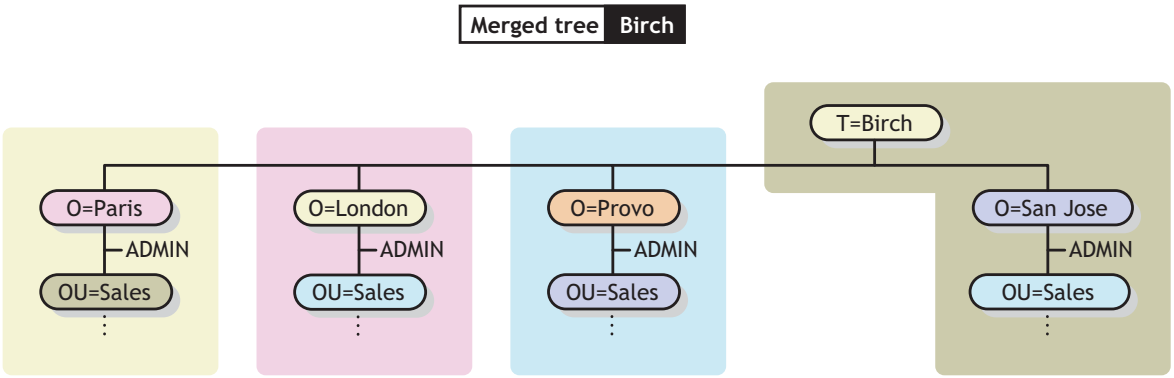


Figure 10-2 *Merged eDirectory Tree*



10.1.6 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 93. 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 8.8 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 280 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 541.

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 16, "Backing Up and Restoring Novell eDirectory," on page 391.

10.1.7 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 "Time Services" in the *OES Planning and Implementation Guide* (http://www.novell.com/documentation/oes11/oes_implement_lx/data/time.html).

10.1.8 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 command line client

  For more information, see Section 10.4, "Using the Client to Merge Trees," on page 241.

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, 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 Section 10.3, "Renaming a Tree," on page 240.

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 232

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 user name and password for the source tree.
6. Specify the target tree name and the Administrator user name 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.

10.1.9 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 Section 5.2, “Merging a Partition,” on page 136.

3 Re-create any leaf objects or aliases in the tree that were deleted before you ran DSMerge.

4 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.

5 Update your client workstation configuration.

   For the Novell Client for Windows, 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.

10.2 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 10-3 and Figure 10-4 illustrate the effects of grafting a tree into a specific container.

Figure 10-3 *eDirectory Trees before a Graft*

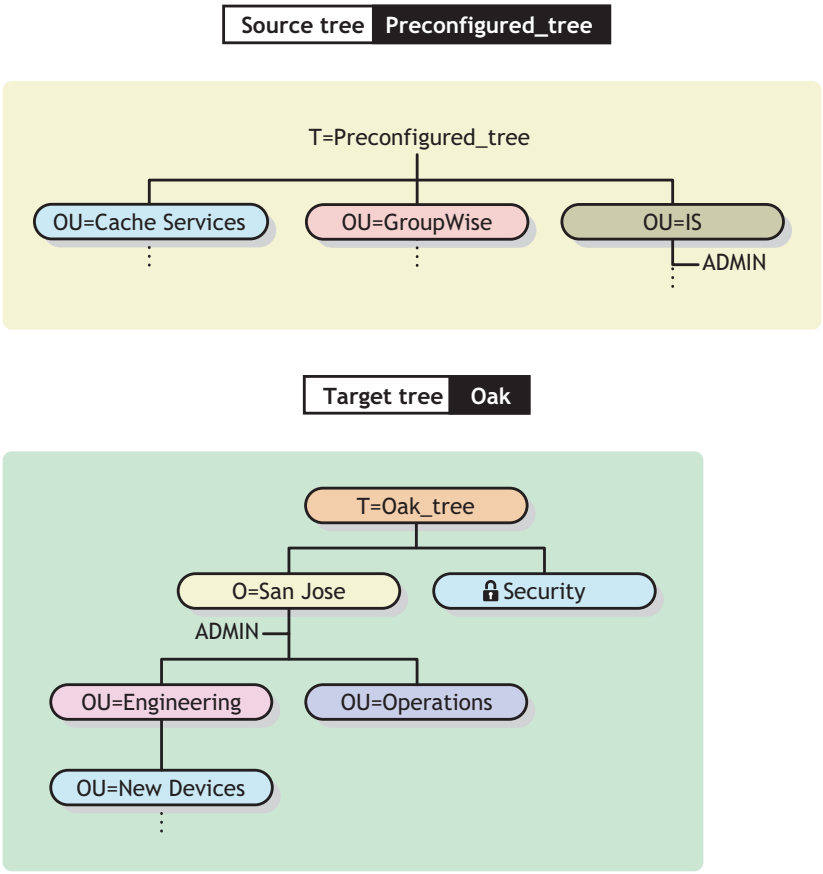
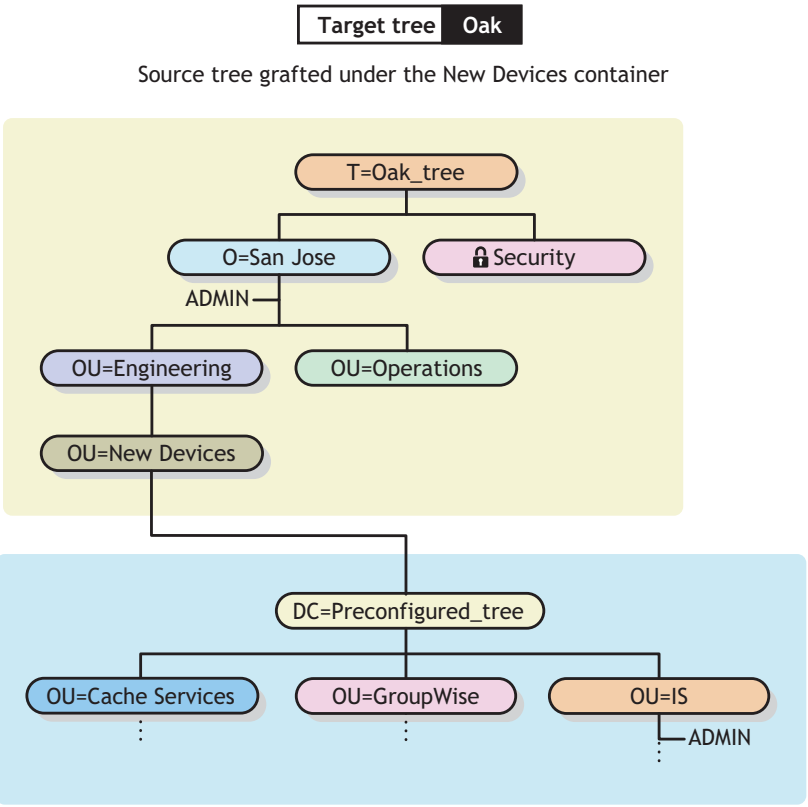


Figure 10-4 *Grafted eDirectory Tree*



This sections contains the following information:

- "Understanding Context Name Changes" on page 237
- "Preparing the Source and Target Trees" on page 238
- "Containment Requirements for Grafting" on page 239
- "Grafting the Source and Target Tree" on page 240

10.2.1 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 number of characters allowed in a DN of any type, including a container DN, is 255 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.

10.2.2 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 93. 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 one of the following: ◆ 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 `-672 No Access` 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 139.

Prerequisite	Required Action
The schema on both the source and target trees must be the same.	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. The graft operation automatically imports the schema from the target tree to the source tree. Run DSMerge again.
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 541.
The source tree's time reference must be reconfigured.	The source tree should usually be set as a secondary server configured to get its time source from a server in the target tree. To reconfigure Timesync, see "Configuring and Administering Time Synchronization" (http://www.novell.com/documentation/oes11/oes_implement_lx/data/time.html#time-cfgnadmin) in the *OES Planning and Implementation Guide*.

10.2.3 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	The target tree container object must include the domain object in its containment list. You can check this using *iMonitor* > *Schema*. If the containment list does not include Domain, run DSRepair to make schema enhancements.
Source Tree Requirements	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. You can check this using *iMonitor* > *Schema*. If the containment list does not include Domain, run DSRepair to make schema enhancements.

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.

10.2.4 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.

10.3 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, 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 user name 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.

10.4 Using the 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 Client, see Section 21.1, "Using the Command Line Client," on page 516.

10.4.1 Using the DSMerge eMTool

1. Run the Client in interactive mode by entering the following at the command line:

   ```
   java -cp path_to_the_file/emboxclient.jar  -i
   ```

   (If you have already put the `emboxclient.jar` file in your class path, you need to enter only `java -i`.)

   The Client prompt appears:

   ```
   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 8028, unless you have a Web server that is already using the port. The `-n` option opens a nonsecure connection.

   The 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 user name `Admin` and user password `test`) with the source tree you are currently logged in to (with source tree user name `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 user name `Admin` and user password `test`) into the `Fruit` container in the target tree `Orange` (with target tree user name `Admin` and user password `test`).

A space must be between each switch. The order of the switches is not important.

The Client will indicate whether the DSMerge operation was successful.

See "DSMerge eMTool Options" on page 242 for more information on the DSMerge eMTool options.

4 Log out from the Client by entering the following command:

```
logout
```

5 Exit the Client by entering the following command:

```
exit
```

10.4.2 DSMerge eMTool Options

The following tables lists the DSMerge eMTool options. You can also use the `list -t dsmerge` command in the Client to list the DSMerge options with details. See "Listing eMTools and Their Services" on page 519 for more information.

Merge Operation	Client Command
Check whether the tree can be renamed	`dsmerge.pr -u`*`User`* `-p`*`User_password`* `-n`*`New_tree_name`*
Rename the tree	`dsmerge.r -u`*`User`* `-p`*`User_password`* `-n`*`New_tree_name`*
Check whether two trees can be merged	`dsmerge.pm -u`*`Source_tree_user-`* `p`*`Source_tree_user_password`* `-T`*`Target_tree_name`* `-` `U`*`Target_tree_user`* `-P`*`Target_tree_password`*
Merge two trees	`dsmerge.m -u`*`Source_tree_user-`* `p`*`Source_tree_user_password`* `-T`*`Target_tree_name-`* `U`*`Target_tree_user`* `-P`*`Target_tree_password`*

Check whether the source tree can be grafted into the target tree container	`dsmerge.pg -u`*`Source_tree_user`* `-`
	`p`*`Source_tree_user_password`* `-T`*`Target_tree_name`* `-`
	`U`*`Target_tree_user`* `-P`*`Target_tree_password`* `-`
	`C`*`Target_tree_container`*

Graft the source tree into the container in the target tree	`dsmerge.g -u`*`Source_tree_user`* `-`
	`p`*`Source_tree_user_password`* `-T`*`Target_tree_name`* `-`
	`U`*`Target_tree_user`* `-P`*`Target_tree_password`* `-`
	`C`*`Target_tree_container`*

Cancel the running DSMerge operation	`cancel`

11 Encrypting Data In eDirectory

In Novell eDirectory 8.8 and later, you can encrypt specific data when they are stored on the disk and when they are transmitted between two or more eDirectory 8.8 servers. This provides greater security for the confidential data.

Refer to the *Novell eDirectory 8.8 SP7 What's New Guide* (http://www.novell.com/documentation/edir88/edir88new/data/front.html) for more information on the need for encryption of data and the scenarios in which you can encrypt data.

You can protect data by encrypting the following:

- Attributes: For protecting confidential data stored on the disk.

  See Section 11.1, "Encrypted Attributes," on page 243.
- Replication: For protecting confidential data during replication between eDirectory 8.8 servers.

  See Section 11.2, "Encrypted Replication," on page 253.

11.1 Encrypted Attributes

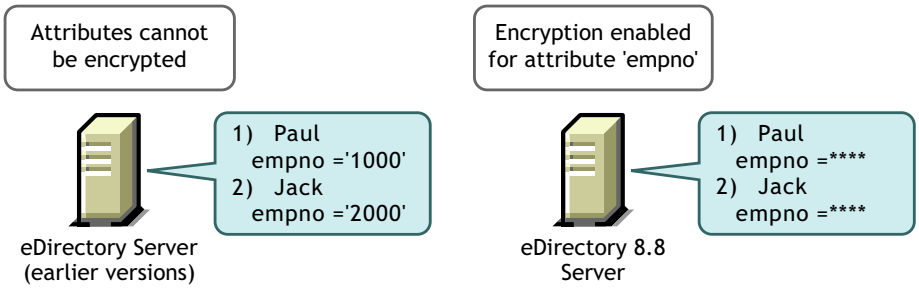
In eDirectory 8.8 and later, you can encrypt the attributes to protect data while they are stored on the disk. Encrypted attributes is a server-specific feature.

When you encrypt an attribute, the value of the attribute is encoded. For example, you can encrypt an attribute `empno` stored in the DIB. If empno=1000, then the value of the attribute (1000), is not stored as clear text on the disk. You can read this encrypted value only when you access the directory over a secure channel.

All attributes in a schema can be enabled for encryption. However, we recommend you not to enable Common Name (CN) attribute for encryption and enable only the sensitive data for encryption. Refer to Section 11.3, "Achieving Complete Security While Encrypting Data," on page 264 before you decide on marking any attributes for encryption.

There is no limitation in accessing Public and Server readable encrypted attributes, this means that a client can access these attributes over clear text but you can mark these attributes for encryption at the DIB level. Enabling encryption on an attribute which is flagged `[Public Read]` in schema, does not prevent it from being accessed via non-secure methods.

Figure 11-1 *Encrypted Attributes*



The data in eDirectory can be stored in any of the following ways:

- In the Data Information Base (DIB) or database
- As backup data
- LDIF file

You can encrypt attributes by creating and applying encrypted attributes policies to the servers.

To encrypt the attributes, do the following using iManager:

1. Create and define an encrypted attribute policy.
   - **1a** Select the attributes for encryption.
   - **1b** Select the encryption scheme for the attributes.

     Refer to "Creating and Defining Encrypted Attributes Policies" on page 247 for more information.
2. Apply the encrypted attributes policy to a server.

   Refer to "Applying Encrypted Attributes Policies" on page 247 for more information.

You can also encrypt attributes through LDAP.

Refer to "Managing Encrypted Attributes Policies Through LDAP" on page 247 for more information.

NOTE: Encrypted Attributes Policy assignment takes effect when Limber runs.

As a best practice, we recommend you to do the following:

- Mark only sensitive attributes for encryption. Do not mark all attributes for encryption (for example, public or server readable attributes).
- Use AES while marking an attribute for encryption as it is the strong encryption algorithm.

The rest of this section provides the following information:

- Section 11.1.1, "Using Encryption Schemes," on page 245
- Section 11.1.2, "Managing Encrypted Attributes Policies," on page 245
- Section 11.1.3, "Accessing the Encrypted Attributes," on page 250
- Section 11.1.4, "Viewing the Encrypted Attributes," on page 251
- Section 11.1.5, "Encrypting and Decrypting Backup Data," on page 252
- Section 11.1.6, "Cloning the DIB Fileset Containing Encrypted Attributes," on page 252

- Section 11.1.7, “Adding eDirectory 8.8 Servers to Replica Rings,” on page 252
- Section 11.1.8, “Backward Compatibility,” on page 252
- Section 11.1.9, “Migrating to Encrypted Attributes,” on page 252
- Section 11.1.10, “Replicating the Encrypted Attributes,” on page 252

11.1.1 Using Encryption Schemes

eDirectory 8.8 provides the highest level of security for an attribute by supporting the following encryption schemes:

- Advanced Encryption Standard (AES)
- Triple DES
- Data Encryption Standard (DES)

You can select different encryption schemes for different attributes in a single encrypted attributes policy. For example, in an encrypted attributes policy EP1, you can select both AES as the encryption scheme for an attribute cubeno and Triple DES for an attribute empno. Refer to “Creating and Defining Encrypted Attributes Policies” on page 247 for more information.

You can change the encryption scheme for an encrypted attribute by editing the encrypted attributes policy. You can also unencrypt an attribute that you have encrypted earlier. Refer to “Editing Encrypted Attributes Policies” on page 247 for more information.

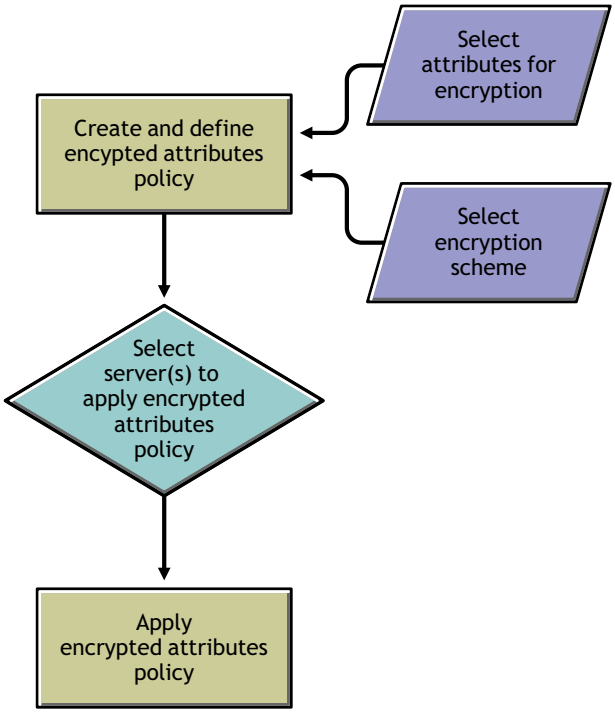
You can choose to have different encryption schemes in different servers of the replica ring. For example, an attribute might be enabled for encryption using AES on Server1, Triple DES on Server2 and no encryption scheme on Server3.

11.1.2 Managing Encrypted Attributes Policies

You can manage encryption of the attributes by creating and defining policies and applying them to servers.

You define an encrypted attributes policy by selecting the attributes for encryption and an encryption scheme.

Figure 11-2 *Encrypting Attributes*



You can manage encrypted attributes policies using iManager. This section provides the following information:

- "Managing Encrypted Attributes Policies Through iManager" on page 246
- "Managing Encrypted Attributes Policies Through LDAP" on page 247
- "Copying the Encrypted Attributes Policies" on page 249
- "Partition Operations" on page 249

Managing Encrypted Attributes Policies Through iManager

This section contains the following procedures:

- "Creating and Defining Encrypted Attributes Policies" on page 247
- "Editing Encrypted Attributes Policies" on page 247
- "Applying Encrypted Attributes Policies" on page 247
- "Deleting Encrypted Attributes Policies" on page 247

If encrypted attributes are present in the eDirectory server, iManager behaves in the following manner:

1. Reading, listing, or modifying encrypted attributes is not allowed over clear text or secure channel.
2. An entry that has non-encrypted attributes is not allowed to read, list, or modify attributes through iManager over clear text or secure channel. This implies that the whole entry is blocked.

Creating and Defining Encrypted Attributes Policies

1. In Novell iManager, click the *Roles and Tasks* button .
2. Click *eDirectory Encryption* > *Encryption Attributes*.
3. In the Encrypted Attributes Policies Management Wizard, select *Create, Edit and Assign policy*.
4. Follow the instructions in the Encrypted Attributes Policies Management Wizard to create and define the policy.

   Help is available throughout the wizard.

Editing Encrypted Attributes Policies

1. In Novell iManager, click the *Roles and Tasks* button .
2. Click *eDirectory Encryption* > *Encryption Attributes*.
3. In the Encrypted Attributes Policies Management Wizard, select *Edit policy*.
4. Follow the instructions in the Encrypted Attributes Policies Management Wizard to edit the policy.

   Help is available throughout the wizard.

Applying Encrypted Attributes Policies

1. In Novell iManager, click the *Roles and Tasks* button .
2. Click *eDirectory Encryption* > *Encryption Attributes*.
3. In the Encrypted Attributes Policies Management Wizard, select *Create, Edit and Assign policy*.
4. Follow the instructions in the Encrypted Attributes Policies Management Wizard to apply the policy.

   Help is available throughout the wizard.

Deleting Encrypted Attributes Policies

1. In Novell iManager, click the *Roles and Tasks* button .
2. Click *eDirectory Encryption* > *Encryption Attributes*.
3. In the Encrypted Attributes Policies Management Wizard, select *Delete policies*.
4. Follow the instructions in the Encrypted Attributes Policies Management Wizard to delete the policy.

   Help is available throughout the wizard.

Managing Encrypted Attributes Policies Through LDAP

IMPORTANT: We strongly recommend you to use iManager for managing encrypted attributes and not LDAP.

This section contains the following procedures:

- "Creating and Defining Encrypted Attributes Policies" on page 248
- "Editing Encrypted Attributes Policies" on page 248

- "Applying Encrypted Attributes Policy" on page 249
- "Deleting Encrypted Attributes Policy" on page 249

NOTE: You should specify the attribute and scheme pair while marking any attribute through LDIF for encryption and not the list of attributes and scheme. This is the current limitation with encrypted attributes.

Creating and Defining Encrypted Attributes Policies

1 Create an attribute encryption policy.

For example, the encrypted attributes policy is AE Policy- test-server, then

```
dn: cn=AE Policy - test-server, o=novell
changetype: add
objectClass: encryptionPolicy
```

2 Add the attrEncryptionDefinition attribute to the Policy object you created and mark the attributes for encryption.

For example, if the attribute name you want to encrypt is CRID then specify the encryption scheme and attribute name as mentioned below:

```
dn: cn=AE Policy - test-server, o=novell
changetype: modify
add: attrEncryptionDefinition
attrEncryptionDefinition: aes$CRID
```

NOTE: Attribute name specifies the NDS name for the attribute. Many attributes in eDirectory have both an LDAP name and an NDS name. Here, specify the attribute name that requires the NDS name.

---

3 Add the attrEncryptionRequiresSecure attribute to the policy.

The value of this attribute specifies whether a secure channel is always necessary to access the encrypted attributes. The value 0 means that it is not always necessary. The value 1 means that it is always necessary.

For example:

```
dn: cn=AE Policy - test-server, o=novell
changetype: modify
add: attrEncryptionRequiresSecure
attrEncryptionRequiresSecure: 0
```

4 Associate the policy with an NCP server.

For example, if the NCP server is test-server:

```
dn: cn=test-server, o=novell
changetype: modify
add: encryptionPolicyDN
encryptionPolicyDN: cn=AE Policy - test-server, o=novell
```

Editing Encrypted Attributes Policies

The following LDIF file illustrates editing an encrypted attributes policy by changing the value of the attrEncryptionRequireSecure attribute:

```
dn: cn=AE Policy - test-server, o=novell
changetype: modify
replace: attrEncryptionRequiresSecure
attrEncrytionRequiresSecure: 1
```

Applying Encrypted Attributes Policy

The following LDIF file illustrates applying an encrypted attributes policy AE Policy-test-server to a server test-server:

```
dn: cn=test-server, o=novell
changetype: modify
add: encryptionPolicyDN
encryptionPolicyDN: cn=AE Policy - test-server, o=novell
```

Deleting Encrypted Attributes Policy

The following LDIF file illustrates deleting an encrypted attributes policy:

```
dn: cn=AE Policy - test-server, o=novell
changetype: delete
```

NOTE: For more information on managing encrypted attributes through LDAP, see Section 14.3, "Using LDAP Tools on Linux, Solaris, or AIX," on page 337 and Section 6.1, "Novell Import Conversion Export Utility," on page 147.

Copying the Encrypted Attributes Policies

In eDirectory 8.8 and later, you can copy the encrypted attributes policies to have identical configurations on many servers. The policies are stored as objects in eDirectory.

Refer to "Copying Objects" on page 97 for step-by-step information on copying a Policy object using iManager.

Partition Operations

When you merge two partitions, the policies of the parent are retained for the resultant partition. When you split a partition, the child partition inherits the policy of the parent partition.

Recommendation: eDirectory stores several attributes for its own operations which should not be marked for encryption. If these attributes are marked for encryption, some of the eDirectory functionality will possibly be broken or it will not perform as expected.

The attributes that should not marked for encryption are:

- federationBoundaryType
- Volume
- ACL
- federationBoundary
- member
- federationControl
- federationSearchPath
- encryptionPolicyDN
- indexDefinition

- dgIdentity
- dgAllowUnknown
- agTimeout
- Host Server
- hostResourcePath
- ndsPredicateState
- ndsStatusExternalReference
- ndsStausLimber
- ndsStatusSchema

Though the list is not exhaustive, similar kind of attributes should not be marked for encryption.

11.1.3 Accessing the Encrypted Attributes

When you encrypt the attributes, you also protect the access to the encrypted attributes. This is because eDirectory 8.8 and later can restrict the access to the encrypted attributes over secure channel such as LDAP secure channel or NCP secure channel. However, only Novell internal customers can set up and use a secure NCP connection because the Dclient application, with which a secure NCP connection is created, is not available for public use.

You can also back up the encrypted attributes by using the Backup (ndsbackup) utility.

By default, the encrypted attributes can be accessed only through a secure channel.

However, if you want the clients to be able to access the encrypted attributes over clear text, then disable the Always Require Secure Channel option. For more information, refer to "Enabling and Disabling Access to Encrypted Attributes Over Clear Text Channels" on page 250.

Enabling and Disabling Access to Encrypted Attributes Over Clear Text Channels

You can enable or disable the access to encrypted attributes over clear text channels by enabling or disabling Always Require Secure Channel option (that is, the attrEncryptionRequireSecure attribute) using either iManager or LDAP.

This section contains the following information:

- "Enabling and Disabling Access to Encrypted Attributes Over Clear Text Channels Using iManager" on page 250
- "Enabling and Disabling Access to Encrypted Attributes Over Clear Text Channels Using LDAP" on page 251

Enabling and Disabling Access to Encrypted Attributes Over Clear Text Channels Using iManager

To enable or disable the access to encrypted attributes over clear text channels using iManager, enable or disable Always Require Secure Channel in the Encrypted Attributes Policies Management Wizard while

- Creating and defining encrypted attributes policies.
- Editing encrypted attributes policies.

Enabling and Disabling Access to Encrypted Attributes Over Clear Text Channels Using LDAP

To enable or disable access to encrypted attributes over clear text channels using LDAP, add the following attribute to the encrypted attributes policy:

`attrEncryptionRequiresSecure`

Setting this attribute to 0 makes a secure channel not always necessary, that is, you can access the encrypted attributes over a clear text channel. Setting it to 1 makes a secure channel always necessary, that is, you can access the encrypted attributes over a secure channel only.

Refer to Step 3 on page 248 for more information.

11.1.4 Viewing the Encrypted Attributes

Viewing the attributes that are encrypted depends on whether you have enabled or disabled the Always Require Secure Channel option. This means whether you have specified that the encrypted attributes need a secure channel to access them or not.

- "Viewing Encrypted Attributes Using iManager" on page 251
- "Viewing Encrypted Attributes Using DSBrowse" on page 251
- "SNMP Traps" on page 251

Viewing Encrypted Attributes Using iManager

If Always Require Secure Channel is enabled, you cannot view the encrypted attributes. You get the error -6089, indicating that you need a secure channel to access the encrypted attributes.

If Always Require Secure Channel is disabled, you can see the encrypted attributes values in iManager.

For more information, refer to "Browsing Objects in Your Tree" on page 214.

Viewing Encrypted Attributes Using DSBrowse

If you have enabled the Always Require Secure Channel option, that is, if a secure channel is always required to access the encrypted attributes, you cannot view those attributes of the entry that are marked for encryption. However, you can view the other attributes of the entry that are not encrypted.

SNMP Traps

NDS® Value Events are blocked if you have specified that you always need a secure channel to access the encrypted attributes. Traps that are related to value events have value data as NULL and the result will be set to -6089, which indicates that you need a secure channel to get the encrypted attribute value. The following traps have the value data as NULL:

- ndsAddValue
- ndsDeleteValue
- ndsDeleteAttribute

11.1.5 Encrypting and Decrypting Backup Data

While backing up data on a server that has attributes marked for encryption, you are prompted to provide a password to encrypt or decrypt backup data. The `-E` option in the Backup utility facilitates this. For more information, refer to the `ndsbackup` man page.

For more information on backing up your data, refer to Chapter 16, "Backing Up and Restoring Novell eDirectory," on page 391.

11.1.6 Cloning the DIB Fileset Containing Encrypted Attributes

While cloning, if the eDirectory database contains encrypted attributes in it, then the cloned DIB fileset will also have these attribute values encrypted. You need to set a password to secure the key used by eDirectory to encrypt the values in the cloned DIB fileset. When you place the cloned DIB fileset on another server, you will be asked to provide this password.

For more information, refer to "Clone DIB Set" on page 219.

11.1.7 Adding eDirectory 8.8 Servers to Replica Rings

You can add eDirectory 8.8 servers to replica rings irrespective of whether the attributes are marked for encryption on one or all the servers hosting the replica or whether Always Require Secure Channel is enabled or disabled.

For more information on adding eDirectory 8.8 server to the replica ring, refer to "Adding a Replica" on page 139.

11.1.8 Backward Compatibility

You need to change all eDirectory utilities like iManager, SNMP, DirXML® and NSureAudit to secure NCP™ to access encrypted attributes. Otherwise, you need to specify that a secure channel is not necessary to access the encrypted attributes. Refer to "Enabling and Disabling Access to Encrypted Attributes Over Clear Text Channels" on page 250 for more information.

11.1.9 Migrating to Encrypted Attributes

When you upgrade to eDirectory 8.8 or later versions, you can encrypt the existing attributes by creating and defining encrypted attributes policies. For more information, refer to Section 11.1.2, "Managing Encrypted Attributes Policies," on page 245.

11.1.10 Replicating the Encrypted Attributes

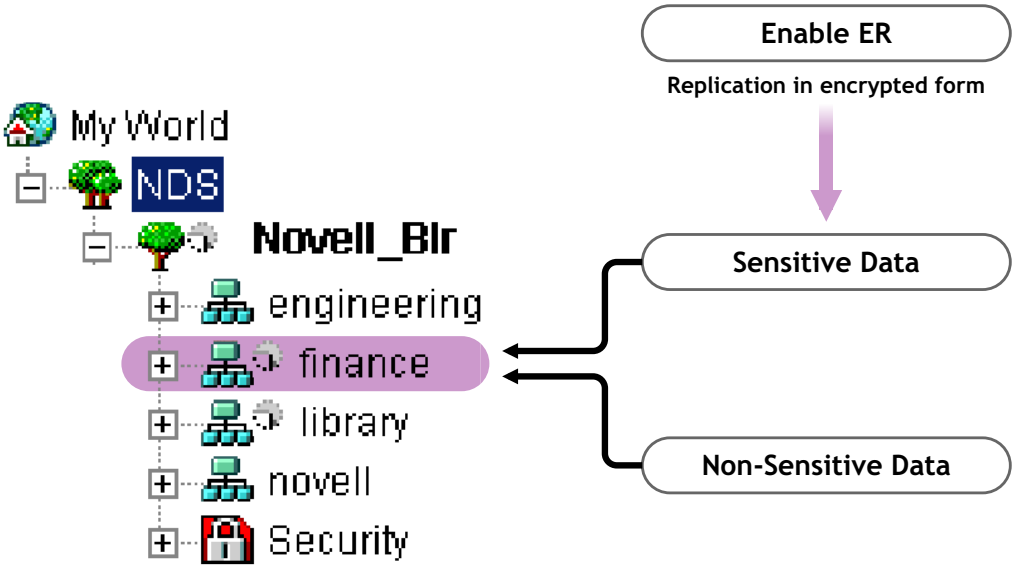
By default, encrypted replication is not enabled even if the server has the encrypted attributes. You need to enable encrypted replication for replicating the encrypted attributes securely. For configuring encrypted replication, refer to Section 11.2, "Encrypted Replication," on page 253.

11.2 Encrypted Replication

In Novell eDirectory 8.8 and later, you can encrypt data that is transmitted between eDirectory 8.8 servers. This offers a high level of security during replication as the data does not flow in clear text.

Refer to the *Novell eDirectory 8.8 SP7 What's New Guide* (http://www.novell.com/documentation/edir88/edir88new/data/bqljql1.html#bqljql1) for more information on the need for encrypted replication and sample scenarios of encrypting data during replication.

Figure 11-3 *Encrypted Replication*



In the above illustration, "finance" and "library" are the partitions in the tree. "finance" might contain sensitive data that requires encryption while replicating. You can enable the partition "finance" for encrypted replication. Partitions like "library" that might not contain sensitive data need not be enabled for encrypted replication.

IMPORTANT: When you enable encrypted replication for a partition, the replication process might slow down. You can enable or disable encrypted replication using iManager.

This section provides the following information:

- Section 11.2.1, "Enabling Encrypted Replication," on page 253
- Section 11.2.2, "Adding a New Replica to a Replica Ring," on page 258
- Section 11.2.3, "Synchronization and Encrypted Replication," on page 263
- Section 11.2.4, "Viewing the Encrypted Replication Status," on page 263

11.2.1 Enabling Encrypted Replication

To enable encrypted replication, you need to configure a partition for encrypted replication. Configuration settings are stored in the partition Root object.

You can choose to enable encrypted replication at a partition level or replica level.

The configurations at the partition level are overridden by the configurations at the replica level. This means, if encrypted replication is

- Enabled at partition level and disabled for specific replicas, then the replication between the specific replicas happens in clear text.
- Disabled at partition level and enabled for specific replicas, then the replication between the specific replicas happens in encrypted form.

Table 11-1 *Overriding Encrypted Replication Configuration at the Partition Level*

Partition Level	Replica Level	Replication
Enabled	Disabled	Unencrypted
Disabled	Enabled	Encrypted

This section contains the following procedures:

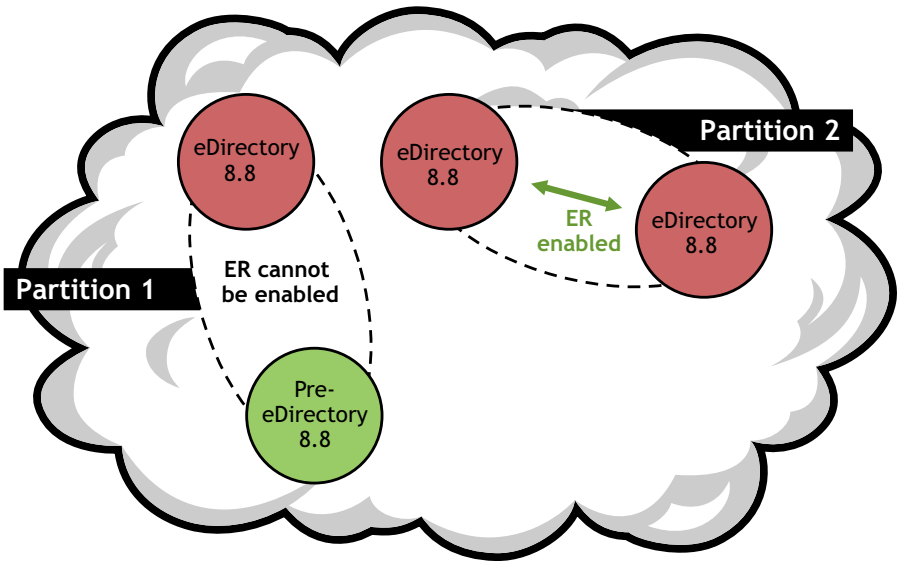
- "Enabling Encrypted Replication at the Partition Level" on page 254
- "Enabling Encrypted Replication at the Replica Level" on page 256

Enabling Encrypted Replication at the Partition Level

When you enable encrypted replication at a partition level, replication between all the replicas hosting the partition is encrypted. For example, consider partition P1 has replicas R1, R2, R3, and R4. You can encrypt the replication between all the replicas, and all replications, inbound or outbound, are encrypted for these replicas.

To enable a partition for encrypted replication, all the servers hosting the partition must be eDirectory 8.8 or later servers. Other partitions in the tree that are not enabled for encrypted replication can have pre-eDirectory 8.8 servers.

Figure 11-4 *Encrypted Replication at Partition Level*



The configurations for encrypted replication at the partition level are overridden if you have encrypted replication configurations at replica level. Refer to Table 11-1 on page 254.

Backward compatibility depends on whether the encrypted replication is enabled or disabled at the partition level. Refer to Section 11.2.2, “Adding a New Replica to a Replica Ring,” on page 258 for more information.

You can enable encrypted replication at the partition level using iManager or LDAP, as explained in the following sections:

- “Enabling Encrypted Replication at the Partition Level using iManager” on page 255
- “Enabling Encrypted Replication at the Partition Level Using LDAP” on page 255
- “Partition Operations” on page 258

Enabling Encrypted Replication at the Partition Level using iManager

1. Click the *Roles and Tasks* button .
2. Click *eDirectory Encryption* > *Encrypted Replication*.
3. Enter or browse to the partition for which you want to enable encrypted replication.
4. Click *Next*.
5. In the Encrypted Replication Wizard, select *Encrypt all Replica synchronizations*.

   Help is available throughout the Wizard.

   NOTE: To disable encrypted replication at the partition level, unselect *Encrypt all Replica synchronizations*.
6. Click *Finish*.

In the Encrypted Replication Wizard, when you enable encrypted replication for the whole partition, you can disable encrypted replication for specific replicas. The replicas that you disable for encrypted replication will not receive or send data in encrypted form. You can also disable encryption for the entire partition by deselecting *Encrypt all Replica synchronizations*.

Enabling Encrypted Replication at the Partition Level Using LDAP

IMPORTANT: We strongly recommend you to use iManager for enabling encrypted replication.

To encrypt replication, you need to use the attribute dsEncryptedReplicationConfig. The syntax is:

```
enable/disable flag#destination replica number#source replica number
```

Replace with either of these flags:

- 0: Encrypted replication is disabled
- 1: Encrypted replication is enabled

Source replica number and destination replica number represents source and destination replica numbers of a partition. These numbers can be specified in any order because if the replication from A to B is encrypted, then replication from B to A is also encrypted.

NOTE: If the source and destination replica number at the partition level is 0 and if the flag is set to 1, all the replicas are considered to be enabled for encrypted replication.

To enable encrypted replication at the partition level, the value of the dsEncryptedReplicationConfig attribute should be set to 1#0#0.

Following is a sample LDIF file for enabling encrypted replication at the partition level:

```
dn: o=ou
changetype:modify
replace: dsEncryptedReplicationConfig
dsEncryptedReplicationConfig:1#0#0
```

These configurations at the partition level are overridden by the configurations at the replica level. Refer to "Enabling Encrypted Replication at the Replica Level using LDAP" on page 257 for more information.

Enabling Encrypted Replication at the Replica Level

When you enable encrypted replication at the replica level, replication between specific replicas is encrypted. Both outbound and inbound replication between the replicas are encrypted.

For example, consider partition P1 has replicas R1, R2, R3, and R4. You can encrypt the replication between replicas R1 and R2 or between R2 and R4.

To enable encrypted replication between replicas of a partition, you need to define an encryption link between the replicas. Refer to "Enabling Encrypted Replication at the Replica Level Using iManager" on page 256 for more information.

If you have enabled encrypted replication for one replica, it means that:

- the inbound synchronization from a server to this replica
- outbound synchronization from this replica to any other server is encrypted.

The replicas you have enabled for encrypted replication must be on eDirectory 8.8 servers. The remaining replicas in the replica ring, that are not enabled for encrypted replication, can be on servers with earlier versions of eDirectory.

If you have enabled only specific replicas for encrypted replication, you can add an eDirectory 8.8 server or a pre-eDirectory 8.8 server to the replica ring.

To disable encrypted replication at the replica level, you need to disable *Encrypt Link* for specific replicas using Encrypted Replication Configuration Wizard in iManager.

You can enable encrypted replication at the replica level using either iManager or LDAP as described in the following sections:

- "Enabling Encrypted Replication at the Replica Level Using iManager" on page 256
- "Enabling Encrypted Replication at the Replica Level using LDAP" on page 257

Enabling Encrypted Replication at the Replica Level Using iManager

You can enable encrypted replication at replica level through iManager by creating encryption links. Encryption links connect the replicas between which you want the replication to be encrypted. You create encryption links while configuring a replica for encrypted replication by selecting a source replica and one or more destination replicas.

For example, consider partition P1 having replicas R1, R2, R3, and R4. To encrypt replication between replicas R1 and R2, you need to create an encryption link by identifying one of them as the source and the other as the destination replica.

After creating encryption links, you can choose to encrypt these links for specific replicas by selecting or deselecting *Encrypt Link* in the Encrypted Replication Configuration Wizard in iManager. Refer to "Enabling Encrypted Replication at the Replica Level Using iManager" on page 256 for more information.

To enable encrypted replication at the replica level:

1. Click the *Roles and Tasks* button .
2. Click *eDirectory Encryption* > *Encrypted Replication*.
3. Enter or browse to the partition for which you want to enable encrypted replication.
4. Click *Next*.
5. In the Encrypted Replication Wizard, in the *Encrypted synchronizations* table, click *New* to define an encryption link.
   - **5a** In the *Select Source Replica* field, specify or browse to the replica you want to use as the source.
   - **5b** In the *Destination Replicas* field, specify or browse to one or more replicas you want to use as the destination for replication.
   - **5c** Select *Encrypt Link*.
   - **5d** Click *OK*.
6. Click *Finish*.

Enabling Encrypted Replication at the Replica Level using LDAP

IMPORTANT: We strongly recommend you to use iManager for enabling encrypted replication.

To encrypt replication, you need to use the attribute `dsEncryptedReplicationConfig`. The syntax is:

```
enable/disable flag#destination replica number#source replica number
```

For more information on the syntax, refer to "Enabling Encrypted Replication at the Partition Level Using LDAP" on page 255.

When you specify the replicaNumber of the replicas in the above syntax, you enable the encrypted replication between those replicas. consider the following example syntaxes:

- 1#0#1: Encrypted replication is enabled from and to replica number 1; to and from, every other replica in the partition.
- 0#3#1: Encrypted replication is disabled between replica numbers 3 and 1.
- 0#1#1: Encrypted replication is disabled for replica number 1.

The following is a sample LDIF file that disables encrypted replication between replica numbers 1 and 3:

```
dn: o=ou
changetype: modify
replace: dsEncryptedReplicationConfig
dsEncryptedReplicationConfig: 0#3#1
```

Partition Operations

When you split a partition, the encrypted replication configuration in the parent partition is inherited by the child partition. When you merge a partition, the encrypted replication configuration of the parent partition is retained in the resultant partition.

11.2.2 Adding a New Replica to a Replica Ring

Adding new replica to a replica ring is affected by whether encrypted replication is enabled or disabled for the partition at the partition and replica level.

For more information on adding a replica to a replica ring, refer to Section 5.5, "Administering Replicas," on page 139.

At each of the above levels, you have different scenarios depending on which version of eDirectory server you are trying to add to the replica ring, as explained in the following sections:

- "Enabling Encrypted Replication at the Partition Level" on page 258
- "Enabling Encrypted Replication at the Replica Level" on page 262
- "Enabling Encrypted Replication for the Server You Add" on page 263

Enabling Encrypted Replication at the Partition Level

The scenarios vary depending on the version of eDirectory server you are trying to add. This section contains the following information:

- "Adding Pre-eDirectory 8.8 Servers to the Replica Ring" on page 258
- "Adding eDirectory 8.8 Servers to the Replica Ring" on page 260

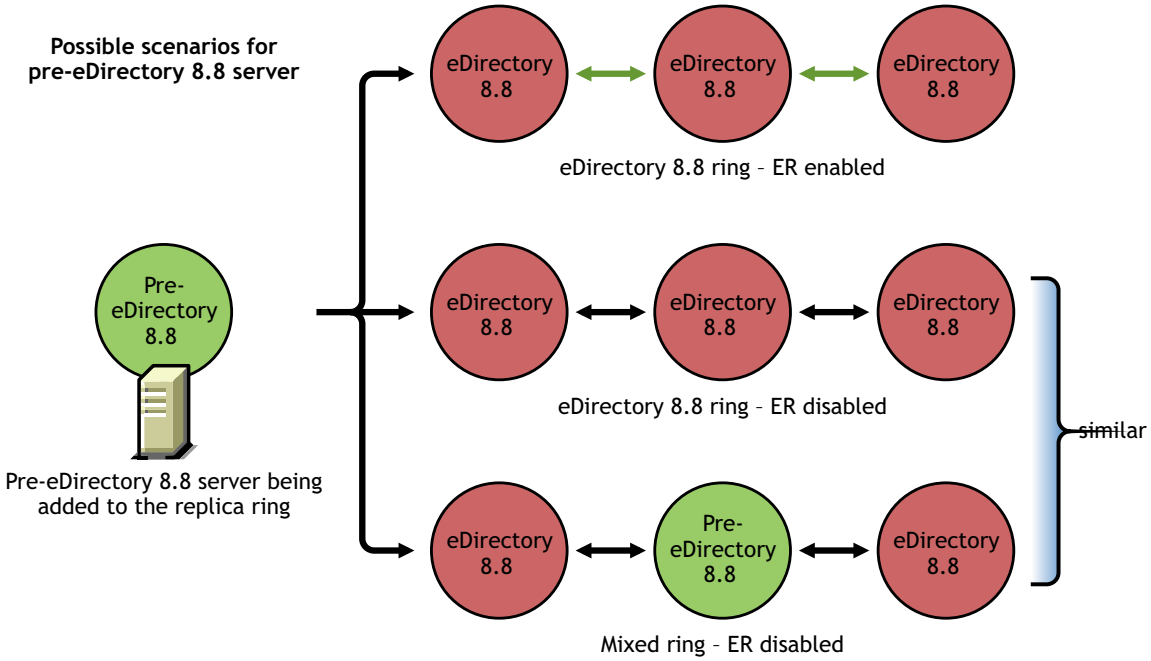
Adding Pre-eDirectory 8.8 Servers to the Replica Ring

The following illustration gives you the possible scenarios when you add a pre-eDirectory 8.8 server to the replica ring:

- Scenario A
- Scenario B
- Scenario C

NOTE: ER in the graphic below indicates encrypted replication.

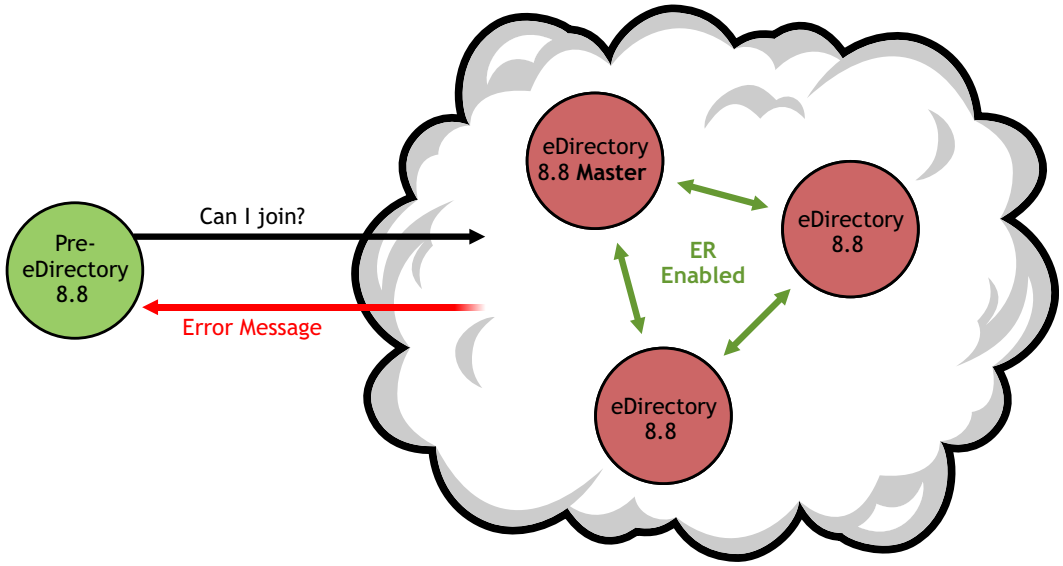
Figure 11-5 *Possible Scenarios for Pre-eDirectory 8.8 Server*



Scenario A: Adding a Pre-eDirectory 8.8 server to an eDirectory 8.8 Replica Ring with Encrypted Replication Enabled

When you try to add a pre-eDirectory 8.8 server to an eDirectory 8.8 replica ring for which you have enabled the encrypted replication, you get the ERR_INCOMPATIBLE_DS error. You will be able to add the server to the replica ring, but you cannot have a replica of the partition on the server.

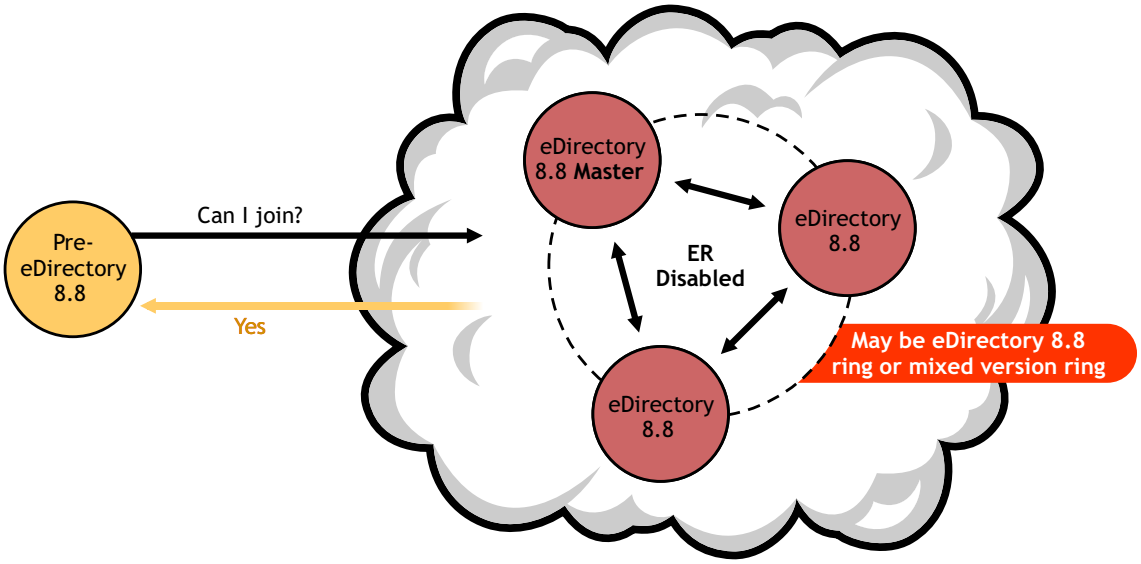
Figure 11-6 *Adding Pre-eDirectory 8.8 Server to eDirectory 8.8 Replica Ring with Encrypted Replication Enabled.*



Scenario B: Adding a Pre-eDirectory 8.8 Server to an eDirectory 8.8 Replica Ring with Encrypted Replication Disabled

You can add a pre-eDirectory 8.8 server to an eDirectory 8.8 replica ring with encrypted replication disabled.

Figure 11-7 *Adding Pre-eDirectory 8.8 Server to Replica Ring with Encrypted Replication Disabled*



Scenario C: Adding a Pre-eDirectory 8.8 Server to a Mixed Replica Ring with Encrypted Replication Disabled

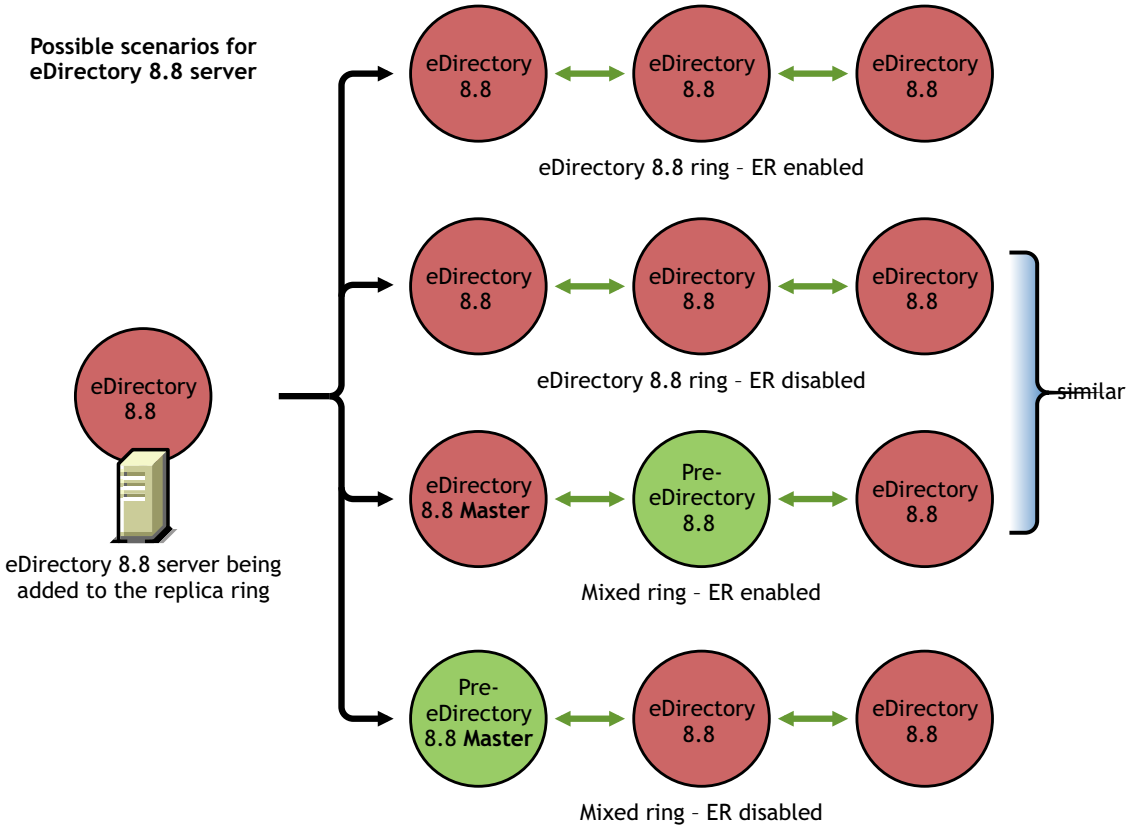
You can add a pre-eDirectory 8.8 server to a replica ring having a mixed version of eDirectory with encrypted replication disabled. Refer to Figure 43 above.

Adding eDirectory 8.8 Servers to the Replica Ring

The following illustration gives you the possible scenarios when you add eDirectory 8.8 server to the replica ring:

- Scenario A
- Scenario B
- Scenario C
- Scenario D

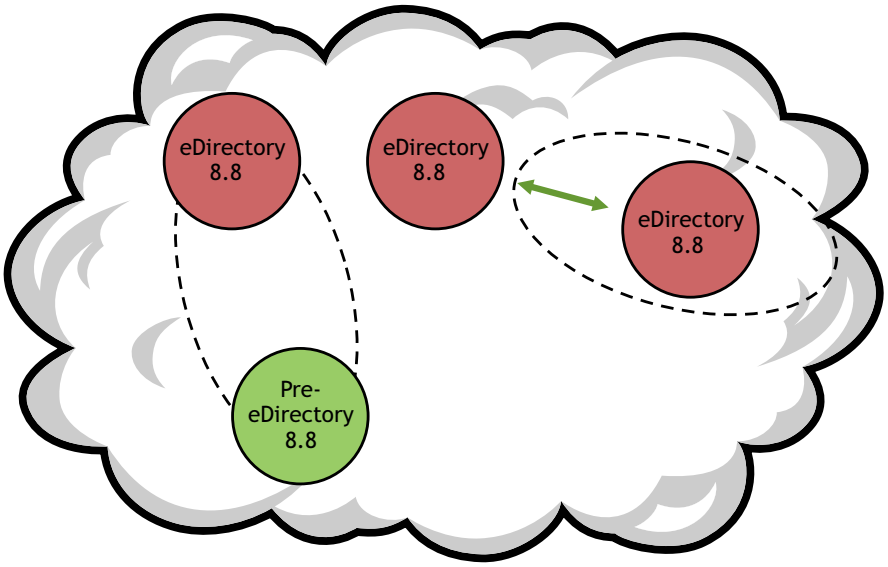
Figure 11-8 *Possible Scenarios for eDirectory 8.8 Server*



Scenario A: Adding eDirectory 8.8 Servers to an eDirectory 8.8 Replica Ring with Encrypted Replication Enabled

In this case, the encrypted replication would already be enabled on the added eDirectory 8.8 server.

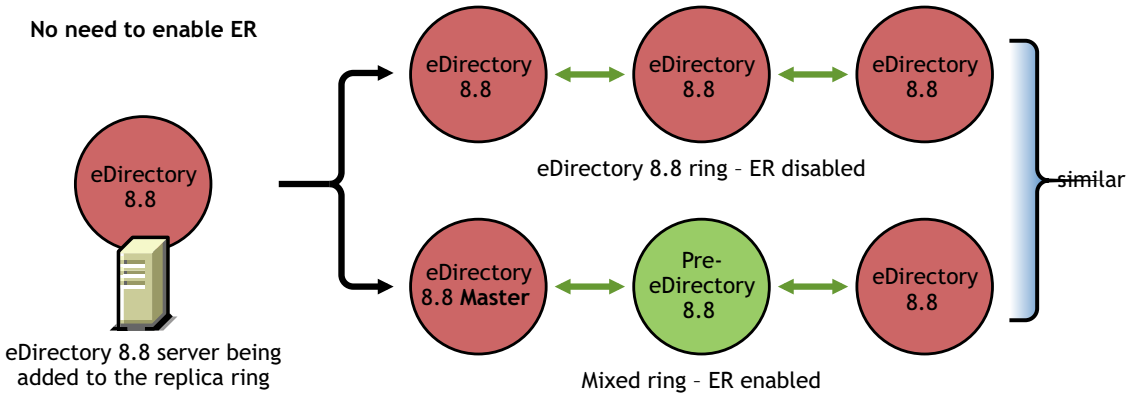
Figure 11-9 *Adding eDirectory 8.8 Server to eDirectory Replica Ring with Encrypted Replication Enabled*



Scenario B: Adding eDirectory 8.8 Servers to an eDirectory 8.8 Replica Ring with Encrypted Replication Disabled

In this case, encrypted replication will be disabled on the added eDirectory 8.8 server.

Figure 11-10 *Adding eDirectory 8.8 Server to Replica Rings where Encrypted Replication is Disabled.*



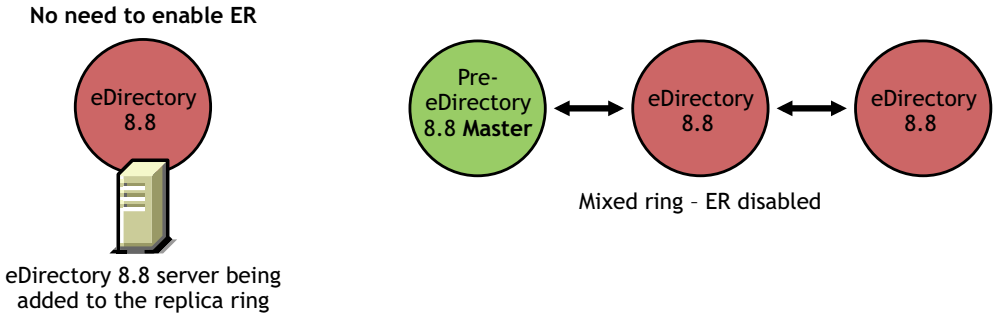
Scenario C: Adding eDirectory 8.8 Servers to a Mixed Replica Ring where Master Replica Is an eDirectory 8.8 Server and Encrypted Replication Is Disabled

In this case, you do not need to enable encrypted replication on the eDirectory 8.8 server you are trying to add. Refer to Figure 11-10 on page 262.

Scenario D: Adding eDirectory 8.8 Servers to a Mixed Replica Ring where Master Replica is a Pre-eDirectory 8.8 Server and Encrypted Replication is Disabled

In this case, you do not need to enable encrypted replication on the eDirectory 8.8 server you are trying to add.

Figure 11-11 *Adding eDirectory 8.8 server to a Replica Ring where Master Replica is a Pre-eDirectory 8.8 Server*



Enabling Encrypted Replication at the Replica Level

If encrypted replication is enabled between a source replica and specific destination replicas, you can add an eDirectory 8.8 server or a pre-eDirectory 8.8 server to the replica ring.

The scenarios vary if encrypted replication is enabled between a source replica and all the other replicas in the replica ring. This is similar to adding replicas to a replica ring with encrypted replication enabled or disabled at the partition level. Refer to "Enabling Encrypted Replication at the Partition Level" on page 258 for more information.

Enabling Encrypted Replication for the Server You Add

If the server you are trying to add is on Linux and UNIX, you can use the `ndsconfig -E` option to enable encrypted replication on the server. Refer to the ndsconfig man pages for more information.

If the server you are trying to add is on Windows, you can enable the Enable Encrypted Replication option in the installation wizard.

If the server you are trying to add is on platforms other than Linux and UNIX, you can enable encrypted replication through iManager or LDAP. Refer to Section 11.2.1, "Enabling Encrypted Replication," on page 253 for more information.

11.2.3 Synchronization and Encrypted Replication

If one replica is enabled for encrypted replication and the configuration changes are not synchronized with the other servers, replication happens in the encrypted form between the replicas. The replicas that are not synced with the configuration changes for encrypted replication continue to sync in clear text.

Even if the encrypted replication configuration has not been synchronized across the replicas, the replication between them will happen in the encrypted form.

11.2.4 Viewing the Encrypted Replication Status

You can view the encrypted replication status through iMonitor as follows:

1. In iMonitor, click *Agent Synchronization* in the Assistant frame.
2. Click *Replica Synchronization* for the partition you want to view.

   The replica status information is displayed. The *Encryption Status* field displays whether the link from the replica to which you are currently connected is encrypted or not.

   Basically, there are three scenarios in encryption replication (ER):

   - **ER enabled at partition level:** The replica to which you are connected to shows *Encryption State* is enabled.

     To find out which replica you are connected to, in the replica frame, the one that is not hyper linked is the one you are connected to. If you browse to the other replicas it shows that the *Encryption State* is also marked Enabled.
   - **ER enabled at replica level:** You have enabled ER for all replicas from one particular replica (that is, One to All.) In this case, when you are connected to that replica, its *Encryption State* is marked Enabled.
   - **ER enabled/disabled for a combination of replicas:** ER enabled/disabled for one combination of replicas - You have enabled ER for the whole partition but not for a selected set of servers or vice versa.

     For example, you have enabled ER for partition A that has three replicas 1, 2, and 3 and disabled ER for 1 <--> 3. In this case, if you are connected to replica 1, the *Encryption State* is displayed as:

     Server 1 Enabled

     Server 2

Server 3 Disabled

This means that Server 1 is enabled for encrypted replication to all the servers in the replica ring but 1<-->3 is disabled by the administrator.

11.3 Achieving Complete Security While Encrypting Data

The first important basic rule to be followed before encrypting the data is:

No information that would eventually be encrypted should ever be written to the hard disk (or any other media) in the clear.

When you mark existing clear text data for encryption, though the data gets encrypted, the existing clear text data might still be present on some part of hard disk where the DIB resides.

There will be "Left Over" clear text pieces of data in some blocks of database if you try to do following operations:

- Mark existing clear text data for encryption
- Change the encryption scheme of an encrypted attribute

The following sections depict deployment scenarios for encrypted data and steps to ensure that the encrypted data is truly secure:

- Section 11.3.1, "Encrypting Data in an All New Setup," on page 264
- Section 11.3.2, "Encrypting Data in an Existing Setup," on page 265
- Section 11.3.3, "Conclusion," on page 266

11.3.1 Encrypting Data in an All New Setup

In case of a new setup, you would have just installed the operating system and then eDirectory. It is assured that there is no clear text data present in the hard disk where the DIB resides.

Complete the following steps to ensure that the encrypted data in eDirectory is truly secure:

1. Plan in advance which attributes you want to encrypt and with what scheme.

   That is, you must decide in advance which attributes you want to encrypt before uploading the data in clear text into the eDirectory.

   WARNING: Once you have loaded any data into the eDirectory in the clear, you should not mark an attribute for encryption. Though you can do it, this leads to security problems.

2. Configure eDirectory and set the encryption schemes that you want on an attribute.
3. Load your existing data into the new server.

   Bulkloading from an LDIF file or replicating with another server are the two most likely scenarios. Make sure that if you bulk load, you don't copy the clear text LDIF file onto the same hard disk where the DIB resides.

   NOTE: Remember the Rule mentioned: No clear text data can ever be written to the disk.

4 Destroy any existing clear text data

Any disks (or on other media) with the clear text data on it should be securely wiped. This includes things like the clear text LDIF file used to bulk load the server, any other server that was used for replication, or tapes with old backups on them.

11.3.2 Encrypting Data in an Existing Setup

This scenario includes the following:

- "Existing Clear Text Data to Encrypted Data" on page 265
- "Changing the Scheme of the Encrypted Data" on page 266

Existing Clear Text Data to Encrypted Data

You can mark clear text data for encryption and ensure that the data is secure through the following methods:

- "Through Replication" on page 265
- "Through Backup and Restore" on page 265

Through Replication

1 Setup encryption on a new server as follows:

1a Plan in advance which attributes you want to encrypt and with what scheme.

That is, you must decide in advance which attributes you want to encrypt before uploading the data in clear text into the eDirectory.

WARNING: Once you have loaded any data into the eDirectory in the clear, you should not mark an attribute for encryption. Though you can do it, this leads to security problems.

1b Start with a clear install (probably including the OS) on a freshly formatted and partitioned disk.

This is to ensure that there is no clear text data on the disk. This means you cannot just take an existing computer which has clear text data previous and re-install eDirectory. You must have thoroughly erased all traces of data from the disk. Run some kind of secure erase software, use a magnetic bulk eraser on the disk, or perform something equally destructive to the data before installing eDirectory.

1c Configure eDirectory and set the encryption schemes that you want on an attribute.

2 Move this server into a replica ring where you have the existing data that you want to encrypt, let the replication happen then take the old server offline.

3 Destroy any existing clear text data

Any disks (or on other media) with the clear text data on it should be securely wiped. This includes things like the clear text LDIF file used to bulk load the server, any other server that was used for replication, or tapes with old backups on them.

Through Backup and Restore

1 Setup encrypting on a new server as follows:

1a Plan in advance which attributes you want to encrypt and with what scheme.

That is, you must decide in advance which attributes you want to encrypt before uploading the data in clear text into the eDirectory.

WARNING: Once you have loaded any data into the eDirectory in the clear, you should not mark an attribute for encryption. Though you can do it, this leads to security problems listed in Note A.

1b Start with a clear install (probably including the operating system) on a freshly formatted and partitioned disk.

This is to ensure that there is no clear text data on the disk. This means you cannot just take an existing computer which has clear text data previous and re-install eDirectory. You must have thoroughly erased all traces of data from the disk. Run some kind of secure erase software, use a magnetic bulk eraser on the disk, or perform something equally destructive to the data before installing eDirectory.

1c Configure eDirectory and set the encryption schemes that you want on an attribute.

2 Restore the backed up DIB (that contains the existing clear text data) on the new server. You can backup the DIB using DIB Clone or Hot Backup.

3 Destroy any existing clear text data

Any disks (or on other media) with the clear text data on it should be securely wiped. This includes things like the clear text LDIF file used to bulk load the server, any other server that was used for replication, or tapes with old backups on them.

Changing the Scheme of the Encrypted Data

The steps require to do this using backup/restore are mentioned below:

1 Change the encryption algorithms for an attribute.

2 Take a DIB backup. You can backup the DIB using DIB Clone or Hot Backup.

3 Restore the backed up DIB to a new fresh server, and delete the old server.

4 Destroy any existing clear text data on the old server. This avoids bits and pieces of data with the old scheme still on the hard disk.

Any disks (or on other media) with the clear text data on it should be securely wiped.This includes things like the clear text LDIF file used to bulk load the server, any other server that were used for replication or tapes with old backups on them.

11.3.3 Conclusion

The scenarios listed here are not exhaustive and there might be more scenarios where this problem occurs. As long as you follow the rule, *No information that would eventually be encrypted should ever be written to the hard disk (or any other media) in the clear*, the encrypted data will be truly secure.

12 Repairing the Novell eDirectory Database

The DSRepair 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 8, “Using Novell iMonitor,” on page 199.

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.
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 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:

- Section 12.1, “Performing Basic Repair Operations,” on page 268
- Section 12.2, “Viewing and Configuring the Repair Log File,” on page 272
- Section 12.3, “Performing a Repair in Novell iMonitor,” on page 273
- Section 12.4, “Repairing Replicas,” on page 273
- Section 12.5, “Repairing Replica Rings,” on page 276
- Section 12.6, “Maintaining the Schema,” on page 278
- Section 12.7, “Repairing Server Network Addresses,” on page 281
- Section 12.8, “Performing Synchronization Operations,” on page 282
- Section 12.9, “DSRepair Options,” on page 285
- Section 12.10, “Using the Client to Repair a Database,” on page 289

12.1 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 269
- “Performing a Local Database Repair” on page 270

- "Checking External References" on page 271
- "Repairing a Single Object" on page 271
- "Deleting Unknown Leaf Objects" on page 271

12.1.1 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.

Troubleshooting specific issues and resolving them is far superior to running an unattended 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 270 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.
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.

Operation	Database Locked?	Description
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.

To perform an unattended full repair:

1. In Novell iManager, click the *Roles and Tasks* button .
2. Click *eDirectory Maintenance* > *Repair eDirectory*.
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.

12.1.2 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 Section 12.2, “Viewing and Configuring the Repair Log File,” on page 272.

1. In Novell iManager, click the *Roles and Tasks* button .
2. Click *eDirectory Maintenance* > *Repair eDirectory*.
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.

12.1.3 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* > *Repair eDirectory*.
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.

12.1.4 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* > *Repair eDirectory*.
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.

12.1.5 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* > *Repair eDirectory*.
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.

12.2 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 272
- "Setting Log File Options" on page 272

12.2.1 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 272 for more information.

1. In Novell iManager, click the *Roles and Tasks* button .
2. Click *eDirectory Maintenance* > *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.

12.2.2 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* > *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.

12.3 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* > *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 213.

12.4 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 274
- "Repairing Selected Replicas" on page 274
- "Repairing Time Stamps" on page 274
- "Designating This Server As the New Master Replica" on page 275
- "Destroying the Selected Replica" on page 276

12.4.1 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 270 for more information.

1. In Novell iManager, click the *Roles and Tasks* button .
2. Click *eDirectory Maintenance* > *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.

12.4.2 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 270 for more information.

1. In Novell iManager, click the *Roles and Tasks* button .
2. Click *eDirectory Maintenance* > *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.

12.4.3 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 Section 12.8, "Performing Synchronization Operations," on page 282 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* > *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 Timestamps and Declare a New Epoch*, then click *Next*.
6. Follow the online instructions to complete the operation.

12.4.4 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 135.

1. In Novell iManager, click the *Roles and Tasks* button .
2. Click *eDirectory Maintenance* > *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.

12.4.5 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 135.

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

2 Click *eDirectory Maintenance* > *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.

12.5 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 276
- "Repairing the Selected Replica Ring" on page 277
- "Sending All Objects to Every Server in the Ring" on page 277
- "Receiving All Objects from the Master to the Selected Replica" on page 278
- "Removing This Server from the Replica Ring" on page 278

12.5.1 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 270 for more information.

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

2 Click *eDirectory Maintenance* > *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.

12.5.2 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 270 for more information.

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

2 Click *eDirectory Maintenance* > *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.

12.5.3 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* > *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.

12.5.4 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* > *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.

12.5.5 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* > *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.

12.6 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 123.

Use the Schema Maintenance Wizard to perform the following operations:

- "Requesting Schema from the Tree" on page 279
- "Resetting the Local Schema" on page 279
- "Performing Optional Schema Enhancements" on page 280
- "Importing Remote Schema" on page 280
- "Declaring a New Schema Epoch" on page 280

12.6.1 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* > *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.

12.6.2 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* > *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.

12.6.3 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.

Previous versions of eDirectory cannot synchronize these changes.

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 perform the operation, then click *Next*.
5. Click *Optional Schema Enhancements*, then click *Next*.
6. Follow the online instructions to complete the operation.

12.6.4 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 10, "Merging Novell eDirectory Trees," on page 229 for more information.

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 perform the operation, then click *Next*.
5. Click *Import Remote Schema*, then click *Next*.
6. Follow the online instructions to complete the operation.

12.6.5 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* > *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.

12.7 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 281
- "Repairing a Server's Network Addresses" on page 282

12.7.1 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* > *Repair Server*.
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 Network Addresses*, then click *Next*.
6. Follow the online instructions to complete the operation.

12.7.2 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* > *Repair Server*.
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 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 compatible with OpenSLP, and the authentication features of OpenSLP are used.

12.8 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 283
- "Reporting the Synchronization Status on This Server" on page 283
- "Reporting the Synchronization Status on All Servers" on page 283
- "Performing a Time Synchronization" on page 284
- "Scheduling an Immediate Synchronization" on page 285

12.8.1 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* > *Repair Sync*.
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 *Sync the Selected Replica on This Server*, then click *Next*.
6. Follow the online instructions to complete the operation.

12.8.2 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* > *Repair Sync*.
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 *Report the Sync Status on This Server*, then click *Next*.
6. Follow the online instructions to complete the operation.

12.8.3 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* > *Repair Sync*.
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 *Report the Sync Status on All Servers*, then click *Next*.
6. Follow the online instructions to complete the operation.

12.8.4 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 8, "Using Novell iMonitor," on page 199 for more information.

For more information, see Section 2.8, "Synchronizing Network Time," on page 90.

1. In Novell iManager, click the *Roles and Tasks* button .
2. Click *eDirectory Maintenance* > *Repair Sync*.
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 *Time Sync*, then click *Next*.
6. Follow the online instructions to complete the operation.

12.8.5 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* > *Repair Sync*.
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 *Schedule Immediate Sync*, then click *Next*.
6. Follow the online instructions to complete the operation.

12.9 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 285
- "DSRepair Command Line Options" on page 286
- "Using Advanced DSRepair Switches" on page 288

12.9.1 Running DSRepair on the eDirectory Server

- "Windows" on page 285
- "Linux, Solaris, and AIX" on page 285

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, and AIX

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 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
```

To repair a single object when the entry id of the object is known, enter the following command:

```
ndsrepair -J <entry ID in hex>
```

To repair a particular partition or a replica, enter the following command:

```
ndsrepair -P
```

This command returns a list of all the partitions present on the server. You can choose any of the partitions to get the list of operations that can be performed.

To repair network addresses, enter the following command:

```
ndsrepair -N
```

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.

12.9.2 DSRepair Command Line Options

Option	Description
`-U`	Unattended Full Repair option. Instructs DSRepair to run and exit without further user assistance. You can view the log file after the repair has completed to determine what actions DSRepair has taken. This option is not a recommended default normal repair. Troubleshooting specific issues and resolving them is far superior to running an unattended repair.
`-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.

Option	Description
-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.
-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 serves 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 *filename*	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. It prompts the user to save or discard changes and view the log file.
`-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.
`-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.

12.9.3 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 16, "Backing Up and Restoring Novell eDirectory," on page 391 for more information.

On Linux, Solaris, and AIX, 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 285 for more information.

Switch	Description
-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.
-RC	Backs up the DIB. This option is available only on Windows.
-OT	Timestamps obituaries while performing a local database repair. All obituaries are timestamped except INHIBIT MOVE.

12.10 Using the 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 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 Client, see Section 21.1, "Using the Command Line Client," on page 516.

12.10.1 Using the DSRepair eMTool

1. Run the Client in interactive mode by entering the following at the command line:

   ```
   java -cp path_to_the_file/emboxclient.jar -i
   ```

   (If you have already put the `emboxclient.jar` file in your class path, you only need to enter `java -i`.)

   The Client prompt appears:

   ```
   Client>
   ```

2. Log in to the server you want to repair by entering the following:

   ```
   login -s server_name_or_IP_address -p port_number
   -u username.context -w password -n
   ```

   The port number is usually 80 or 8028, unless you have a Web server that is already using the port. The `-n` option opens a nonsecure connection.

   The 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 Client will indicate whether the repair is successful.

See "DSRepair eMTool Options" on page 290 for more information on the DSRepair eMTool options.

4 Log out from the Client by entering the following command:

```
logout
```

5 Exit the Client by entering the following command:

```
exit
```

12.10.2 DSRepair eMTool Options

The following tables lists the DSRepair eMTool options. You can also use the `list -t dsrepair` command in the Client to list the DSRepair options with details. See "Listing eMTools and Their Services" on page 519 for more information.

Option	Description
`rso -o -d`	Single object repair Object ID in hex Object DN
`rts`	Time synchronization
`rss`	Report synchronization status of all partitions
`rld -l -t -d -p -i -f -c -o -a -m -v`	◆ Repair local database ◆ Lock eDirectory database during entire repair ◆ Use temporary eDirectory database during repair ◆ Maintain original unrepaired database ◆ Perform database structure check ◆ Perform database structure and index check ◆ Reclaim database free space ◆ Perform tree structure check ◆ Rebuild operational schema ◆ Repair all local replicas ◆ Validate mail directories and stream files ◆ Check local references
`ufr`	Unattended full repair
`rsn -o -d`	Repair selected server's network address Object ID in hex Object DN
`ran`	Repair all network addresses
`rsr -p -d`	Repair selected replica Partition ID Partition DN
`rer`	Repair every replica

Option	Description
`ror -p -d`	Repair selected replica ring Partition ID Partition DN
`rar`	Repair replica ring, all replicas
`ssa -p -d`	Report the replica synchronization status of all servers Partition ID Partition DN
`cer`	Check external references
`rao -p -d -s -d`	Receive all objects for this replica Partition ID Partition DN Server ID Server DN
`sao -p -d -s -d`	Send all objects to every replica in the ring Partition ID Partition DN Server ID Server DN
`dne -p -d`	Repair time stamps and declare a new epoch Partition ID Partition DN
`sri -p -d`	Schedule immediate synchronization Partition ID Partition DN Server ID Server DN
`sks -p -d -s -d`	Synchronize the replica on the selected server Partition ID Partition DN Server ID Server DN
`ske -p -d`	Synchronize the replica on all servers Partition ID Partition DN
`dsr -p -d`	Destroy the selected replica on this server Partition ID Partition DN
`xsr -p -d -s -d`	Remove this server from the replica ring Partition ID Partition DN Server ID Server DN
`dnm -p -d`	Designate this server as the new master replica Partition ID Partition DN
`dul`	Delete unknown leaf objects

13 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.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 systems.

13.1 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	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. Two types of replica synchronization can occur: ◆ 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. 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.
Schema synchronization	Ensures that the schema is consistent across the partitions in the directory tree and that all schema changes are updated across the network. This process runs once every four hours by default.
Heartbeat	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. This process runs by default once every 30 minutes on every server that contains a replica of a partition.
Limber	Ensures that a server's replica pointer table is updated when that server's name or address is changed. Such changes occur when ◆ 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. 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. The limber process also checks that the tree name is correct for each server in a replica ring. Limber runs five minutes after the server boots up and then every three hours.
Backlink	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.

Process	Description
Connection management	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. 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.
Server status check	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. The server status check runs every six minutes.

13.1.1 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. Specify a name and context for the object, then click *OK*.
4. When finished, click *OK*.

Continue with one of the following sections:

- "Adding Servers to a LAN Area Object" on page 295
- "Applying WAN Policies" on page 297

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, then click *Apply*.
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 297.

7 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*.

13.1.2 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.

- "Predefined Policy Groups" on page 296
- "Applying WAN Policies" on page 297
- "Modifying WAN Policies" on page 298
- "Renaming an Existing Policy" on page 299
- "Creating New WAN Policies" on page 299

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.
ndsttyps.wmg	Provides sample policies for various eDirectory traffic types.
onospoof.wmg	Allows only existing WAN connections to be used.
opnspoof.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.
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 Section 13.2, "WAN Traffic Manager Policy Groups," on page 302.

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 Servers*, then click an NCP Server object.
4. Click *Add Policy*, then select the policy group you want.

   See "Predefined Policy Groups" on page 296 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 Section 13.3, "WAN Policy Structure," on page 316.

To understand the syntax of a WAN policy, see "Construction Used within Policy Sections" on page 319.

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 Section 13.3, "WAN Policy Structure," on page 316.

To understand the syntax of a WAN policy, see "Construction Used within Policy Sections" on page 319.

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 Section 13.3, “WAN Policy Structure,” on page 316.

To understand the syntax of a WAN policy, see “Construction Used within Policy Sections” on page 319.

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 Section 13.3, "WAN Policy Structure," on page 316.

To understand the syntax of a WAN policy, see "Construction Used within Policy Sections" on page 319.

6 Click *Apply*, then click *OK*.

13.1.3 Limiting WAN Traffic

WAN Traffic Manager comes with two predefined WAN Policy groups that limit traffic to specific hours. You can modify these policies to limit traffic to any span of hours you select. For more information, see "1-3am.wmg" on page 302 and "7am-6pm.wmg" on page 302.

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 Section 13.3, "WAN Policy Structure," on page 316.

To better understand the syntax of a WAN policy, see "Construction Used within Policy Sections" on page 319.

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*.

13.1.4 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 303.

For information about how to modify a policy, see "Modifying WAN Policies" on page 298.

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*.

13.2 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 297.

13.2.1 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.

13.2.2 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.

13.2.3 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.

13.2.4 Ipx.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.

13.2.5 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 304
- "Sample Catch All without Addresses" on page 304
- "Sample NDS_BACKLINK_OPEN" on page 304
- "Sample NDS_BACKLINKS" on page 305
- "Sample NDS_CHECK_LOGIN_RESTRICTION" on page 306
- "Sample NDS_CHECK_LOGIN_RESTRICTION_OPEN" on page 307
- "Sample NDS_JANITOR" on page 308
- "Sample NDS_JANITOR_OPEN" on page 309
- "Sample NDS_LIMBER" on page 310
- "Sample NDS_LIMBER_OPEN" on page 311
- "Sample NDS_SCHEMA_SYNC" on page 312
- "Sample NDS_SCHEMA_SYNC_OPEN" on page 313
- "Sample NDS_SYNC" on page 314

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`	`ConnectionLastUsed` is the time that eDirectory last sent a packet on this connection.
`FALSE`	`ConnectionLastUsed` 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 and 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 and 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	ConnectionLastUsed is the time that eDirectory last sent a packet on this connection.
FALSE	ConnectionLastUsed 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 and 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).
`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 and 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.
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 and 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`	`ConnectionLastUsed` is the time that eDirectory last sent a packet on this connection.
`FALSE`	`ConnectionLastUsed` 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.

13.2.6 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.

13.2.7 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.

13.2.8 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.

13.2.9 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.

13.2.10 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.

13.3 WAN Policy Structure

A WAN policy consists of three sections:

- "Declaration Section" on page 316
- "Selector Section" on page 318
- "Provider Section" on page 319

13.3.1 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 and 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	Variables defined as REQUIRED in scope can be used in multiple sections, but only once within the Declaration section. No values can be defined for a REQUIRED scope variable. Its value must come from the GetWanPolicy request.
OPTIONAL	Variables defined as OPTIONAL in scope can be used in multiple sections of a policy, but only once within the Declaration section. 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. 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.
LOCAL	Variables defined as LOCAL in scope can be used in multiple sections, but only once within the Declaration section. LOCAL scope variables exist only for a particular policy. Their values are not returned to the calling client. 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.
SYSTEM	Variables defined as SYSTEM in scope can be used in multiple sections, but only once within the Declaration section.

Type

Valid types are listed in the following table.

Type	Description
INT	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: IF TrafficType=NDS_SYNC THEN *action* END.
BOOLEAN	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.
TIME	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.
NETADDRESS	NETADDRESS scope variables must receive their values in the Selector or Provider sections. Do not assign values to NETADDRESS scope variables in the Declaration.

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 *R1* is not the same as the variable *r1*. 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.

13.3.2 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 319. See also "Provider Section" on page 319.

13.3.3 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 319.

13.3.4 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 316.

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.

14 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, while 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:

- Section 14.1, “Key Terms for LDAP Services,” on page 326
- Section 14.2, “Understanding How LDAP Works with eDirectory,” on page 328
- Section 14.3, “Using LDAP Tools on Linux, Solaris, or AIX,” on page 337
- Section 14.4, “Extensible Match Search Filter,” on page 347
- Section 14.5, “LDAP Transactions,” on page 348

For more information on LDAP, see the following Web sites:

- OpenLDAP (http://www.openldap.org/)
- An LDAP Roadmap & FAQ (http://www.kingsmountain.com/ldapRoadmap.shtml)

14.1 Key Terms for LDAP Services

- Section 14.1.1, “Clients and Servers,” on page 326
- Section 14.1.2, “Objects,” on page 326
- Section 14.1.3, “Referrals,” on page 327

14.1.1 Clients and Servers

LDAP Client— An application (for example, Internet Explorer or the Novell Import Conversion Export utility).

LDAP Server— A server where `nldap.dlm` (for Windows) or `libnldap.so` (for Linux, Solaris and AIX systems) is running.

14.1.2 Objects

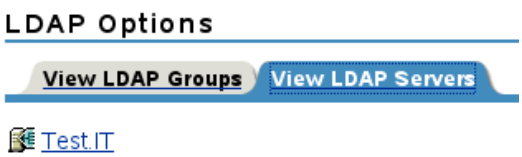
LDAP Group object— Sets up and manages the Novell LDAP properties on an LDAP server.

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.



14.1.3 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 Section 15.8, “Configuring for Superior Referrals,” on page 380.

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.

14.2 Understanding How LDAP Works with eDirectory

This section explains the following:

- "Connecting to eDirectory from LDAP" on page 328
- "Class and Attribute Mappings" on page 331
- "Enabling Nonstandard Schema Output" on page 334
- "Syntax Differences" on page 335
- "Supported Novell LDAP Controls and Extensions" on page 336

14.2.1 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 user name 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 329.

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 user name. 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 Section 15.4, "Configuring LDAP Objects," on page 356.
- 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 Step 8 and Step 9.
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 user name.

1. In Novell iManager, click the *Roles and Tasks* button .
2. Click *LDAP* > *LDAP Options*.
3. Click *View LDAP Groups*.
4. Click the name of an LDAP Group object to configure.
5. Specify the name and context of an eDirectory User object in the *Proxy User* field.
6. Click *Apply*, then click *OK*.

Using the ldapconfig Utility on Linux and 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 User Name) 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 user name 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 user name and password entered at the LDAP client.
- The eDirectory user name and password used for LDAP client access can also be used for Novell 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 user name 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 user name 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 328 for more information.
2. If users will use one proxy user or multiple eDirectory user names to access LDAP, use iManager to create these user names in eDirectory or through LDAP.
3. Assign the appropriate eDirectory rights to the user names 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.

14.2.2 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 Options* > *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*.

Class Mapping in LDAP Groups

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.

NOTE: eDirectory does not propagate class mappings in LDAP Group objects across LDAP servers. To use the same class mapping on more than one server, manually add the mapping to all LDAP group objects in your environment.

---

1. In Novell iManager, click the *Roles and Tasks* button .
2. Click *LDAP* > *LDAP Options*.
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
st stateOrProvinceName	S
certificateRevocationList;binary certificateRevocationList	ndspkiCertificateRevocationList
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.

14.2.3 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 older 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 Options*.
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*.

14.2.4 Syntax Differences

LDAP and eDirectory use different syntaxes. Some important differences include the following:

- “Commas” on page 335
- “Typeful Names” on page 335
- “Escape Character” on page 336
- “Multiple Naming Attributes” on page 336

Commas

LDAP uses commas as delimiters rather than periods. For example, a distinguished (or complete) name in eDirectory looks like this:

CN=JANEB.OU=MKTG.O=EMA

Using LDAP syntax, the same distinguished name would be

CN=JANEB,OU=MKTG,O=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 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 UID 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.

14.2.5 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 Section 15.10, "Getting Information about the LDAP Server," on page 387.

For a list of supported LDAP controls and extensions, see "LDAP Controls" (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) in the LDAP and eDirectory Integration NDK.

14.3 Using LDAP Tools on Linux, Solaris, or AIX

eDirectory includes the following LDAP tools, stored in `/opt/novell/eDirectory/bin`, to help you manage the LDAP directory server.

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/ltoolenu/data/hevgtl7k.html) in the *LDAP Libraries for C Doc*.

To perform secure LDAP tools operations, refer to "Ensuring Secure eDirectory Operations on Linux, Solaris, and AIX Systems" on page 87 and include the DER file in all command line LDAP operations that establish secure LDAP connections to eDirectory.

14.3.1 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]
```

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 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 file`	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 replog 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 debuglevel`	Sets the LDAP debugging level to debuglevel. The ldapmodify tool must be compiled with LDAP_DEBUG defined for this option to have any effect.
`-D binddn`	Uses binddn to bind to the LDAP directory. *binddn* should be a string-represented DN as defined in RFC 1779.
`-e key filename`	Files the certificate filename for SSL bind.
`-f file`	Reads a series of lines from 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. If the file is a single hyphen (-) character, then the lines are read from standard input.
`-h ldaphost`	Specifies an alternate host on which the LDAP server is running.
`-l limit`	Specifies the connection timeout (in seconds).
`-M`	Enables Manage DSA IT control (non-critical).
`-MM`	Enables Manage DSA IT control (critical).

Option	Description
-n	Shows what would be done, but does not actually modify entries. Useful for debugging in conjunction with -v.
-p *ldapport*	Specifies an alternate TCP™ port where the LDAP server is listening.
-P *version*	Specifies the LDAP version (2 or 3).
-v	Uses verbose mode with many diagnostics written to standard output.
-w *passwd*	Uses *passwd* as the password for simple authentication.
-W	Prompts for simple authentication. This option is used instead of specifying the password on the command line.
-Z	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. If a port is specified with this option, it must accept clear text connections. 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.
-ZZ	Starts TLS before binding to perform the operation. If an error occurs during the Start TLS operation, the operation is aborted. If a port is specified with this option, it must accept clear text connections. 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.

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]...
```

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 DNs 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 DNs to delete and ignores any DNs 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 file`	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 338 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).

ldapmodify

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]
```

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` *file*	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 replog record should actually be applied.

NOTE: Refer to "Common Options for All LDAP Tools" on page 338 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: 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` *file*	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` *newsuperior*	Specifies the distinguished name of the container to which the entry is moving.

NOTE: Refer to "Common Options for All LDAP Tools" on page 338 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....]
```

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` *deref*	Specifies how to handle the dereferencing of an alias. It uses the following values: ◆ 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` *searchbase*	Use *searchbase* 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` *scope*	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`.

Option	Description
`-S attribute`	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 ldap_sort for more details. ldapsearch 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 path`	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 prefix`	Specifies the URL prefix for files (default: `file://tmp/`).
`-z sizelimit`	Waits at most *sizelimit* entries for a search to complete.

NOTE: Refer to "Common Options for All LDAP Tools" on page 338 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>.....]
```

Option	Description
`list`	Lists the specified indexes. If the index is not specified, ndsindex lists all existing indexes on the server.
`add`	Creates new indexes.
`delete`	Deletes the specified indexes.
`resume`	Resumes the specified indexes from an off-line state.
`suspend`	Suspends the specified indexes to an off-line state.
`-s` *`eDirectory Server DN`*	Specifies the eDirectory Server DN.

NOTE: Refer to "Common Options for All LDAP Tools" on page 338 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
```

14.4 Extensible Match Search Filter

The LDAP 3 core protocol specification defined in RFC 2251 (http://www.ietf.org/rfc/rfc2251.txt) 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.
[":*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 and later versions support 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 and later versions.

```
(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 and later versions:

```
(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.

14.5 LDAP Transactions

eDirectory LDAP server supports clubbing of multiple update operations into a single atomic operation - also called a transaction. The support for transactions over LDAP in eDirectory is based on two Internet specifications – "LDAP Transactions" (http://www.watersprings.org/pub/id/draft-zeilenga-ldap-txn-05.txt) and "LDAP: Grouping of Related Operations" (http://www.watersprings.org/pub/id/draft-zeilenga-ldap-grouping-05.txt).

LDAP transactions allow an LDAP application to send several LDAP update operations (add, modify, delete, rename) as a group and then commit or abort this whole group of operations.

There are few entities which figure in the context of LDAP transactions:

- CreateGroupingRequest (2.16.840.1.113719.1.27.103.1) – This is LDAP extended operation which allows grouping of related operations. The extended operation carries a value – createGroupType which identifies the type of grouping requested. For LDAP transactions, the grouping type is transactionGroupingType. (2.16.840.1.113719.1.27.103.8)
- CreateGroupingResponse (2.16.840.1.113719.1.27.103.1) – This is the response of the LDAP server to the createGroupingRequest and contains 2 response fields – groupCookie and an optional createGroupValue.
- GroupingControl (2.16.840.1.113719.1.27.103.7) - This is used to indicate association of an operation to a grouping via the groupCookie which is the value carried by this control.
- EndGroupingRequest (2.16.840.1.113719.1.27.103.2) – This is another LDAP extended operation used to indicate the end of a grouping Request. In case of LDAP transactions, this indicates the settling of the transaction – resulting in a commit or an abort of the transaction.
- EndGroupingResponse (2.16.840.1.113719.1.27.103.2) – This is the response of the LDAP server to the endGroupingResponse indicating either success or otherwise to the LDAP client.

Following is the sequence of requests and responses exchanged between the LDAP server and the LDAP client in an LDAP transaction:

- If a client wants to send a number of LDAP operations to be processed by a server in an atomic operation, i.e., transaction, it should first send a createGroupingRequest, with a createGroupType of transactionGroupingType and no createGroupValue.
- If the eDirectory server is capable of handling transactions, it sends back a success result code, with a groupingCookie, which uniquely identifies the grouping requested by the client. Otherwise, the server shall return a non-successful result code indicating the reason for the failure to the client.
- If the client receives success result code from the server, it then attaches a GroupingControl, which includes the groupingCookie returned by the server, to subsequent update operations to indicate that they are to be processed as part of a single transaction. If the server is willing and able to process the update operation as part of the transaction, the server shall return success and put this request in a queue. If the server is unwilling or unable to process the update operation as part of the transaction, the server shall return a non-successful result code indicating the reason for the failure to the client.
- After the client has sent all the update operations accompanied by the grouping control to the server, the client sends an endGroupingRequest with the groupingCookie to the server to indicate that it wants to settle the transaction. The absence of endGroupValue indicates a commit request where as presence of an empty endGroupValue indicates abort request.
- The server applies all the pending operations in one transaction. If it succeeds, it shall return success. Otherwise, it shall return a non-successful result code.
- If at any time during the above exchange between the client and server, the server is unwilling or unable to continue the specification of a transaction, the server issues an endGroupingNotice (2.16.840.1.113719.1.27.103.4). Subsequent use of cookie by the client shall result in a response containing a non-success result code.

The support for LDAP transactions is indicated by the presence of the transactionGroupingType in the supportedGroupingTypes attribute of the rootDSE entry.

The LDAP transaction implementation in eDirectory is based on a dated version of the LDAP transaction specification. The latest revision of the LDAP transactions draft as of this writing is available at "Lightweight Directory Access Protocol (LDAP) Transactions" (http://tools.ietf.org/html/rfc5805).

14.5.1 Limitations

The LDAP transactions feature has the following limitations:

- All the objects affected by the operations grouped as a transaction need to be hosted locally on the server. None of these operations should require the LDAP server to chain to another server.
- Schema modifications and Modify DN operation (Subtree move?) is not allowed to be grouped in an LDAP transaction.
- Passwords and attributes with stream syntax cannot be added as part of an LDAP transaction.
- Nesting of one transaction within another is not supported.

15 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 8.8 SP7 Installation Guide*.

This section explains the following:

- Section 15.1, "Loading and Unloading LDAP Services for eDirectory," on page 351
- Section 15.2, "Verifying That the LDAP Server Is Loaded," on page 352
- Section 15.3, "Verifying That the LDAP Server Is Running," on page 353
- Section 15.4, "Configuring LDAP Objects," on page 356
- Section 15.5, "Refreshing the LDAP Server," on page 362
- Section 15.6, "Authentication and Security," on page 363
- Section 15.7, "Using the LDAP Server to Search the Directory," on page 371
- Section 15.8, "Configuring for Superior Referrals," on page 380
- Section 15.9, "Persistent Search: Configuring for eDirectory Events," on page 385
- Section 15.10, "Getting Information about the LDAP Server," on page 387
- Section 15.11, "Auditing LDAP Events," on page 389

For information on LDAP tools, see the LDAP Tools NDK (http://developer.novell.com/documentation/cldap/ltoolenu/data/hevgtl7k.html).

15.1 Loading and Unloading LDAP Services for eDirectory

To load LDAP Services for eDirectory, enter the following commands:

Server	Command
Windows	In the DHost (NDSCONS) screen, click *nldap.dlm* > *Start*.
Linux, Solaris, or AIX	At the Linux, Solaris, or AIX prompt, enter: `/opt/novell/eDirectory/sbin/nldap -l`

To unload LDAP Services for eDirectory, enter the following commands:

Server	Command
Windows	In the DHost (NDSCONS) screen, click *nldap.dlm* > *Stop*.
Linux, Solaris, and AIX	In the DHost remote management page, to unload LDAP, click the *LDAP v3 for Novell eDirectory 8.8* action icon to stop. or At the Linux, Solaris, or AIX prompt, enter: `/opt/novell/eDirectory/sbin/nldap -u`

15.2 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 Section 15.3, "Verifying That the LDAP Server Is Running," on page 353.

On Windows

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 8.8*.

   The Module Information section displays `nldap.nlm` in the filename field.

On Linux and UNIX

To verify if the LDAP Server is running, run the following command:

```
ndstrace -c modules | grep nldap
```

If the LDAP server is not loaded or running, an error appears stating that the nldap module is not loaded.

You can also use the following options:

- To check if LDAP server is running and listening on the SSL port, run the `nldap -s` command.
- To check if LDAP server is running and listening on the TCL port, run the `nldap -c` command.

These will list all the instances of eDirectory running with out any error. If the LDAP server is not loaded and not listening on either of the ports, the above commands display the error -255 (ensure that LDAP Server is running).

15.3 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 353
- "Verifying That The LDAP Server Is Running" on page 354
- "Verifying That A Device Is Listening" on page 355

15.3.1 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 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.

15.3.2 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`.

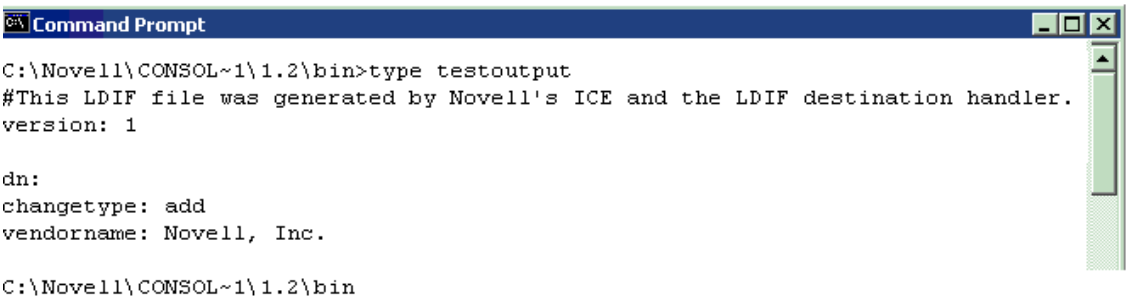
   Search the rootDSE. Include parameters that identify the source handler and the export handler. For example enter

   ```
   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 Section 6.1, “Novell Import Conversion Export Utility,” on page 147. For information specific to LDAP source handlers, see “LDAP Source Handler Options” on page 160. For information specific to LDIF destination handlers, see “LDIF Destination Handler Options” on page 159.
3. View results of the ICE command.



The example (Step 2 and Step 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 149.

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 168.

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`).

15.3.3 Verifying That A Device Is Listening

Verify that a device is listening on port 389.

For Windows, Linux, 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 204 and "Configuring Trace Settings" on page 210.

For information on LDAP requests, see "Communicating with eDirectory through LDAP" in the *Novell eDirectory 8.8 SP7 Installation Guide*.

15.4 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 8.7.3 SP9 with an LDAP server object in eDirectory 8.8 SP5.

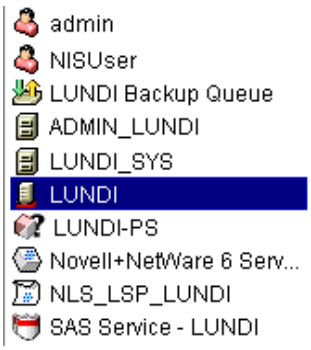
The amount of common information held in an LDAP Group object is limited. LDAP doesn't need to read many attributes because the data contained in the attributes is incredibly common. Many LDAP servers will need to use the same data. Without a common or shared Group object, you would have to replicate that data across each LDAP server.

The LDAP server object allows more server-specific configuration options and data than the LDAP Group object allows.

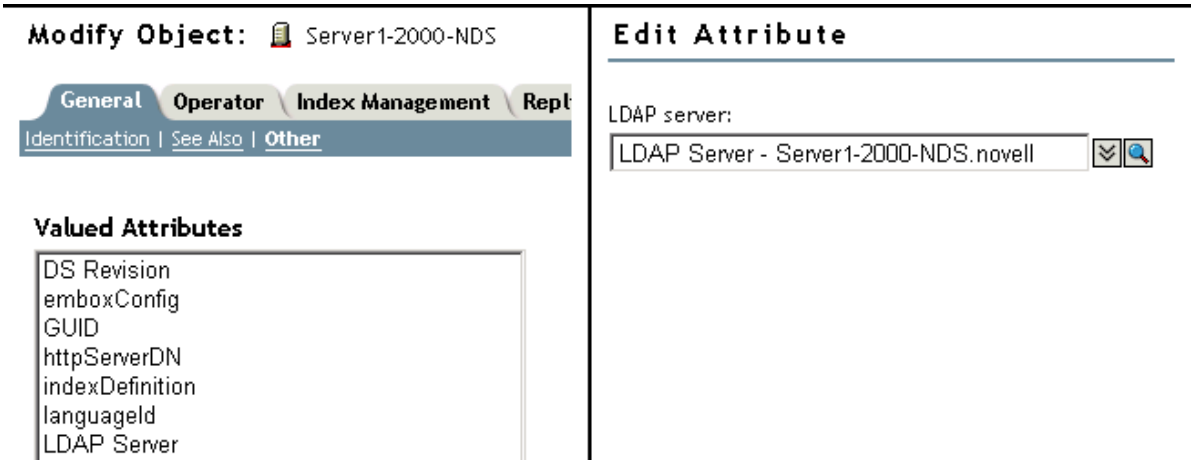
Both objects have DN-syntax attributes that point to each other.

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.

15.4.1 Configuring LDAP Server and LDAP Group Objects on Linux, Solaris, AIX Systems

The LDAP configuration utility is ldapconfig. You can use ldapconfig on Linux, Solaris, and AIX 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, and AIX 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, and AIX systems:

```
ldapconfig [-t tree_name | -p host_name[:port]] [-w password] [-a admin_FDN] -s
attribute=value,...
```

Parameter	Description
`-t treename`	Name of the eDirectory tree where the component will be installed.
`-p hostname`	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

Parameter	Description
`get | -V`	Lets you view all LDAP server/group attributes.
`get | -v attribute list`	Displays the current values of the attributes in the attribute list.
`set | -s attribute-value pairs`	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.
attribute	A configurable LDAP server or group attribute name. For more information, see "Attributes on the LDAP Server Object" on page 358 and "Attributes on the LDAP Group Object" on page 362.

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.

Attribute	Description
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. If the user has the administrator rights on the LDAP server object, the searchSizeLimit value is not considered.
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. If the user has the administrator rights on the LDAP server object, the searchTimeLimit value is not considered.
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.
ldapTLSVerifyClientCertificate	Enables or disables verification of the client certificate for a TLS operation through LDAP.
ldapNonStdAllUserAttrsMode	Enables or disables the non standard, all user, and operational attributes.

Attribute	Description
ldapBindRestrictions	Enables the LDAP bind restrictions and cipher level on the LDAP client connections. This attribute can be used to control the client connections. Using iManager you can set any of the following four LDAP bind restrictions: ◆ NONE - This is enabled by default. This option will enable both anonymous simple bind and non-anonymous simple bind. The value of this option is 0. ◆ Disallows anonymous simple bind - Setting this value will disable the anonymous simple bind. Non-anonymous simple bind will be enabled. Value 1. ◆ Disallows non-anonymous simple bind - This option will disable non-anonymous simple bind. Value 2. ◆ Disallows anonymous simple bind and non-anonymous simple bind - This option will disable anonymous simple bind and non-anonymous simple bind. Value 3. **NOTE**: Disabling non-anonymous simple bind will enforce appropriate grace login limits. In addition to the above options, you can set an additional cipher level also using the same attribute. Using iManager you can choose the following options: ◆ Use Cipher High - This will use a cipher level larger than 128-bit encryption, and some cipher suites with 128-bit keys. Value 48. ◆ Use Medium Cipher - This will use a cipher level of 128-bit encryption. Value 32. ◆ Use Low Cipher - This will use 64 or 56-bit encryption, but excluding export cipher suites. Value 16. The default is Export with a Cipher level including 40 and 56-bit encryption. For more information on the combination values of ldapbindrestrictions and cipher levels that can be used, refer to Table 15-1.
ldapChainSecureRequired	This is a boolean attribute. If enabled, chaining to other eDirectory will be over secure NCP. By default, the attribute is disabled.

Attribute	Description

ldapInterfaces	A multi-valued SYN_CI_STRING attribute used to store LDAP URLs on which LDAP server listens (on both cleartext and secure ports). This attribute is useful in configuring multiple instances, that requires each instance of the eDirectory server to listen on a specific interface. The attribute can be configured with the IP addresses and port numbers in the LDAP URL format. The LDAP server listens on these IP addresses and ports.
	
	Example: To configure an instance of LDAP server to listen on two IP addresses (on both clear text and secure port) of a machine, the attribute can be populated as follows:
	
	`ldap://192.168.1.1:389`
	
	`ldaps://192.168.2.1:636`
	
	`ldap://192.168.100.101:389`
	
	`ldaps://192.168.100.101:636`
	
	The default value of `ldapInterfaces` attribute is `ldap://`. This means LDAP server listens on all the IP addresses configured in the machine.
	
	The server continues to listen on all the interfaces in the machine if cleartext or TLS ports in the LDAP object are not unchecked.

ldapStdCompliance	eDirectory LDAP server by default does not return the sub-ordinate referrals for ONE level search. To enable this, you need to turn on ldapStdCompliance with a value 1. Setting this value will make the LDAP server return the sub-ordinate referrals for ONE level search.
ldapChainSecureRequired	This is a boolean attribute. If this is enabled, the chaining to other eDirectory will be over secure NCP. By default, the attribute will be disabled.
ldapEnablePSearch	Specifies whether or not the persistent search feature is enabled on the LDAP server. Values= true, false
ldapMaximumPSearchOperations	An integer value that limits the number of concurrent persistent search operations possible. A value of 0 specifies unlimited search operations.
ldapIgnorePSearchLimitsForEvents	Indicates whether size and time limits should be ignored after the persistent search request has sent the initial result set. Values= true, false 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.

Table 15-1 *Combination Values of ldapbindrestrictions and Cipher Levels*

ldapbindrestriction	Cipher Level	Combination Value
None	None	0
	High	48
	Medium	32
	Low	16
Disallows anonymous simple bind	None	1
	High	49
	Medium	33
	Low	17
Disallow local bind	None	2
	High	50
	Medium	34
	Low	18
Disallow anonymous simple bind and unbind	None	3
	High	51
	Medium	35
	Low	19

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 364. 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, referralIncludeFilter, referralExludeFilter as well as how LDAP servers process LDAP referrals, see "Using Referrals" on page 372.

15.5 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 Options > 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`.

15.6 Authentication and Security

This section contains information on the following:

- "Requiring TLS for Simple Binds with Passwords" on page 364
- "Starting and Stopping TLS" on page 364
- "Configuring the Server for TLS" on page 365
- "Configuring the Client for TLS" on page 366
- "Exporting the Trusted Root" on page 367
- "Authenticating with a Client Certificate" on page 367

- "Using Certificate Authorities from Third-Party Providers" on page 367
- "Using SASL" on page 369

15.6.1 Requiring TLS for Simple Binds with Passwords

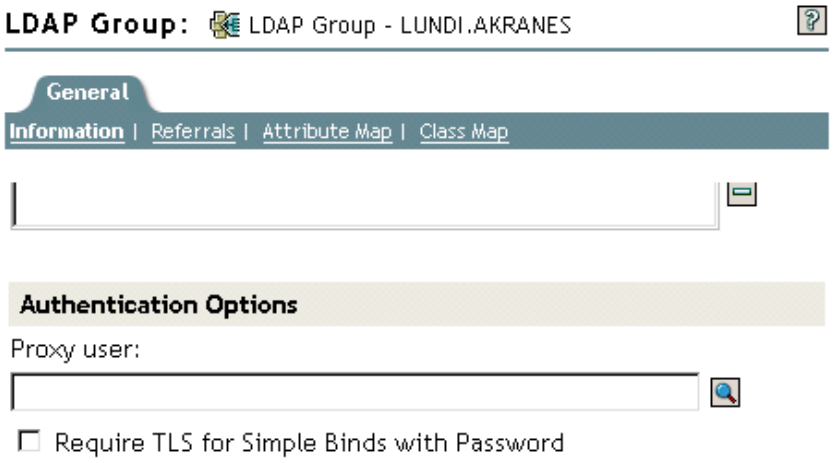
Secure Socket Layer (SSL) 3.1 was released through Netscape. IETF took ownership for that standard by implementing Transport Layer Security (TLS) 1.0. TLS 1.0 has backward compatibility with SSLv2 and v3.

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 Options* > *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.



5. Click *Apply*, then click *OK*.

15.6.2 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.

15.6.3 Configuring the Server for TLS

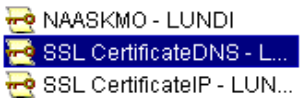
When a TLS session is instantiated, a handshake occurs. The server and the client exchange data. The server determines how the handshake occurs. To establish that the server is legitimate, the server always sends the server's certificate to the client. This handshake guarantees to the client that the 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.

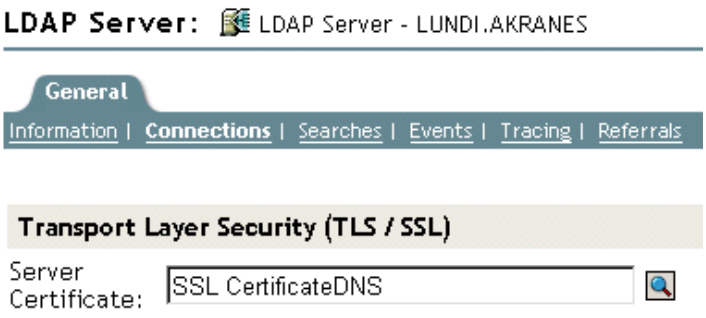
Value	Description
0	Off. During a handshake, the server provides a certificate to the client. The server never requires the client to send a certificate. The client can use or ignore the certificate. A secure session is established.
1	During the handshake, the server provides a certificate to the client 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 Section 15.5, "Refreshing the LDAP Server," on page 362. ConsoleOne and Novell iManager automatically refresh the server.

15.6.4 Configuring the Client for TLS

An LDAP client is an application (for example, 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. IE has one way, and ICE has another way. These are different LDAP clients. Each client has its method for locating the certificates that it trusts.

15.6.5 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/crt33/crtadmin/data/a2ebopb.html#a2ebopd) in the *Novell Certificate Server 3.3 Administration Guide*.

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.

15.6.6 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 Options*.
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*.

15.6.7 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.

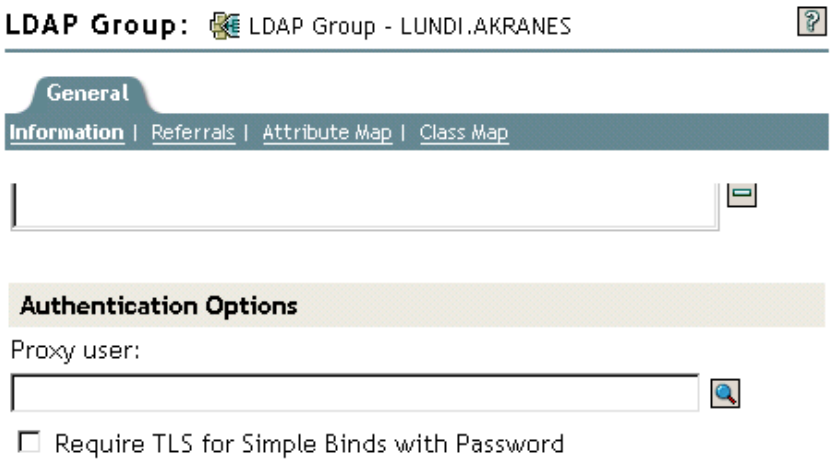
LDAP Services for eDirectory 8.8 supports multiple certificate authorities. Novell's tree CA is just one certificate authority. The LDAP server might have other CAs (for example, from VeriSign*, an external company.) This additional CA is also a trusted root.

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.

15.6.8 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 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 Linux 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 *Directory Administration* > *Create Object*, then create a proxy user (for example, LDAPProxy).
3. Assign a null password to that user.
4. (Optional) Assign the proxy user rights to specified directories.

5 Click *LDAP* > *LDAP Options* > *View LDAP Groups* > the LDAP Group object.

6 In the *Proxy User* field, click the *Browse* button, browse to and select the LDAPProxy user, then click *OK*.

15.6.9 Using SASL

Simple Authentication and Security Layer (SASL) is a mechanism for adding authentication support and data security services to connection-based protocols through different mechanisms. It presents a well-formed interface between the protocols and mechanisms. In addition, it provides a protocol for securing subsequent protocol exchanges within a data security layer along with data integrity, data confidentiality, and other services.

SASL is designed to allow new protocols to reuse the existing mechanisms without requiring redesign of the mechanisms, and it also allows existing protocols to make use of new mechanisms without the redesign of protocols. To use SASL, each protocol provides a method for identifying which mechanism is to be used, a method for exchange of mechanism-specific server-challenges and client-responses, and a method for communicating the outcome of the authentication exchange.

SASL mechanisms are named by strings, consisting of uppercase letters, digits, hyphens, and underscores. SASL mechanism names must be registered with the Internet Assigned Numbers Authority (IANA).

If a server supports the requested mechanism, it initiates an authentication protocol exchange. This consists of a series of server challenges and client responses that are specific to the requested mechanism. During the authentication protocol exchange, the mechanism performs authentication, transmits an authorization identity from the client to server, and negotiates the use of a mechanism-specific security layer. If the use of a security layer is agreed upon, then the mechanism must also define or negotiate the maximum cipher-text buffer size that each side is able to receive.

The LDAP server supports the following mechanisms:

- DIGEST-MD5
- EXTERNAL
- NMAS_LOGIN
- GSSAPI

These mechanisms are installed on the server during an eDirectory installation or upgrade. However, on Linux and UNIX, the nmasinst utility must be used to install the NMAS methods.

As specified above, 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 are the registered mechanisms, the correct naming conventions must be used to make use of them.

The LDAP bind protocol allows the client to use various SASL mechanisms for authentication. When the application uses the LDAP bind API, it must choose either the simple bind and supply a DN and password, or choose the SASL bind and supply the SASL mechanism name and the associated SASL credentials required by the mechanism.

DIGEST-MD5

LDAP supports the DIGEST-MD5 mechanism through the bind request. Instead of requesting an LDAP simple bind (DN and clear-text password), you request an LDAP SASL bind by providing the DN and the MD5 credentianls. The DIGEST-MD5 mechanism does not require TLS. The LDAP server supports DIGEST-MD5 over clear and secure connections.

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 tries to sniff this connection, the password cannot be detected. However, the entire connection can be spoofed or hijacked.

This mechanism is an LDAP SASL bind (not a simple bind). Therefore, the LDAP server accepts these requests, even if you selected the *Require TLS for Simple Binds with Passwords* check box during installation.

EXTERNAL

The EXTERNAL mechanism informs the LDAP server that the user DN and credentials have already been supplied to the server. Therefore, the DN and credentials do not need to come across in the bind request.

The LDAP bind request uses the SASL EXTERNAL mechanism to instruct the server to do the following:

- Ask an EXTERNAL layer what the credentials were
- Authenticate the user with those credentials and user

After this is done, a secure handshake occurs. The LDAP server requests credentials from the client and the client passes them to the server, then the server receives the certificate that was passed from the client, passes the certificate to the NMAS module, and authenticates 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 online documentation (http://www.novell.com/documentation/nmas33/).

Even if the client sends an EXTERNAL mechanism, the LDAP server could fail the request.The following could be possible 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.

NMAS_LOGIN

Novell Modular Authentication Service (NMAS) is a development framework that allows you to write applications that authenticate to the network using various login and authentication methods. The NMAS framework allows you to design a flexible and expandable login and authentication system using modular plug-in methods that leverage Novell International Cryptographic Infrastructure (NICI) and Novell Directory Services (eDirectory).

The NMAS_LOGIN mechanism provides the LDAP server with the biometrics capability of NMAS. For more information, see the Novell Modular Authentication Services NDK (http://developer.novell.com/documentation/nmas/index.html?page=/documentation/nmas/nmas_enu/data/bktitle.html).

GSSAPI

The GSSAPI mechanism enables a Kerberos user to authenticate to an eDirectory server using a ticket, without needing to enter a separate LDAP user password. This functionality is targeted at LDAP application users in environments that already have the Kerberos infrastructure in place. Such users must be able to use the Kerberos server-issued tickets to authenticate to the LDAP server without providing a separate LDAP user password.

For information on configuring GSSAPI, refer to Appendix E, "Configuring GSSAPI with eDirectory," on page 563.

15.7 Using the LDAP Server to Search the Directory

This section contains information on the following:

- "Setting Search Limits" on page 371
- "Using Referrals" on page 372
- "Searching Filtered Replicas" on page 379

15.7.1 Setting Search Limits

The following attributes on the LDAP server object control how the LDAP server searches the Directory:

- Search Entry Limit

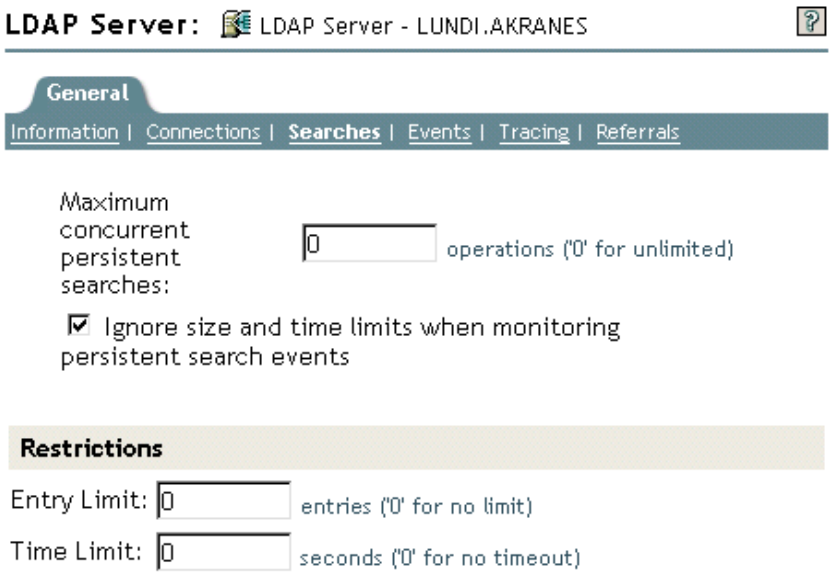
  Limits the size of a search. The default is 0, for no limit on size. So that the LDAP server isn't overloaded, you can limit the number of entries that the LDAP server returns from a search request.

  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.



1. In Novell iManager, click the *Roles and Tasks* button .
2. Click *LDAP* > *LDAP Options* > *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.

15.7.2 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:

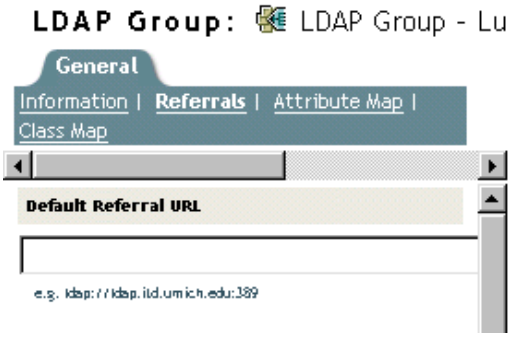
- The client contacts a server that has the entry and can perform the desired operation
- The LDAP server returns an error indicating that the entry doesn't exist
- The LDAP server indicates that no more referrals can be followed

A 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 8.8 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

A 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.

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. With eDirectory 8.8 you can 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:

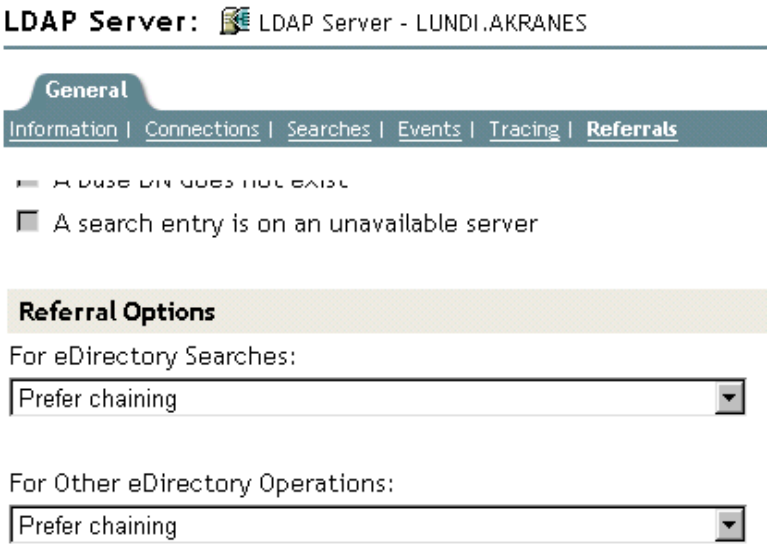
- "Prefer Chaining" on page 375 (the default option)

- "Prefer Referrals" on page 375
- "Always Refer" on page 376

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 374.

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 Section 15.9, "Persistent Search: Configuring for eDirectory Events," on page 385.

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 in 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).

Referral Filtering

If you have multiple replica servers running in a tree and have configured LDAP server(s) to return referrals using the Prefer Referrals/Always Refer option, then the LDAP server will return referrals if the object identified by DN in the requested operation is not present locally. In such a case, LDAP client sends a request to the server, and the server returns a referral list of all the LDAP servers holding that object. Using this referral list, LDAP clients will follow any of these referrals to perform the operation. If the client chooses to follow the referral to a resource-starved server or a server that is located across a slow link, clients would see a slow response from the server. This in turn affects the performance of the LDAP client. Since LDAP application developers will not have complete knowledge about the servers and network configurations, the solution for this problem is to provide a referral filtering mechanism at the LDAP server to return the referrals of specific server(s). Administrators would have the requisite knowledge, e.g. the nature of LDAP servers in the network and network link speeds to make appropriate configuration of referral filtering.

Set up the referral filter on the LDAP Group object using the attributes “referralIncludeFilter” and “referralExcludeFilter”. Setting these filters in these attributes will be applicable to all the LDAP servers belonging to this LDAP Group object. The LDAP server will return all the LDAP referrals matching with the referralIncludeList filter and drop the ones that match the referralExcludeFilter filter.

If only referralIncludeFilter is specified, the LDAP referrals which match the referralIncludeFilter values will be returned to the LDAP clients and all other referrals will be excluded from the referral list. Similarly, if only referralExcludeFilter is specified, the LDAP referrals which do not match the referralExcludeFilter values will be returned to the LDAP clients. If both filters exist and the referral does not match any of these filters, it will be excluded.

If all available referrals are disallowed by the filter, the server will behave as if no referrals are available and return LDAP_OTHER (80), which some client tools report as “Unknown error.” After adding or modifying these filter attributes, if the LDAP server is not refreshed, changes will take place after the subsequent automatic refresh.

Currently, adding or modifying these filter attributes can be done only with the tab in ConsoleOne® and iManager.

Format to Specify LDAP Referral Filtering —The LDAP referral filter format is a simple IP address format:

`[ldap://] | [ldaps://] IPAdress[:port]`

Here, specifying the clear text port or TLS port will be same as pre-pending `ldap://` or `ldaps://` strings. If neither `ldap` or `ldaps` is specified, the match filter is applicable for both clear text as well as TLS referrals.

Examples:

Examples	Description
`1.2.3.4`	# matches both LDAP and LDAPS referrals on any port
`1.2.`	# matches all IP addresses of 1.2.X.Y
`1.2.3.`	# matches all IP addresses of 1.2.3.Y
`ldap://` or `ldap://*`	# matches all the clear text port LDAP referrals
`ldaps://` or `ldap://*`	# matches all the ssl port LDAP referrals
`*`	# matches all
`ldaps://5.6.7.8:636`	# matches for SSL port 636 on IP addresses 5.6.7.8

These filter attributes (`referralIncludeFilter` and `referralExcludeFilter`) are multi-valued. You can choose as many matching filters as you need.

Example Scenarios

- To make an LDAP server return only referrals with the IP address 1.2.X.Y where X = {0 to 255} and Y = {0 to 255} and exclude all others, enter the following:

  referralIncludeFilter = { 1.2 }

- To make an LDAP server return referral, that exclude all the referrals that match IP address 164.99.X.Y, where X is not equal to 100 and match 164.99.100.Y, enter the following:

  ```
  referralIncludeFilter = { 164.99.100., "*"}
  ```

  ```
  referralExcludeFilter = { 164.99. }
  ```

  Here, even though the IP address 164.99.100.Y matches `referralExcludeFilter`, since these IP addresses have more matched fields, these referrals will be returned to the LDAP clients.

  NOTE: While specifying a partial IP address, the trailing "." can be omitted.

- To make an LDAP server return only clear text port referrals and drop SSL port referrals, enter the following:

  ```
  referralIncludeFilter = { "ldap://" }
  ```

  OR

  ```
  referralExcludeFilter = { "ldaps://" }
  ```

- To make an LDAP server return from a set of IP addresses and drop all other IP address referrals, enter the following:

  ```
  referralIncludeFilter = { 1.2.3.4, 2.3.4.5:389, 3.4.5.6:636, ldaps://4.5.6.7 }
  ```

  ```
  referralExcludeFilter = { "*" }
  ```

  NOTE: Here, `referralExcludeFilter` is not required. Any populated `referralIncludeFilter` implies to exclude all others.

- There are two filters, as follows:

  ```
  referralIncludeFilter = { 1.2.3.4 }
  ```

`referralExcludeFilter = { 2.3.4.5 }`

A referral with IP address 3.4.5.6 will be excluded as it does not match the referralInclude filter, even though it does not match the `referralExcludeFilter` as well.

Invalid Filters —The following filters are not supported.

".2.3.4" or "*.2.3.4" will not match the IP addresses *X*.2.3.4.

"2.3.4*" will not match the IP addresses like 2.3.41 or 2.3.42.

DNS names like sever1.mydomain.com, or *.mydomain.com are not supported. Adding the port ranges to the filters like allow referral IP address on the port start-to-end is not supported.There are no validation checks done before adding these filter values to these attributes. But in case of an invalid filter, the LDAP server will ignore those filters and log the information into `ndsd.log` file.

Known Issues —The LDAP rootDSE search returns altServers if there are any replica servers in the LDAP URL format. These URLs do not get filtered using this mechanism.

No Support for ManageDsaIT

In LDAP Services for eDirectory 8.8, 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 8.8 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.

15.7.3 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.

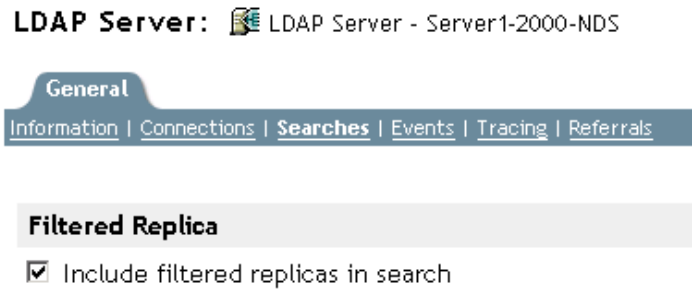
Because data in a filtered replica is incomplete, an LDAP search could produce constrained results. Therefore, by default an LDAP search request does not examine filtered replicas.

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 Options*.
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*.



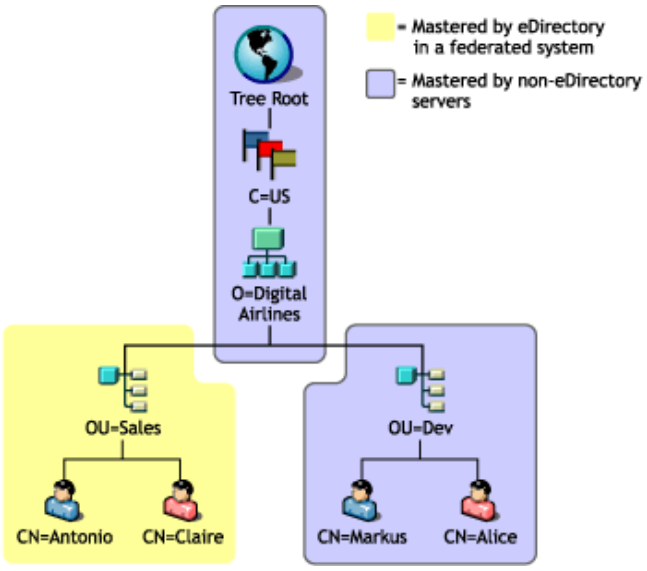
15.8 Configuring for Superior Referrals

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 8.8 has the capability to return referrals to a superior DSA in a federated tree.

15.8.1 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.

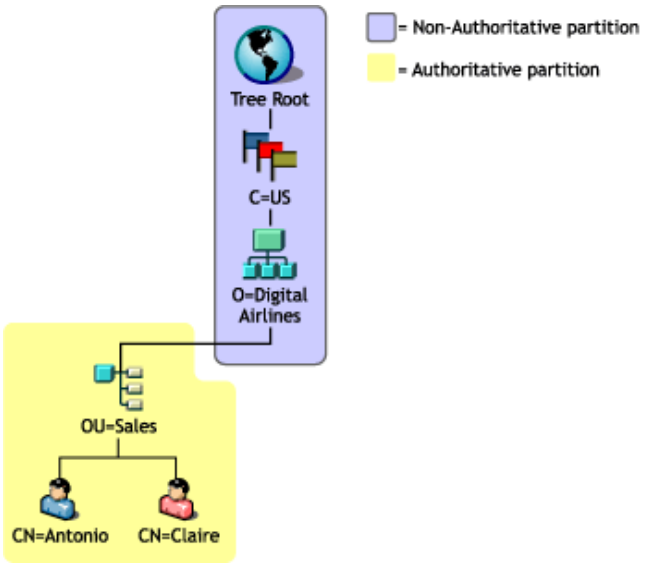
In this scenario, the Root, C=US, and O=Digital Airlines objects are held on the eDirectory server in a nonauthoritative area.

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.

15.8.2 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 380.



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 8.8 requires that all nonauthoritative partitions must be contiguous and held in local replicas.

15.8.3 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).

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 8.8, 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.

15.8.4 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). 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.

15.8.5 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 affectsMultipleDSAs should be returned.
- Extended

15.8.6 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.

15.9 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 article "Persistent Search: A Simple LDAP Change Notification Mechanism" (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.

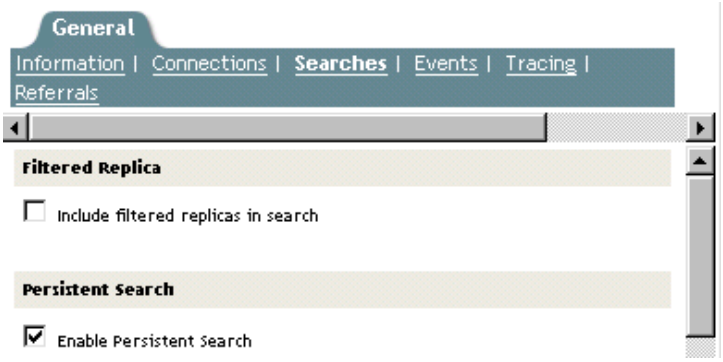
For more information, see "Understanding and Using Persistent Search in Novell eDirectory" (http://support.novell.com/techcenter/articles/dnd20030204.html).

15.9.1 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 *Directory 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.

4 Click *OK*, then click *Searches* on the *General* tab.



5 Enable persistent searches.

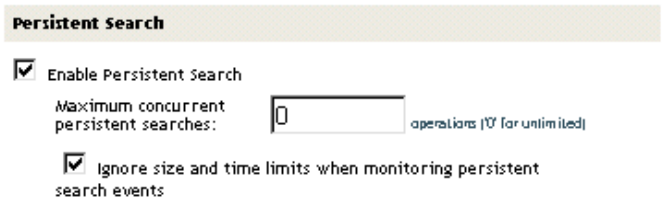
By default, the *Enable Persistent Search* check box is checked. To disable and prevent persistent searches on this server, uncheck the check box.

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.

---

6 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.



7 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.

8 Click *Apply*, then click *OK*.

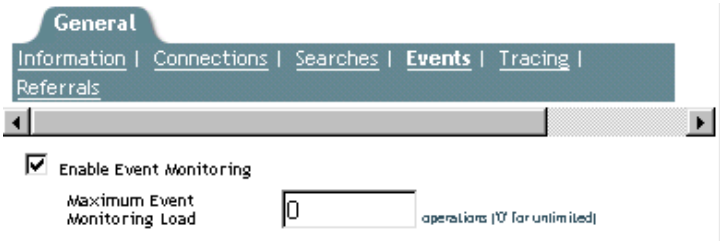
15.9.2 Controlling Use of the Monitor Events Extended Operation

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

2 Click *LDAP > LDAP Options*.

3 Click *View LDAP Servers*, then click the name of an LDAP server.

4 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*.

15.10 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 scope is base (`-s base`). Also note 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)
- LDAP Classes for Java (http://developer.novell.com/documentation/jldap/jldapenu/data/bktitle.html)

For information on LDAP search filters, see LDAP Search Filters (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.

15.11 Auditing LDAP Events

LDAP auditing enables the applications to monitor/audit LDAP operations such as Add, Modify, Search, and so on, and to fetch useful information from the LDAP server such as the connection information, the client IP to which the server was connected when LDAP operation happened, the message ID, the result code of the operation, and so on.

For more information on auditing LDAP events, refer to the LDAP Event Services (http://developer.novell.com/documentation/ldapover/ldap_enu/data/ag7bleo.html).

16 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.

The eDirectory Backup Tool enables you 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 402.
- **Lets you back up related files.** You can back up files on the server that are related to the database, such as NICI security files, 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 Section 16.3, "Using Roll-Forward Logs," on page 404.
- **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 Section 18.5, "Upgrading Hardware or Replacing a Server," on page 498.
- **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 DSBK Client.

The eDirectory Backup Tool 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.

For OES 2 Linux, you can back up eDirectory using Novell Storage Management Services. SMS provides target service agent (TSA) for backing up eDirectory. TSA for the eDirectory services eDirectory targets and provides an implementation of the SMS APIs for the Directory trees. Applications can make use of its features by writing these APIs.

TSANDS supports the following features that backup applications can take advantage of:

- Filters that can be applied to the eDirectory objects.
- Selective restores eDirectory objects from the backed up data.
- Ability to rename a particular set of resources.
- Support for incremental and differential backups based on the eDirectory modification date.
- Formats data in a SIDF and therefore any SIDF-compliant software can interpret the data.

For more information on TSANDS usage, refer to the TSANDS man page.

This chapter contains the following topics:

- Section 16.1, "Checklist for Backing Up eDirectory," on page 392
- Section 16.2, "Understanding Backup and Restore Services," on page 394
- Section 16.3, "Using Roll-Forward Logs," on page 404
- Section 16.4, "Preparing for a Restore," on page 408
- Section 16.5, "Backing Up and Restoring NICI," on page 411
- Section 16.6, "Using DSBK," on page 412
- Section 16.7, "Recovering the Database If Restore Verification Fails," on page 426
- Section 16.8, "Scenarios for Backup and Restore," on page 429
- Section 16.9, "Disaster Recovery Plan using DSBK," on page 435

16.1 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 139.

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.

The Backup Tool 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 Section 16.3, "Using Roll-Forward Logs," on page 404.

  - 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, refer to the Section 16.3.2, "Location of the Roll-Forward Logs," on page 406. 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.

- ❒ If you use NICI, ensure that your eDirectory backups include NICI security files as eDirectory requires the same NICI files to open the DIB and read the encrypted data.

  For more information about NICI security, see the *Novell International Cryptographic Infrastructure 2.7 Administration Guide* (http://www.novell.com/documentation/nici27x/nici_admin_guide/data/a20gkue.html) and refer to Section 16.5, "Backing Up and Restoring NICI," on page 411.

- ❒ For multiple-server trees, if you are using the Backup Tool 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 Tool Does a Restore" on page 397.

- ❒ 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 Section 18.5, "Upgrading Hardware or Replacing a Server," on page 498.

- ❒ 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 Tool Does a Restore" on page 397, Section 16.3, "Using Roll-Forward Logs," on page 404, and Section 16.7, "Recovering the Database If Restore Verification Fails," on page 426.

  For information on replication, see Section 1.6, "Replicas," on page 55 and Chapter 5, "Managing Partitions and Replicas," on page 135.

- ❒ 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 DSBK 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.

  DSBK 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 DSBK, see Section 16.6, "Using DSBK," on page 412.

- ❒ (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 eMBox 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.

  iManager lets you do backup and restore tasks remotely, outside the firewall, but it does not support cold backup and advanced tasks.

  eMBox 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 configuringeMBox, refer to the Section 21.3, "Using the eMBox Client for Backup and Restore," on page 525.

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 402.

- ❒ Regularly test your disaster recovery strategy to make sure it meets your goals.

16.2 Understanding Backup and Restore Services

- "About the eDirectory Backup Tool" on page 395

- "What's Different between Backup and Restore in DSBK and TSA for NDS Backup" on page 395
- "Overview of How the Backup Tool Does a Restore" on page 397
- "Format of the Backup File Header" on page 398
- "Format of the Backup Log File" on page 401
- "Using DSMASTER Servers as Part of Disaster Recovery Planning" on page 402
- "Transitive Vectors and the Restore Verification Process" on page 404

16.2.1 About the eDirectory Backup Tool

The Backup Tool 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.

NOTE: 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 Section 16.3, "Using Roll-Forward Logs," on page 404.

The Backup Tool does not back up all the objects in eDirectory at once, but only backs up 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 between Backup and Restore in DSBK and TSA for NDS Backup" on page 395.

The 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 Novell eDirectory Partner Products (http://www.novell.com/partnerguide/section/466.html).

For a description of the format for the backup files and log files that the Backup Tool creates, see "Format of the Backup Log File" on page 401 and "Format of the Backup File Header" on page 398.

16.2.2 What's Different between Backup and Restore in DSBK and TSA for NDS Backup

In previous versions of eDirectory, backup and restore was focused on backing up the tree, object by object.

The Backup Tool in eDirectory 8.7 introduced a completely new focus and new architecture. It's server-centric, not tree-centric, and 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.

For more comparison information, see the following table.

Issue	Legacy TSA for NDS Backup	Backup Tool "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 *Novell eDirectory 8.6 Administration Guide*.	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 402.
Speed	N/A	Significantly improved. Speed is one of the most important features of the new 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. For example, you can back up and restore NICI files. You can also create your own list of related files to include with the backup.
Ability to restore NICI files for a server	Not designed to provide this.	Lets you back up and restore NICI files, so you can access encrypted data after a restore. This can save you a lot of time when restoring.

Issue	Legacy TSA for NDS Backup	Backup Tool “Hot Continuous 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 Section 16.3, “Using Roll-Forward Logs,” on page 404.

16.2.3 Overview of How the Backup Tool Does a Restore

Before restoring, you need to collect all your backup files by following the instructions in Section 16.4, “Preparing for a Restore,” on page 408. When you direct the Backup Tool to begin the restore through iManager or DSBK, the process is done by the Backup Tool 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.

   NOTE: 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. You can prevent this by using `-s` switch.

   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.

   NOTE: 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 404.

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 Section 16.7, "Recovering the Database If Restore Verification Fails," on page 426 for how to recover the server.

NOTE: 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.

16.2.4 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.

NOTE: 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.
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 Tool 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.

Attribute	Explanation
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 -s *file_size*.
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.
password	Specifies the NICI backup password. The same password has to be specified to restore the NICI files.
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 server, with NICI security files included in the backup:

```
<?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:\WINDOWS\system32\novell\nici\bhawkins\XARCHIVE.001" encoding="base64"
type="nici">the data is included here</file>

<file size="4228" name="C:\WINDOWS\system32\novell\nici\bhawkins\XMGRCFG.KS2"
encoding="base64" type="nici">the data is included here</file>

<file size="168" name="C:\WINDOWS\system32\novell\nici\bhawkins\XMGRCFG.KS3"
encoding="base64" type="nici">the data is included here</file>

<file size="aaac" name="C:\WINDOWS\system32\novell\nici\nicintacl.exe"
encoding="base64" type="nici">the data is included here</file>

<file size="150" name="C:\WINDOWS\system32\novell\nici\NICISDI.KEY"
encoding="base64" type="nici">the data is included here
</file>

<file size="4228" name="C:\WINDOWS\system32\novell\nici\system\Xmgrcfg.ks2"
encoding="base64" type="nici">the data is included here
</file>

<file size="168" name="C:\WINDOWS\system32\novell\nici\system\Xmgrcfg.ks3"
encoding="base64" type="nici">the data is included here
</file>

<file size="1414" name="C:\WINDOWS\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.

16.2.5 Format of the Backup Log File

The eDirectory Backup Tool keeps a log that shows a high-level view of Backup Tool 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 Tool 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 NICI files or 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
Backing up security file: sys:/system/nici/INITNICI.LOG
Backing up security file: sys:/system/nici/NICISDI.KEY
Backing up security file: sys:/system/nici/XARCHIVE.000
Backing up security file: sys:/system/nici/XARCHIVE.001
Backing up security file: sys:/system/nici/XMGRCFG.KS2
Backing up security file: sys:/system/nici/XMGRCFG.KS3
Backing up security file: sys:/system/nici/XMGRCFG.NIF
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
Restoring file sys:/system/nici/INITNICI.LOG
Restoring file sys:/system/nici/NICISDI.KEY
Restoring file sys:/system/nici/XARCHIVE.000
Restoring file sys:/system/nici/XARCHIVE.001
Restoring file sys:/system/nici/XMGRCFG.KS2
Restoring file sys:/system/nici/XMGRCFG.KS3
Restoring file sys:/system/nici/XMGRCFG.NIF
Database restore finished
Completion time 00:00:15
Restore completed successfully
```

16.2.6 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 Tool 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 Tool 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 434.

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.

NOTE: 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 Section 1.6, “Replicas,” on page 55.

- 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 DSRepair, and then re-add the replicas to the servers using replication from the copy on the DSMASTER server. These steps are documented in Section 16.7, “Recovering the Database If Restore Verification Fails,” on page 426.

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.

16.2.7 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 ring 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 Tool Does a Restore" on page 397.

For information on what to do if the restore verification fails, see Section 16.7, "Recovering the Database If Restore Verification Fails," on page 426.

16.3 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, but 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 Section 16.7, "Recovering the Database If Restore Verification Fails," on page 426.

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.

NOTE: 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 407.

In this section:

- "Issues to Be Aware of When Turning On Roll-Forward Logging" on page 405
- "Location of the Roll-Forward Logs" on page 406
- "Backing Up and Removing Roll-Forward Logs" on page 407
- "Cautionary Note: Removing eDirectory Also Removes the Roll-Forward Logs" on page 408

You can turn on and configure roll-forward logging using either iManager or DSBK. See "Configuring Roll-Forward Logs with iManager" on page 536 or "Configuring Roll-Forward Logs with DSBK" on page 416.

16.3.1 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 406.
- **Document the location of the roll-forward logs.** For more information, see "Location of the Roll-Forward Logs" on page 406.
- **Monitor the available disk space where the logs are located.** For more information, see "Backing Up and Removing Roll-Forward Logs" on page 407.
- **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.

16.3.2 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.

  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 DSBK. 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 DSBK 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 407.
- **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 Section 16.4, "Preparing for a Restore," on page 408.

16.3.3 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 DSBK Client, enter the `getconfig backup` command. See "Configuring Roll-Forward Logs with DSBK" on page 416 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.

16.3.4 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.

16.4 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 408. If you are not sure how to gather the right backup files, see "Locating the Right Backup Files for a Restore" on page 409.

16.4.1 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 409.

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 (http://support.novell.com/docs/Readmes/InfoDocument//2960653.html).

- ❒ 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.

During the restore process, the eDirectory Backup Tool first restores the full backup. After this is complete, the Backup Tool prompts you to enter the filenames of the incremental files. It provides you with the ID of the next file. After all incremental files are restored, the Backup Tool moves on to the roll-forward logs. See also "Overview of How the Backup Tool Does a Restore" on page 397.

After you have gathered all the files, perform the restore using either iManager or the DSBK Client. See "Restoring from Backup Files with DSBK" on page 417 or "Restoring from Backup Files with iManager" on page 537.

16.4.2 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 Tool 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 398.

   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 Tool 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 Tool 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.

16.5 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. NICI back up and restore is supported only for a root user, and not for a non-root user.

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.

The database key required to open the DIB is wrapped with NICI keys. Hence if an eDirectory backup is performed independent of NICI backup then it is of no use. The eDirectory backup solution (DSBK and eMBox Backup) has a switch (`-e`) that enables:

1. Backing up the NICI keys when an eDirectory backup is run
2. Restoring the NICI keys when an eDirectory restore is run

For more information on the eDirectory backup solution, refer to the Section 16.6, "Using DSBK," on page 412.

16.5.1 Backing Up NICI

NICI backup can be performed along with full eDirectory backup and also with incremental eDirectory backup.

The command to perform a NICI backup is as follows:

```
dsbk backup -f file_name -l log_file_name -e password
```

`-f` and `-l` are mandatory options that have to be used with the backup command.

`-e` is the switch to backup NICI files.

`file_name` specifies the file name and location of the backup file you want the Backup Tool to create.

`log_file_name` specifies the file name and location of a log file created to record the results of the backup operation.

`password` specifies the NICI backup password. The password can be specified as a clear text. On UNIX platforms passing the password as a file is also supported. This same password has to be specified to restore the NICI files.

NOTE: If a NICI backup password is not specified with the `-e` switch, then the following error messages are displayed:

In DSBK:

```
Enter password along with the (-e) option!
DSBK error! 4
```

16.5.2 Restoring NICI

1 Restore NICI files alone (not DIB).

```
dsbk restore -f file_name -l log_file_name -e password
```

`-f` and `-l` are mandatory options that have to be used with the restore command.

`-e` is the switch to restore NICI files.

`file_name` specifies the file name and location of the backup file that contains the information to be restored.`log_file_name` specifies the file name and location of a log file created to record the results of the restore operation.`password` specifies the NICI backup password that was used when the NICI files were backed up. If a wrong password is specified when trying to restore the NICI files then an error message is displayed.

2. Restart the ndsd server.
3. Restore the DIB.

```
dsbk restore -f file_name -l log_file_name -a -r -o
```

`-f` and `-l` are mandatory options that have to be used with the restore command.

`-a` activates DIB after verifying, `-r` restores DIB set, and `-o` opens database when finished.

If NICI backup was performed during a full backup and also during an incremental backup and if different NICI backup passwords were used during the full backup and the incremental backup then when restoring the NICI files the password that was used with the full backup should be used to restore the NICI files.

NOTE: If a password is not specified with the `-e` switch then the following error messages are displayed:

In DSBK:

```
Enter password along with the (-e) option!
DSBK error! 4
```

If a wrong password is specified during the NICI restore, the following error is displayed:

```
NICI RESTORE: "NICI Files has not been restored(Check your parameters)" Error!: -32
```

16.6 Using DSBK

DSBK is a thin command line parser that performs eDirectory Backup, and lets you to initiate a backup from the server console without having to log in first or set up Role-Based Services. It runs as a script on Linux/UNIX and a console utility on Windows.

After a DSBK operation has completed, the results of the operation are written to a file (`dsbk.pipe` on Linux) 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.

NOTE: Ensure that you have gone through all the guidelines given by Novell before finalizing on your backup/restore setup.

Before performing backup and restore tasks, review Section 16.1, “Checklist for Backing Up eDirectory,” on page 392 for an overview of the issues involved in planning an effective eDirectory backup strategy.

This section covers the following:

- Section 16.6.1, “Prerequisites,” on page 413
- Section 16.6.2, “Using DSBK on Various Platforms,” on page 414
- Section 16.6.3, “Backing Up Manually with DSBK,” on page 415
- Section 16.6.4, “Configuring Roll-Forward Logs with DSBK,” on page 416
- Section 16.6.5, “Restoring from Backup Files with DSBK,” on page 417
- Section 16.6.6, “Backup and Restore Command Line Options,” on page 419
- Section 16.6.7, “Running DSBK as a cron Job,” on page 426

16.6.1 Prerequisites

- ❒ On Linux, make sure to do the following:

  1. Create a file `dsbk.conf` in `/etc`.
  2. Create a temporary file. For example, `/tmp/dsbk.tmp`.
  3. Specify the location of the temporary file created in the previous step in the `/etc/dsbk.conf` file.

- ❒ 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 Section 16.3, “Using Roll-Forward Logs,” on page 404. For how to turn them on, see “Configuring Roll-Forward Logs with DSBK” on page 416.

- ❒ Decide which additional files you want to back up along with eDirectory, and create an include file if necessary.

  You can back up the stream files using switches. We recommend that you always back up NICI files. For more information on how to back up NICI, refer to Section 16.5, “Backing Up and Restoring NICI,” on page 411.

  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 shortly after doing the eDirectory backup, to place the eDirectory backup files safely on tape. The Backup Tool 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, using the `-s` option 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 419.

16.6.2 Using DSBK on Various Platforms

- "Using DSBK on Linux, Solaris, and AIX" on page 414
- "Using DSBK on Windows" on page 415

Using DSBK on Linux, Solaris, and AIX

The Linux version of DSBK requires a configuration file. The configuration file contains a single line that points to another file to be created and used to pass commands to the DSBK module through the DSTrace utility.

Let's look at an example. In `/etc/dsbk.conf`, we set a value of:

```
/tmp/dsbk.command
```

NOTE: Ensure that the owner of the instance has Read/Write permissions for the above mentioned command file.

This file is then used as an intermediary for the DSBK module.

The output for the command is written into the eDirectory instance specific log file (Default instance: `/var/opt/novell/eDirectory/log/ndsd.log`):

```
DSBK HELP
To get help on a specific function type "help <function name>"
 Current functions:
  backup
  restore
  restadv
  getconfig
  setconfig
  cancel
```

By setting up DSBK, commands can be entered into a crontab to execute `dsbk getconfig` and `dsbk backup` commands on a regular basis, allowing for full backups once weekly and incrementals on the other days, or whatever combinations are desired.

Using RFL in DSBK

- Turn on the RFL using the following command:

  ```
  dsbk setconfig -L
  ```

  The `-L` option starts a new roll forward logging session.

- Set a location for the roll-forward logs to be created using the following command:

  ```
  dsbk setconfig -L -r <roll forward log directory>
  ```

- Get a location for the roll-forward logs to be created using the following command:

  ```
  dsbk getconfig
  ```

TIP: When using the DSBK utility interactively, have a second terminal window open with `tail -f <instance specific ndsd.log>` running so that the output to the entered commands is immediately readable.

Once the back up is completed, back it up using standard filesystem backup utilities.

NOTE: For detailed information on DSBK command line options, refer to the Section 16.6.6, "Backup and Restore Command Line Options," on page 419.

Using DSBK on Windows

This section discusses the basic operation of the DSBK utility on the Windows platform.

For using DSBK on a Windows server that hosts eDirectory, perform the following steps:

1. Invoke the utility through the *Novell eDirectory Services* console. *dsbk.dlm* is one of the options available in the list of services in the Services tab. The *dsbk* subcommand and any parameters for that subcommand are specified in the *Startup Parameters* field.
2. View the current configuration for the backup using the `getconfig` switch. The output of all the DSBK commands is appended to the file `c:\novell\nds\backup.out`. This location is always used for Windows installations, regardless of the path where eDirectory is installed or the location of the eDirectory configuration files.
3. Set a location for the roll-forward logs to be created using the following command:

   ```
   setconfig -r <roll forward log directory> -L
   ```

   The `-L` option starts a new roll forward logging session.
4. Start backup on the tree by giving the following command:

   ```
   backup -f <backup file> -l <logfile> -t -w -b -e <password>
   ```

   Use the following options:

   - `-t`: Takes the backup of stream files.
   - `-w`: Overwrites any existing backup file with same name.
   - `-b`: Performs a full backup.

`-e <password>`: Performs a NICI backup using the password provided.

For example, start the backup as follows:

```
backup -f c:\dsbk.bak -l c:\backup.log -t -w -b -e novell
```

You can confirm the status of the backup done in the `backup.out` file.

NOTE: For detailed information on DSBK command line options, refer to the Section 16.6.6, "Backup and Restore Command Line Options," on page 419.

You can turn on the RFL using the following command:

```
setconfig -r <roll forward log directory> -L
```

16.6.3 Backing Up Manually with DSBK

Use DSBK 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.

Using DSBK, 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 Section 18.5, "Upgrading Hardware or Replacing a Server," on page 498).

- Set the database to stay closed and locked after a backup.
- Set the maximum backup file size.

Procedure

To back up the eDirectory database on a server using DSBK:

1 Enter the `dsbk backup` command, following this general pattern:

```
dsbk backup -b -f backup_filepath_and_backup_filename -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, in Windows, enter the following command:

```
dsbk 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 Tool would replace it.

The output is entered in `ndsd.log`, which indicates whether the backup is successful.

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 Tool only places them on the server.

16.6.4 Configuring Roll-Forward Logs with DSBK

Use DSBK 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 Section 16.3, "Using Roll-Forward Logs," on page 404.

Procedure

1 Find out the current settings by entering

```
dsbk 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 C:\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 ***
```

2 Change the settings using the `setconfig` command, following this general pattern:

`dsbk 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.

Ideally, you would have a separate disk partition/volume dedicated to roll-forward logs to make it easier to monitor disk space and rights.

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 407.

16.6.5 Restoring from Backup Files with DSBK

Use DSBK to restore an eDirectory database from data stored in backup files you created manually. The results of the restore process are written to the log file you specify.

DSBK also lets you use advanced restore options not available in iManager. They are described in "Backup and Restore Command Line Options" on page 419, under restore and restadv.

Additional Prerequisites

❒ 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 419.
- ❒ Review the description of the restore process in "Overview of How the Backup Tool Does a Restore" on page 397.

Procedure

To restore an eDirectory database on a server using DSBK:

1. Make sure you have gathered the backup files you need, as described in Section 16.4, "Preparing for a Restore," on page 408.
2. Enter the `dsbk restore` command, following this general pattern:

```
dsbk 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 406.

   For example:

```
dsbk restore -r -a -o -f $HOME/backup/nds.bak -d $HOME/backup/rfldir/nds.rfl -
l $HOME/backup/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.

   DSBK restores the full backup. The output is entered in `ndsd.log`, which will indicates whether the restore was successful.

3. (Conditional) If the restore was not successful, check the log file to see the errors.

   If the restore verification fails, see Section 16.7, "Recovering the Database If Restore Verification Fails," on page 426.

   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.

   ---

4. (Conditional) If you restored NICI security files, after completing the restore, restart the server to reinitialize NICI and then restore DIB.
5. Make sure the server is responding as usual.

6 (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 Section 16.3, "Using Roll-Forward Logs," on page 404.

Your restore should now be complete, and NICI reinitialized with the restored NICI files so you can access encrypted information. 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.

16.6.6 Backup and Restore Command Line Options

The eDirectory Backup Tool 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 file_name`	(Mandatory) Backup filename and path
	Specifies the filename and location of the backup file you want the Backup Tool to create. This file must be on the server you are backing up. For example, `backup -f C:\backup\ndsbak.bak` will back up the database to `C:\backup\ndsbak.bak`.
`-l file_name`	(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 `-i` nor `-c` 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
`-u file_name`	(Optional) User includes filename and path. 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 could be important when restoring the server's eDirectory database. In the include file, list the full path of each file you want backed up, followed by a semicolon (;). Don't include any spaces or hard returns in the list of files. 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 401 and “Format of the Backup File Header” on page 398. **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.
`-s file_size`	(Optional) Backup file size limit (MB) Specifies the maximum size (MB) 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. 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. For example, you could set the maximum size of the backup files to 10 MB using the following switches as part of your command: `backup -f C:\backup\mydib.bak -s 10`. If the database is 35 MB, this is the resulting set of backup files: `C:\backup\mydib.bak`, size is 9.6 MB `C:\backup\mydib.bak.00001`, size is 9.6 MB `C:\backup\mydib.bak.00002`, size is 9.6 MB `C:\backup\mydib.bak.00003`, size is 5.6 MB The smallest possible size is close to 1 MB. The first file could be larger, depending on how many files are being included with the backup. The first file contains an attribute under the backup tag called `number_of_files`. 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 `backup_file`. This is the original name of the file. For more information, see “Format of the Backup File Header” on page 398. When restoring a set of backup files like the set in the example above, the command would be `restore -f C:\backup\mydib.bak -l log_file_path_and_filename` The Backup Tool identifies that there are multiple files and looks for them in the same directory as the first, but with the above name mutations. **TIP**: The backup files can also be made much smaller using a third-party file compression tool. They compress approximately 80%.

Option and Switches	Description
`-w`	(Optional) Overwrite existing backup file of same name 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 Tool 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. 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. **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. 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, DSBK will ask you whether you want to overwrite the file.
`-c`	(Optional) Perform a cold backup 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.
`-o`	(Optional) Leave database closed after cold backup 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 Section 18.5, “Upgrading Hardware or Replacing a Server,” on page 498).
`-d`	(Optional) Disable DS agent after a cold backup 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 Section 18.5, “Upgrading Hardware or Replacing a Server,” on page 498). The DS agent is disabled by setting the login disabled attribute on the pseudo server. This results in a -663 error when eDirectory starts.
`-e` *password*	Perform a NICI backup *password* specifies the NICI backup password. This same password has to be specified to restore the NICI files.
`restore`	Perform a restore of the database and associated files.

Option and Switches	Description
`-f file_name`	(Mandatory) Backup filename and path Specifies which full backup to restore from. This file must be located on the server being restored. For example, `restore -f C:\backup\ndsbak.bak` will restore from the file `C:\backup\ndsbak.bak`. 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.
`-l file_name`	(Mandatory) Log filename and path Specifies the log file to record the results of the restore operation.
`-r`	(Optional) Restore DIB set Specifies that the eDirectory database should be restored. **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.
`-d dir_name`	(Optional) Roll-forward log directory 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. 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 Tool does not replay any logs against the database, even if roll-forward logging was turned on at the time of the backup. To determine the first required roll-forward log, open the last backup file being restored in a text editor and read the `current_log` 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 398. **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.
`-u`	(Optional) Restore user included files Restores the user files that were included with the backup of the database. 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.
`-a`	(Optional) Activate DIB after verifying 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 Tool Does a Restore” on page 397.

Option and Switches	Description
`-o`	(Optional) Open database when finished Directs the Backup Tool 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 Tool Does a Restore” on page 397.
`-s`	Directs the Backup Tool not to reset roll forward log after Restore operation. It is mainly used in the instance of default RFL location.
`-n`	(Optional) Do not verify database after restore Directs the Backup Tool 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 ring it participates in. For information about transitive vectors, see “Transitive Vectors and the Restore Verification Process” on page 404. The database is not renamed from RST to NDS unless another option is used to do so. **IMPORTANT**: We do not recommend using this option unless suggested by Novell Support.
`-v`	(Optional) Override restore Renames the database from RST to NDS without trying to verify. **IMPORTANT**: We do not recommend using this option unless suggested by Novell Support.
`-k`	(Optional) Remove lockout on database Removes the lockout on the NDS database.
`-i`	Comma separated list of incremental files in order.
`-e password`	Restore the backed up NICI files *password* specifies the NICI backup password that was used when the NICI files were backed up. If a wrong password is specified when trying to restore the NICI files then an error message is displayed.
`restadv`	Advanced restore options. **NOTE**: The DS agent will be closed for all advanced restore options.
`-l file_name`	(Mandatory) Log filename and path Specifies the log file to record the results of the restore operation.
`-o`	(Optional) Open database when finished Directs the Backup Tool 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 Tool Does a Restore” on page 397.

Option and Switches	Description
`-n`	(Optional) Try to verify a previously failed restore Tries to verify a previously restored RST database.
`-m`	(Optional) Remove restored DIB files Removes the RST database if it is present.
`-v`	(Optional) Override restore Renames the database from RST to NDS without trying to verify. **IMPORTANT**: We do not recommend using this option unless suggested by Novell Support.
`-k`	(Optional) Remove lockout on database Removes the lockout on the NDS database.
`-i`	Comma separated list of incremental files in order. **IMPORTANT**: This option is applicable to DSBK only.

`getconfig`	Retrieves the current roll-forward log configuration.
	
	No options are needed.
	
	Displays the current settings. For example, on a server with roll-forward logging turned off, the `getconfig` command would return information like the following:
	
	`Roll forward log status OFF`
	`Stream file logging status OFF`
	`Current roll forward log directory C:\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 ***`

`setconfig`	Sets the roll-forward log configuration.
`-L`	(Optional) Start keeping roll-forward logs. 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. You must use roll-forward logging for servers that participate in replica ring, so that you can restore a server back to the synchronization state that the other servers expect. 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 407. For more information, see Section 16.3, “Using Roll-Forward Logs,” on page 404.

Option and Switches	Description
`-l`	(Optional) Stop keeping roll-forward logs 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. 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. For more information, see Section 16.3, “Using Roll-Forward Logs,” on page 404.
`-T`	(Optional) Start logging of stream files (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. 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.
`-t`	(Optional) Stop logging of stream files 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. Turning off logging of stream files can help slow the growth of roll-forward logs.
`-r` *`dir_name`*	(Optional) Set roll-forward log directory Changes the directory where the roll-forward logs are placed. For example, if the command used was `setconfig -r vol2:\rfl`, a directory is created under `vol2:\rfl` and the roll-forward logs are placed in it. 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 `vol2:\rfl\nds.rfl\`. If you renamed the eDirectory database from NDS to ND1, the roll-forward log directory would be changed to `vol2:\rfl\nd1.rfl\`. You can find out the current location by entering the `getconfig` command. 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. **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. For more information, see Section 16.3, “Using Roll-Forward Logs,” on page 404.

Option and Switches	Description
`-n file_size`	(Optional) Set minimum roll-forward log size 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.
`-m file_size`	(Optional) Set maximum roll-forward log size 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.
`-s`	(Optional) Start a new roll-forward log 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.
`cancel`	Cancels any running backup or restore operation. No options are needed. **NOTE**: This option is not applicable to DSBK.

16.6.7 Running DSBK as a cron Job

The `dsbk` script does not contain the full path to the DSTrace binary. Therefore, if you run the script as a cron job using the default settings, the script fails. However, do not change the `/opt/novell/eDirectory/bin/dsbk` script to add the path, because subsequent eDirectory patches will overwrite this file and revert any customizations you may have made to the script.

Instead, before you run dsbk as a cron job, set the `PATH` environment variable within the `crontab` file to include the directory where `ndstrace` is located. The cron job can then find and run the `ndstrace` application.

16.7 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 Tool Does a Restore" on page 397 and "Transitive Vectors and the Restore Verification Process" on page 404.

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.

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 427. Then continue with "Repair the Failed Server and Re-add Replicas to the Server" on page 428.

16.7.1 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 ring 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 eDirectory 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.

- **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 Section 12.9, "DSRepair Options," on page 285.

WARNING: If you use DSRepair with `-a` or `-Ad`, some of the advanced options can cause damage to your tree.

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 Re-add Replicas to the Server" on page 428.

16.7.2 Repair the Failed Server and Re-add 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 re-adding the replicas to the server. This way, the server receives a new, up-to-date copy of each replica. When each replica has been re-added, 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 427.

2 Specify the advanced restore option to override the restore, then specify a log filename:

```
dsbk restadv -v -l logfilename
```

This advanced restore option renames 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.

- **Windows:** Click *Start* > *Settings* > *Control Panel* > *Novell eDirectory Services*. Select *dsrepair.dlm*. In the Startup Parameters field, type `-XK2 -rd`. Click *Start*.
- **UNIX:** Enter the following command:

  ```
  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.

---

4 When the repair is finished, remove the lockout and open the database using the following advanced restore options in the eMBox Client:

```
dsbk 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 Replica Management* > *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 146 for more information.

7 To restore NICI security files, first restore the NICI files alone and then restart the NDSD server and restore the DIB.

8 (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 Section 16.3, "Using Roll-Forward Logs," on page 404.

16.8 Scenarios for Backup and Restore

- "Scenario: Losing a Hard Drive Containing eDirectory in a Single-Server NetWork" on page 430

- "Scenario: Losing a Hard Drive Containing eDirectory in a Multiserver Environment" on page 431
- "Scenario: Losing an Entire Server in a Multiple-Server Environment" on page 433
- "Scenario: Losing Some Servers in a Multiple-Server Environment" on page 433
- "Scenario: Losing All Servers in a Multiple-Server Environment" on page 434

16.8.1 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 Backup Tool functionality provides a simple solution for Indira to back up and restore eDirectory. It's server-centric and it's fast.

On eDirectory 8.7.3 or to later versions, Indira sets up unattended backups for her server using batch files to run the Backup Tool.

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.

16.8.2 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 ring. 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 ring. 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* > *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.

16.8.3 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 ring. 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 404), 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 ring, 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 Section 16.7, "Recovering the Database If Restore Verification Fails," on page 426.

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.

16.8.4 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 ring, 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.

16.8.5 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 Tool, 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 404). 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 ring.

- 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.
- 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 Section 16.7, "Recovering the Database If Restore Verification Fails," on page 426.)

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.

16.9 Disaster Recovery Plan using DSBK

A disaster recovery plan enables you to recover your disk back to the configuration at the time of corruption. You have to backup your server's disk to a remote location, so that you can recover the server even if the operating system gets corrupted.

This section provides a sample disaster recovery plan for an eDirectory server:

- Section 16.9.1, "Disaster Recovery Plan on Linux and UNIX," on page 435
- Section 16.9.2, "Disaster Recovery Plan on Windows," on page 437

16.9.1 Disaster Recovery Plan on Linux and UNIX

To take a backup of the server's disk:

1. Configure DSBK:
   - **1a** Create a file `dsbk.conf` in `/etc`.
   - **1b** Create a temporary file. For example, `/tmp/dsbk.tmp`.
   - **1c** Specify the location of the temporary file created in the previous step in the `/etc/dsbk.conf` file.
2. Mount the server's disk to a remote machine in the read/write mode, to store all backup files on a remote machine disk.

   For example, `eDirServer# mount <remote machine IP>:/home/backup/ /mnt/dsbkBkp`

3 Set the custom backup location using the following command:

```
dsbk setconfig -L -T -r /mnt/dsbkBkp
```

NOTE: Ensure that you run DSBK from the following location on the server: `/opt/novell/eDirectory/bin`.

4 Take a full backup along with NICI to the remote location file system:

```
dsbk backup -f <backup file location> -l <log file location> -e <password for NICI backup> -t -b
```

For example, `dsbk backup -f /mnt/dsbkBkp/fb1.bak -l /mnt/dsbkBkp/fb1.log -e novell -t -b`.

NOTE: The `-e` option is used to back up NICI. In the example, `novell` is password for NICI Backup. You may choose your own password, and the same password must be used during NICI restore.

5 Take incremental backups using the following command:

```
dsbk backup -f <incremental backup file location> -l <incremental log file location> -t -i
```

For example:

Day 1: `dsbk backup -f /mnt/dsbkBkp/ib1.bak -l /mnt/dsbkBkp/ib1.log -t -i`

Day 2: `dsbk backup -f /mnt/dsbkBkp/ib2.bak -l /mnt/dsbkBkp/ib2.log -t -i`

NOTE: While taking an incremental backup, you do not have to back up NICI.

If the eDirectory server gets corrupted, then perform the following steps to recover the eDirectory server using the remote location backup:

1 If the operating system is corrupted, install the operating system as before.

2 If only eDirectory is corrupted, then do a clean up of the system for eDirectory by removing the eDirectory RPMs.

3 Install the same eDirectory as before and configure a single server dummy tree. For example,

```
ndsconfig new -t dummy_bkp_tree -n novell -a admin.novell -w novell
```

4 Restore NICI from the full backup file (without the `-d`, `-r`, `-a`, `-o` options):

```
dsbk restore -f <backup file location> -l <log file location> -e <password used to NICI backup>
```

For example, `dsbk restore -f /mnt/dsbkBkp/fb1.bak -l /mnt/dsbkBkp/restore1.log -e novell`

5 After restoring NICI, restart the eDirectory server.

6 Restore both the full and incremental backup files. For example,

```
dsbk restore -f /mnt/dsbkBkp/fb1.bak -l /mnt/dsbkBkp/restore2.log -d /mnt/dsbkBkp/nds.rfl/ -r -a -e novell -o -i /mnt/dsbkBkp/ib1.bak, /mnt/dsbkBkp/ib2.bak
```

For more information on backup and restore commands, refer to the "Using DSBK on Linux, Solaris, and AIX" on page 414.

16.9.2 Disaster Recovery Plan on Windows

To take a backup of the server's disk:

1. Map the server's disk to a remote machine in the read/write mode. For example, `O:\dsbkBkp`
2. To run DSBK command:
   - **2a** Open eDirectory server console by running `NDScons.exe`.
   - **2b** Click *dsbk.dlm* from the *Services* tab.
   - **2c** Enter DSBK commands in the *Startup Parameter* field.
3. Set the custom backup location using the following command:

   ```
   setconfig -L -T -r O:\dsbkBkp
   ```
4. Take a full backup along with NICI, to the remote location file system:

```
backup -f <backup file location> -l <log file location> -e <password for NICI
backup> -t -b
```

> **NOTE**: The `-e` option is used to backup NICI. In the example, `novell` is password for NICI Backup. You may choose your own password, and the same password must be used during NICI restore.

5. Take incremental backups using the following command:

```
backup -f <incremental backup file location> -l <incremental log file location> -t
-i
```

For example:

Day 1: `backup -f O:\dsbkBkp\ib1.bak -l O:\dsbkBkp\ib1.log -t -i`

Day 2: `backup -f O:\dsbkBkp\ib2.bak -l O:\dsbkBkp\ib2.log -t -i`

> **NOTE**: While taking an incremental backup, you do not have to back up NICI.

If the eDirectory server gets corrupted, then perform the following steps to recover the eDirectory server using the remote location backup:

1. If the operating system is corrupted, install the operating system as before.
2. If only eDirectory is corrupted, then do a clean up of the system for eDirectory.
3. Install the same eDirectory as before and configure a single server dummy tree.
4. Restore NICI from the full backup file (without the `-d`, `-r`, `-a`, `-o` options):

   For example:

```
restore -f <backup file location> -l <log file location> -e <password used for NICI
backup>
```

   For example, `restore -f O:\dsbkBkp\fb1.bak -l O:\dsbkBkp restore1.log -e novell`
5. After restoring NICI, restart the eDirectory server.
6. Restore both the full and incremental backup files.

   For example:

```
restore -f O:\dsbkBkp\fb1.bak -l O:\dsbkBkp\restore2.log -d O:\dsbkBkp\nds.rfl\
-r -a -e novell -o -i O:\dsbkBkp\ib1.bak, O:\dsbkBkp\ib2.bak
```

For more information on backup and restore commands, refer to the "Using DSBK on Windows" on page 415.

17 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 IBM Tivoli NetView or Solstice SunNet 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 8.8. It contains the following topics:

- Section 17.1, “Definitions and Terminology for SNMP,” on page 439
- Section 17.2, “Understanding SNMP Services,” on page 440
- Section 17.3, “eDirectory and SNMP,” on page 442
- Section 17.4, “Installing and Configuring SNMP Services for eDirectory,” on page 445
- Section 17.5, “Monitoring eDirectory Using SNMP,” on page 453
- Section 17.6, “Troubleshooting,” on page 478

17.1 Definitions and Terminology for SNMP

The following tables contain terminologies used in this chapter.

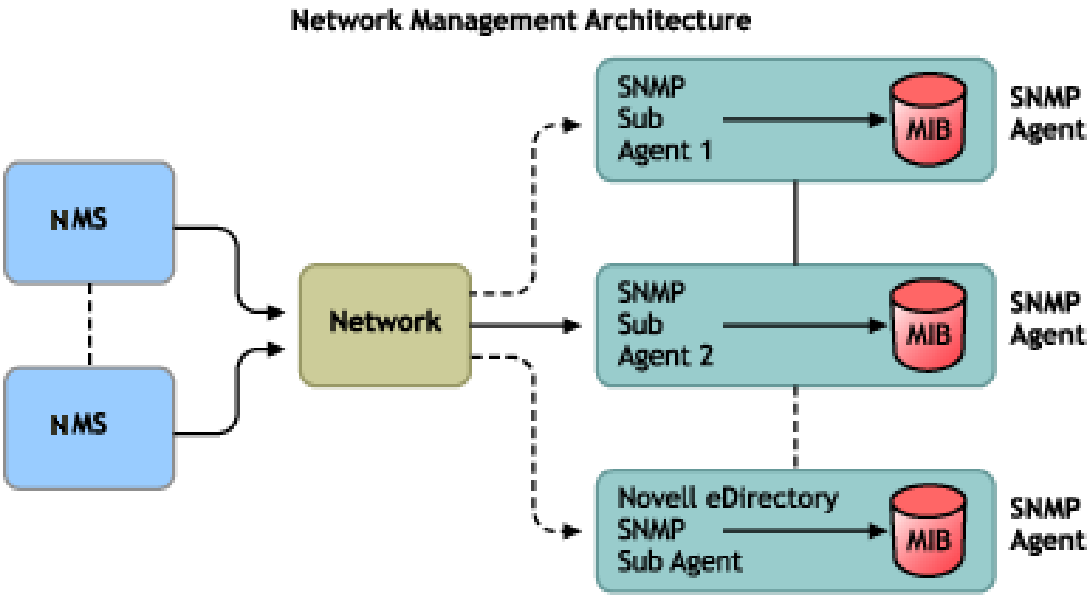
Terminology	Definition
EMANATE	Enhanced Management Agent Through Extensions is a product from SNMP Research International, Inc.
SNMP	Simple Network Management Protocol is used to exchange data about the network activity.
NAA	Native Agent Adapter
NMS	Network Management Station
MA	Management Agent
SA	Subagent
MIB	Management Information Base
NCP	NetWare/Novell Core Protocol
NMA	Network Management Application
edir.mib	Novell eDirectory server Monitoring MIB, which has MIB objects and traps relevant to Novell eDirectory.
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.

17.2 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 17-1 *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. NMS 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)
- SNMP FAQ (http://www.faqs.org/faqs/snmp-faq/part1)
- RFC 1157 (http://www.ietf.org/rfc/rfc1157.txt)
- SNMPLink (http://www.snmplink.org)
- SNMPInfo (http://www.snmpinfo.com)
- SNMP RFC Standard MIBs and Informative Links (http://www.wtcs.org/snmp4tpc/snmp_rfc.htm)
- RFC 2605 (http://www.ietf.org/rfc/rfc2605.txt?number=2605)

17.3 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.

17.3.1 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

17.3.2 Understanding How SNMP Works with eDirectory

SNMP implementation on eDirectory provides useful eDirectory information on statistics on the accesses, operations, errors, and cache performance. Traps on the occurrence of events can also be sent with SNMP implementation. Traps and statistics are defined in the MIB.

NOTE: You might have to access the encrypted attributes only over a secure channel, if you have specified that you always need a secure channel to access these attributes. For more information, refer to Section 11.1, "Encrypted Attributes," on page 243.

Directory Service Monitoring 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).enterprises(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 474.

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 two traps cannot be configured.

NOTE: For more information on traps, see "Traps" on page 454.

For more information on statistics and traps, see `edir.mib`.

`edir.mib` is located in the following directories:

Windows: *`install_directory`*`\SNMP`
Linux and UNIX: `/etc/opt/novell/eDirectory/conf/ndssnmp/`

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

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
`-c <createobj>`	Trap command that specifies the creation of an object.
`-a <userFDN>`	Fully distinguished name of a user having administrative rights
`-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 Linux and 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
```

17.4 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 447.

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 444 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
```

17.4.1 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 Linux and UNIX platforms.

To load the SNMP server module, enter the following commands:

Server	Command
Windows	In the DHost (NDSCONS) screen, select *ndssnmp.dlm* > click *Start*.
Linux, Solaris, and AIX	In the DHost remote management page, to load the SNMP trap server click on the SNMP Trap Server for Novell eDirectory 8.8 action icon to start. or At the prompt, enter the following: `/opt/novell/eDirectory/bin/ndssnmp -l`

To unload the SNMP server module, enter the following commands:

Server	Command
Windows	In the DHost (NDSCONS) screen, select *ndssnmp.dlm*, then click *Stop*.
Linux, Solaris, and AIX	In the DHost remote management page, to unload the SNMP trap server, click the *SNMP Trap Server for Novell eDirectory 8.8* action icon to stop. or At the prompt, enter the following: `/opt/novell/eDirectory/bin/ndssnmp -u`

17.4.2 Subagent Configuration

- "Static Configuration" on page 446
- "Dynamic Configuration" on page 447

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, or AIX. The `ndssnmp.cfg` file is located in the following directories:

Windows: *`install_directory`*`\SNMP\`

Linux and UNIX: `/etc/opt/novell/eDirectory/conf/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 user name and password when starting the subagent. If the status is off, then the user name 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` (Linux and UNIX)

`SSLKEY c:\home\guest\snmp-cert.der` (Windows)

NOTE: This option is not supported if there are multiple instances to be monitored that do not accept a common certificate.

`SERVER hostname/IP_address:NCP_port`

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
```

On Linux and UNIX, if you have multiple instances of eDirectory, you can include all the eDirectory servers you want to monitor as follows:

```
SERVER myserver:1524
```

```
SERVER myserver:2524
```

```
SERVER myserver:6524
```

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 467.

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* > *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.

17.4.3 Setting Up SNMP Services for eDirectory

This section describes setting up the SNMP services for eDirectory on the following platforms:

- "Windows" on page 448
- "Linux" on page 449
- "Solaris" on page 451
- "AIX" on page 452

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

Windows

- "Configuring the Master Agent" on page 448
- "Starting the Master Agent" on page 449
- "Stopping the Master Agent" on page 449
- "Starting the Subagent" on page 449

Configuring the Master Agent

NOTE: The SNMP master agent should be installed before eDirectory is installed. Refer to Microsoft SNMP Services (http://technet.microsoft.com/en-us/library/bb726977.aspx) 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 this option is not enabled, you will be unable to connect to SNMP on Windows.

On Windows platform: 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

1 To start the master agent, do the following:

Click *Start* > *Settings* > *Control Panel* > *Administrative Tools* > *Services* > *SNMP* > *Start*.

2 Enter the following at the command prompt:

```
Net start SNMP
```

Stopping the Master Agent

To stop the master agent, do either of the following:

1 Click *Start* > *Settings* > *Control Panel* > *Administrative Tools* > *Services* > *SNMP* > *Stop*.

2 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.

Linux

On Linux `net-snmp` should be installed. By default, it is installed on most Linux systems.

Setting up SNMP Services on Linux

- "Configuring the Master Agent" on page 449
- "Starting the Master Agent" on page 450
- "Starting the Subagent" on page 450
- "Stopping the Subagent" on page 451

Configuring the Master Agent

To configure the master agent on Linux, make the changes to your `snmpd.conf` file as mentioned in "snmpd.conf Changes" on page 450.

The snmpd.conf file is located in the /etc/snmp directory on SLES and in the /etc directory on other Linux platforms.

snmpd.conf Changes

In the snmpd.conf file, enter the following line:

```
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
com2sec notConfigUser default public	com2sec demouser default public
group notConfigGroup v1 notConfigUser	group demogroup v1 demouser
view systemview included system	view all included .1
access notConfigGroup "" any noauth exact systemview none none	access demogroup "" any noauth exact all all all

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 user name and password when prompted. Upon successful authentication, the following message is displayed if INTERACTION = ON in the /etc/opt/novell/eDirectory/conf/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.

NOTE: When the server goes down, the master agent and subagent also go down. Therefore, to start the master agent and the sub-agent during server reboot time, execute the following commands:

```
chkconfig snmpd on
chkconfig ndssnmpsa on
```

Stopping the Subagent

To stop the subagent, execute the following command:

```
/etc/init.d/ndssnmpsa stop
```

Solaris

- "Configuring the Master Agent" on page 451
- "Starting the Master Agent" on page 451
- "Configuring the Subagent" on page 452
- "Starting the Subagent" on page 452
- "Stopping the Subagent" on page 452

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 Oracle Software Downloads Web site (http://www.oracle.com/technetwork/indexes/downloads/index.html) 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

To start the subagent, execute the following command:

```
/etc/init.d/ndssnmpsa start
```

Enter the user name and password when prompted. Upon successful authentication, the following message is displayed if `INTERACTION = ON` in the `/etc/opt/novell/eDirectory/conf/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 452
- "Starting the Master Agent" on page 453
- "Starting the Subagent" on page 453
- "Stopping the Subagent" on page 453

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)
```

Add the following in the `/etc/snmpd.peers` file:

```
ndssnmpsa 1.3.6.1.4.1.23.2.98 ndssnmpsa_password
```

Starting the Master Agent

To start the master, execute the following command:

```
/usr/sbin/snmpdv1
```

Starting the Subagent

To start the subagent, execute the following command:

```
/etc/ndssnmpsa start
```

Enter the user name and password when prompted. Upon successful authentication, the following message is displayed if `INTERACTION= ON` in the `/etc/opt/novell/eDirectory/conf/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
```

17.5 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.

17.5.1 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.

Trap Number	Trap Name	Trap Is Generated When
1	ndsCreateEntry	A new object is added in the directory. Example: Create an object using LDAP tools, ICE, ConsoleOne®, or iManager.
2	ndsDeleteEntry	An existing object is deleted. Example: Create an object using LDAP tools, ICE, ConsoleOne, or iManager.
3	ndsRenameEntry	An existing object is renamed. Example: Rename an object using LDAP tools, ICE, ConsoleOne, or iManager.
4	ndsMoveSourceEntry	An object is moved to a different context. The trap gives the context of the object before movement. Example: Move an object using ldapmodrdn or ldapsdk.
5	ndsAddValue	A value is added to an object attribute. Example: Add new values to attributes using LDAP tools, ICE, ConsoleOne, or iManager. **NOTE**: If the return value is NULL, you might have to access the directory over a secure channel. For more information, refer to “Accessing the Encrypted Attributes” on page 466
6	ndsDeleteValue	A value is deleted from an object attribute. Example: Delete new values to attributes using LDAP tools, ICE, ConsoleOne, or iManager. **NOTE**: If the return value is NULL, you might have to access the directory over a secure channel. For more information, refer to “Accessing the Encrypted Attributes” on page 466
7	ndsCloseStream	A stream attribute is modified.

Trap Number	Trap Name	Trap Is Generated When
8	ndsDeleteAttribute	A value is deleted from a single-value attribute. Example: Delete an attribute using LDAP tools, ICE, ConsoleOne, or iManager. **NOTE**: If the return value is `NULL`, you might have to access the directory over a secure channel. For more information, refer to "Accessing the Encrypted Attributes" on page 466.
9	ndsCheckSecurityEquiv	The security equivalence vector for the particular entry is checked. Example: Change the security equivalence attribute using LDAP tools, ICE, ConsoleOne, or iManager.
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	ndsSyncInEnd	Inbound synchronization is completed.

Trap Number	Trap Name	Trap Is Generated When
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.
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.

Trap Number	Trap Name	Trap Is Generated When
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.
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.

Trap Number	Trap Name	Trap Is Generated When
40	ndsSchemaSync	Schema are synchronized. Example: Schedule schema synchronization using `ldapsdk schsync`.
41	ndsNameCollision	Two objects on different servers have the same name (they **collide**). Example: 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.
43	ndsChangeModuleState	An eDirectory module (NLM / DLM) is loaded or unloaded. Example: Load or unload the nldap module.
44	ndsLumberDone	The limber background process is started.
45	ndsBacklinkProcDone	The backlink process is completed. Example: Configure DSTrace to start backlink after a particular interval of time.
46	ndsServerRename	A server is renamed. Example: Use ldapmodrdn or ldapsdk to rename the server.
47	ndsSyntheticTime	Objects are created with future time stamps. To synchronize eDirectory servers, synthetic time might be invoked. Example: Add a secondary server to the tree using ndsconfig.
48	ndsServerAddressChange	Limber changes a server referral. Example: Change the IP address of the server and restart ndsd.
49	ndsDSARead	An entry is read. This trap is generated for all operations on eDirectory. Example: Use ldapsearch to generate traps.
50	ndsLogin	eDirectory is logged in to. Example: Login to the tree using ndslogin.

Trap Number	Trap Name	Trap Is Generated When
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 servers 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.
60	ndsRecvReplicaUpdates	A replica receives an update during synchronization. Example: An eDirectory server in a multiple-server tree setup requests updates on the replica that it holds. This operation can be done using ConsoleOne or iManager.

Trap Number	Trap Name	Trap Is Generated When
61	ndsRepairTimeStamps	A replica's time stamps are repaired. Example: Perform a DIB repair operation for timestamps using DSRepair (ndsrepair on Linux and UNIX, or NDSCons on Windows).
62	ndsSendReplicaUpdates	A replica is updated during synchronization. Example: 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.
63	ndsVerifyPass	A password is verified. Example: When the password expires, re-enter the password for confirmation at the change password prompt.
64	ndsBackupEntry	An entry is backed up. Example: Back up Directory objects using the Backup utility (ndsbackup on Linux and UNIX, NDSCons on Windows).
65	ndsRestoreEntry	An entry is restored. Example: Restore the backed-up Directory objects using the Backup utility (ndsbackup on Linux and UNIX, NDSCons on Windows).
66	ndsDefineAttributeDef	An attribute definition is added to the schema. Example: 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 NMAS™. The schema can also be extended using ConsoleOne, iManager, or the schema extension utility ndssch on Linux and UNIX.
67	ndsRemoveAttributeDef	An attribute definition is removed from the schema. Example: Delete an attribute definition from the eDirectory tree schema. The attribute can be deleted using ConsoleOne, iManager or the schema extension utility ndssch on Linux and UNIX.
68	ndsRemoveClassDef	A class definition is removed from the schema. Example: Delete an object class definition from the eDirectory tree schema. This can be deleted using ConsoleOne, iManager, or the schema extension utility ndssch on Linux and UNIX.

Trap Number	Trap Name	Trap Is Generated When
69	ndsDefineClassDef	A class definition is added to the schema. Example: 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 Linux and UNIX.
70	ndsModifyClassDef	A class definition is modified. Example: Modify an existing object class or attribute definitions.
71	ndsResetDSCounters	The internal eDirectory counters are reset.
72	ndsRemoveEntryDir	A file directory associated with an entry is removed.
73	ndsCompAttributeValue	Attribute values are compared. Example: Compare an attribute value against any object.Perform an LDAP search operation against a `User` object to check if its telephone number is the same as the input value.
74	ndsOpenStream	A stream attribute is opened or closed. Example: Create or open a stream for read or write operations. Create a login script for a `User` object. It creates a file under the DIB directory, which results in the generation of this trap.
75	ndsListSubordinates	A List Subordinate Entries operation is performed on a container object. It is a one-level search. Example: Using ConsoleOne or iManager, click a container object to list the objects under it.
76	ndsListContainerClasses	A List Containable Classes operation is performed on an entry. Example: For a given object, list the container classes that can contain the given object. When queried against a user object, the container classes that can contain it are Organization, Organizational Unit, and Domain Classes.

Trap Number	Trap Name	Trap Is Generated When
77	ndsInspectEntry	An Inspect Entry operation is performed on an entry. Example: Inspect any entry to obtain information about the entry and to check if there are any errors that the entry has experienced.This event is generated as part of the Flat Cleaner background process of eDirectory, which results in this trap generation.
78	ndsResendEntry	A Resend Entry operation is performed on an entry. Example: 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.
79	ndsMutateEntry	A Mutate Entry operation is performed on an entry. Example: Mutate a bindery object class to `User` object class.
80	ndsMergeEntries	Two entries are merged. Example: Merge two `User` objects. Merge `Entry2` (`ndsEntryName2`) into `Entry` (`ndsEntryName`).
81	ndsMergeTree	Two eDirectory trees are merged. Example: Merge two eDirectory trees using DSMerge (ndsmerge on Linux and UNIX, NDSCons on Windows).
82	ndsCreateSubref	A subordinate reference is created. Example: 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.
83	ndsListPartitions	A List Partitions operation is performed. Example: Using ConsoleOne or iManager, from Partition and Schema view, click the eDirectory Server object to list the partitions held by the server.
84	ndsReadAttribute	A value of an attribute is read. Example: Perform a search operation on the tree.
85	ndsReadReferences	An entry’s references are read.

Trap Number	Trap Name	Trap Is Generated When
86	ndsUpdateReplica	An Update Replica operation is performed on a partition replica. Example: Delete a user from one of the servers. The other replica is updated for the delete operation.
87	ndsStartUpdateReplica	A Start Update Replica operation is performed on a partition replica. Example: Delete a user from one of the servers. The other replica is updated for the delete operation.
88	ndsEndUpdateReplica	An End Update Replica operation is performed on a partition replica. Example: Delete a user from one of the servers. The other replica is updated for the delete operation.
89	ndsSyncPartition	A Synchronize Partition operation is performed on a partition replica. Example: Delete a user from one of the partitions. The sync can be observed using DSTrace.
90	ndsSyncSchema	The master replica of the root receives a request to synchronize its schema with the server. Example: Add a new class using ConsoleOne > Wizard > Schema, LDAP tools, or ndssch utilities.
91	ndsCreateBackLink	A backlink is created. A backlink is created when an object not present locally is being referenced. Example: 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.
93	ndsChangeTreeName	The tree name is changed. Example: Using the merge utility DSMerge/ndsmerge to rename the tree.
94	ndsStartJoinPartition	A Start Join operation is performed to merge partitions. Example: Merge or join partitions using ConsoleOne or LDAP tools.

Trap Number	Trap Name	Trap Is Generated When
95	ndsAbortJoinPartition	A Join Partition operation is aborted to stop merge partition. Example: Merge or join partitions using ConsoleOne or LDAP tools.
96	ndsUpdateSchema	An Update Schema operation is performed. Example: Add a new class using ConsoleOne > *Wizard* > *Schema*, LDAP tools, or ndssch.
97	ndsStartUpdateSchema	A Start Update Schema operation is performed. Example: Add a new class using ConsoleOne > *Wizard* > *Schema*, LDAP tools, or ndssch.
98	ndsEndUpdateSchema	An End Update Schema operation is performed. Example: Add a new class using ConsoleOne > *Wizard* > *Schema*, LDAP tools, or ndssch.
99	ndsMoveTree	A Move Tree operation is performed. Example: Move a partition from one container to another.
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.

Trap Number	Trap Name	Trap Is Generated When
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.
111	ndsNewSchemaEpoch	The schema is reset using DSRepair. Example: Create a new schema epoch using `ndsrepair -S -Ad` on Linux and 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.

Trap Number	Trap Name	Trap Is Generated When
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: ◆ `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.
2002	ndsServerStop	The subagent loses its connection with the eDirectory server. This trap consists of two variables: ◆ `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. Example: Bring down the eDirectory server when the subagent is up and running.

Accessing the Encrypted Attributes

In eDirectory 8.8 and later, you can protect specific sensitive data when you store them on the disk and when you are trying to access them over the wire, by encrypting them. You can specify if you always need a secure channel to access the encrypted attributes or not. For more information, refer to Section 11.1.3, “Accessing the Encrypted Attributes,” on page 250.

When you have specified that you need only secure channels to access the encrypted attributes, NDS Value Events are blocked.Traps that are related to value events will have value data as `NULL` and you get an error, -6089, indicating that you need a secure channel to get the encrypted attributes value. Following are the traps which will have the value data as `NULL`:

- ndsAddValue
- ndsDeleteValue
- ndsDeleteAttribute

17.5.2 Configuring Traps

The method of configuring traps differs from platform to platform.

Platform	Utility
Windows	ndssnmpcfg
Linux and UNIX	ndssnmpconfig

Windows

The utility to configure traps on Windows is ndssnmpcfg. This utility is present in the *install_path*\ 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
-a	Fully Distinguished Name of a user having administrative rights
-c	Trap Commands (See “Windows Trap Commands” on page 467.)

Windows Trap Commands

Trap Commands	Description	Usage

DISABLE	Disabling a trap refers to the NMS not receiving traps although they are being generated.	To disable specific traps (for example, traps 10, 11, and 100):
		`ndssnmpcfg "DISABLE 10, 11, 100"`
		To disable all traps except 10, 11, and 100:
		`ndssnmpcfg "DISABLE ID != 10, 11, 100"`
		To disable all traps in the range 20 to 30:
		`ndssnmpcfg "DISABLE 20-29"`
		To disable all traps:
		`ndssnmpcfg "DISABLE ALL"`

Trap Commands	Description	Usage

ENABLE	Enabling a trap refers to the NMS receiving traps when they are generated.	`ndssnmpcfg "ENABLE trapSpec"`
		trapSpec can be any one of the following:
		To enable specific traps (for example, traps 10, 11, and 100):
		`ndssnmpcfg "ENABLE 10, 11, 100"`
		To enable all traps except 10, 11, and 100:
		`ndssnmpcfg "ENABLE ID != 10, 11, 100"`
		To enable all traps in the range 20 to 30:
		`ndssnmpcfg "ENABLE 20-29"`
		To enable all traps:
		`ndssnmpcfg "ENABLE ALL"`

INTERVAL	This utility is used to set and view the time interval.
	The time interval determines how many seconds to delay before sending duplicate traps.
	The time interval set should be between 0 and 2592000 seconds.
	If the time interval set is out of range, then the default time interval is considered.
	If the time interval is set to zero, all the traps are sent.
		To view the time interval:
		`ndssnmpcfg "213,240,79 INTERVAL"`
		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"`
		To view the default time interval:
		`ndssnmpcfg "DEFAULT INTERVAL"`
		To set the default time interval:
		`ndssnmpcfg "DEFAULT INTERVAL=10"`

Trap Commands	Description	Usage
LIST	Use this utility to view lists of trap numbers that meet specified criteria.	ndssnmpcfg LIST *trapSpec* *trapSpec* is used to specify groups of trap numbers and can be any of the following keywords: ALL, ENABLED, DISABLED, FAILED, or a logical expression Examples: To list all enabled traps along with trap names: ndssnmpcfg LIST ENABLED To list all disabled traps along with trap names: ndssnmpcfg LIST DISABLED To list all traps (117) along with trap names: ndssnmpcfg LIST ALL To list specific traps like 12, 224, and 300 along with trap names: ndssnmpcfg LIST ID = 12,224,300 To list all traps except selected traps like 12, 224, and 300 along with trap names: ndssnmpcfg LIST ID != 12,224,300 To list all traps which have been enabled for failure with trap names: ndssnmpcfg LIST FAILED

Trap Commands	Description	Usage
READ_CFG	Use this command to reconfigure the directory configuration from the configuration file ndstrap.cfg. Any changes specified in the configuration file will then take effect. This utility is primarily used to put various commands together in the ndstrap.cfg and do the operation in one instance. The ndstrap.cfg is located in *install directory*\SNMP The ndstrap.cfg 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, ndssnmpcfg is executed with the READ_CFG command.	ndssnmpcfg "READ_CFG"
FAILURE	This command is used to list all traps enabled for failure. Whenever an event fails, a failure trap is generated. **NOTE**: If the trap is enabled for failure and then disabled and again enabled using the enable trapid command, the trap is enabled for success and not for failure.	ndssnmpcfg "FAILURE *trapSpec*" *trapSpec* consists of one or more trap numbers separated by commas or spaces, the keyword ALL, or a logical expression.Examples: To set failure for multiple traps: ndssnmpcfg "FAILURE 10,11,100" To set failure for all traps except the traps mentioned: ndssnmpcfg "FAILURE ID != 24,30" To set failure for all traps: ndssnmpcfg "FAILURE ALL"

Linux and UNIX

The utility to configure traps on Linux and 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
`-p`	userFDN password for authentication
`-a`	Fully distinguished name of a user having administrative rights
`-c`	Trap commands (See “Linux and UNIX Trap Commands” on page 471.)

Linux and UNIX Trap Commands

Trap Commands	Description	Usage

`DISABLE`	Disabling a trap refers to the NMS not receiving traps though they are being generated.	To disable specific traps (for example, traps 10, 11 and 100):
		`ndssnmpconfig "DISABLE 10, 11, 100"`
		To disable all traps except 10, 11, and 100:
		`ndssnmpconfig "DISABLE ID != 10, 11, 100"`
		To disable all traps in the range 20 to 30:
		`ndssnmpconfig "DISABLE 20-29"`
		To disable all traps:
		`ndssnmpconfig "DISABLE ALL"`

`ENABLE`	Enabling a trap refers to the NMS receiving traps when they are generated.	`ndssnmpconfig "ENABLE trapSpec"`
		trapSpec can be any one of the following:
		To enable specific traps (for example, traps 10, 11, and 100):
		`ndssnmpconfig "ENABLE 10, 11, 100"`
		To enable all traps except 10, 11, and 100:
		`ndssnmpconfig "ENABLE ID != 10, 11, 100"`
		To enable all traps in the range 20 to 30:
		`ndssnmpconfig "ENABLE 20-29"`
		To enable all traps:
		`ndssnmpconfig "ENABLE ALL"`

Trap Commands	Description	Usage

INTERVAL	This utility is used to set and view the time interval.
	
	The time interval determines how many seconds to delay before sending duplicate traps.
	
	The time interval should be between 0 and 2592000 seconds.
	
	If the time interval is out of range, then the default time interval is considered.
	
	If the time interval is set to zero, all the traps are sent.
		To view the time interval:
		
		`ndssnmpconfig "213,240,79 INTERVAL"`
		
		To set the time interval between multiple traps (for example, to set the time interval between traps 12, 17, and 101 to 5):
		
		`ndssnmpconfig "12 17 101 INTERVAL 5"`
		
		To view the default time interval:
		
		`ndssnmpconfig "DEFAULT INTERVAL"`
		
		To set the default time interval:
		
		`ndssnmpconfig "DEFAULT INTERVAL=10"`

Trap Commands	Description	Usage

LIST	Use this utility to view lists of trap numbers that meet specified criteria.	ndssnmpconfig LIST <*trapSpec*>
		
		trapSpec is used to specify groups of trap numbers and can be any of the following keywords:
		
		ALL, ENABLED, DISABLED, FAILED, or a logical expression
		
		Examples:
		
		To list all enabled traps along with trap names:
		
		`ndssnmpconfig LIST ENABLED`
		
		To list all disabled traps along with trap names:
		
		`ndssnmpconfig LIST DISABLED`
		
		To list all traps (117) along with trap names:
		
		`ndssnmpconfig LIST ALL`
		
		To list specific traps like 12, 224, and 300 along with trap names:
		
		`ndssnmpconfig LIST ID = 12,224,300`
		
		To list all traps except selected traps like 12, 224, and 300 along with trap names:
		
		`ndssnmpconfig LIST ID != 12,224,300`
		
		To list all traps that have been enabled for failure with trap names:
		
		`ndssnmpconfig LIST FAILED`

Trap Commands	Description	Usage
READ_CFG	Use this command to reconfigure the directory configuration from the configuration file ndstrap.cfg. Any changes specified in the configuration file will then take effect. This utility is primarily used to put various commands together in the ndstrap.cfg file and perform the operation in one instance. The ndstrap.cfg file is located in /etc/ndssnmp/. The ndstrap.cfg 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 ndssnmpcfg is executed with the READ_CFG command.	ndssnmpconfig "READ_CFG"
FAILURE	This command is used to list all traps enabled for failure. Whenever an event fails, a failure trap is generated. **NOTE**: If the trap is enabled for failure and then disabled and again enabled using the enable trapid command, the trap is enabled for success and not for failure.	ndssnmpconfig "FAILURE *trapSpec*" *trapSpec* consists of one or more trap numbers separated by commas or spaces, the keyword ALL, or a logical expression. Examples: To set failure for multiple traps: ndssnmpconfig "FAILURE 10,11,100" To set failure for all traps except the traps mentioned: ndssnmpconfig "FAILURE ID != 24,30" To set failure for all traps: ndssnmpconfig "FAILURE ALL"

17.5.3 Statistics

- "ndsDbCache" on page 475
- "ndsDbConfig" on page 476
- "ndsProtoIfOps" on page 476
- "ndsServerInt" on page 477

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.
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

ndsProtoIfOps

Managed Objects in Directory	Description
ndsProtoIfSrvApplIndex	An index to uniquely identify the eDirectory Server Application.
ndsProtoIfIndex	An index to uniquely identify an entry corresponding to an eDirectory Server protocol interface.
ndsProtoIfDescription	Information on the port being used by the DS protocol interface.
ndsProtoIfUnauthBinds	Number of unauthenticated/anonymous bind requests received.
ndsProtoIfSimpleAuthBinds	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.
ndsProtoIfStrongAuthBinds	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.
ndsProtoIfBindSecurityErrors	Number of bind requests that have been rejected due to inappropriate authentication or invalid credentials.

Managed Objects in Directory	Description
ndsProtoIfInOps	Number of requests received from DUAs or other eDirectory servers.
ndsProtoIfReadOps	Number of read requests received.
ndsProtoIfCompareOps	Number of compare requests received.
ndsProtoIfAddEntryOps	Number of addEntry requests received.
ndsProtoIfRemoveEntryOps	Number of removeEntry requests received.
ndsProtoIfModifyEntryOps	Number of modifyEntry requests received.
ndsProtoIfModifyRDNOps	Number of modifyRDN requests received.
ndsProtoIfListOps	Number of list requests received.
ndsProtoIfSearchOps	Number of search requests (baseObject searches, oneLevel searches, and whole subtree searches) received.
ndsProtoIfOneLevelSearchOps	Number of oneLevel search requests received.
ndsProtoIfWholeSubtreeSearchOps	Number of whole subtree search requests received.
ndsProtoIfExtendedOps	Number of extended operations.
ndsProtoIfReferrals	Number of referrals returned in response to requests for operations.
ndsProtoIfChainings	Number of operations forwarded by this eDirectory server to other eDirectory servers.
ndsProtoIfSecurityErrors	Number of requests received that did not meet the security requirements.
ndsProtoIfErrors	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.
ndsProtoIfReplicationUpdatesIn	Number of replication updates fetched or received from eDirectory servers.
ndsProtoIfReplicationUpdatesOut	Number of replication updates sent to or taken by eDirectory servers.
ndsProtoIfInBytes	Incoming traffic, in bytes, on the interface. This includes requests from DUAs as well as responses from other eDirectory servers.
ndsProtoIfOutBytes	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.

Managed Objects in Directory	Description
ndsSrvIntProtoIfIndex	An index to uniquely identify an entry corresponding to an eDirectory server protocol interface.
ndsSrvIntIndex	Together with ndsSrvIntSrvApplIndex and ndsSrvIntProtoIfIndex, 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.

17.6 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” for more details.

Table 17-1 *Log File Location*

Platform	Subagent	Server	Master
Windows	*install_directory*\nds\snmp\dssnmpsa.log	*install_directory*\nds\snmp\dssnmpsrv.log	NA
Solaris	/var/opt/novell/eDirectory/log/ndssnmpsa.log	/var/opt/novell/eDirectory/log/ndsd.log	/var/adm/messages
Linux	/var/opt/novell/eDirectory/log/ndssnmpsa.log	/var/opt/novell/eDirectory/log/ndsd.log	/var/log/messages
AIX	/var/opt/novell/eDirectory/log/ndssnmpsa.log	/var/opt/novell/eDirectory/log/ndsd.log	/var/adm/messages

Table 17-1 lists the default location of the server log file for UNIX/Linux platforms. To know the location of the `ndsd.log` file, run the `ndsconfig get n4u.server.log-file` command against the eDirectory instance.

18 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

- Section 18.1, “Advanced Referral Costing,” on page 481
- Section 18.2, “Improving Bulkload Performance,” on page 490

Health Checks

- Section 18.3, “Keeping eDirectory Healthy,” on page 495
- Section 18.4, “Resources for Monitoring,” on page 497

Hardware Replacements

- Section 18.5, “Upgrading Hardware or Replacing a Server,” on page 498

eDirectory Recovery

- Section 18.6, “Restoring eDirectory after a Hardware Failure,” on page 504

18.1 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 18-1 *Advanced Referral Costing*

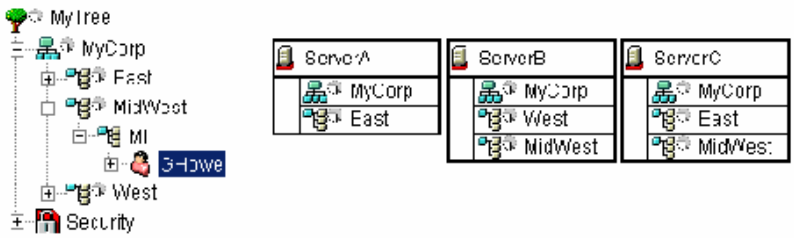


Figure 18-1 illustrates an LDAP subtree search to Server A for `cn=GHowe`, starting at `O=MyCorp`. However, the `cn=GHowe` object is located in the `ou=MidWest` partition, which is not represented on Server A.

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. Note that the process of choosing server 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.

The following sections provide information about how you can improve the performance of eDirectory servers:

- Section 18.1.1, "Improving Server-to-Server Connection," on page 482
- Section 18.1.2, "Advantages of Referral Costing," on page 484
- Section 18.1.3, "Deploying ARC," on page 485
- Section 18.1.4, "Enabling Advanced Referral Costing," on page 486
- Section 18.1.5, "Tuning Advanced Referral Costing," on page 486
- Section 18.1.6, "Monitoring Advanced Referral Costing," on page 487

18.1.1 Improving Server-to-Server Connection

Advanced Referral Costing (ARC) is an improved costing algorithm that is disabled by default. It is available in eDirectory 8.7.3.9 and eDirectory 8.8.2. The main purpose of ARC is to prevent server outages. Some of the benefits of ARC can include:

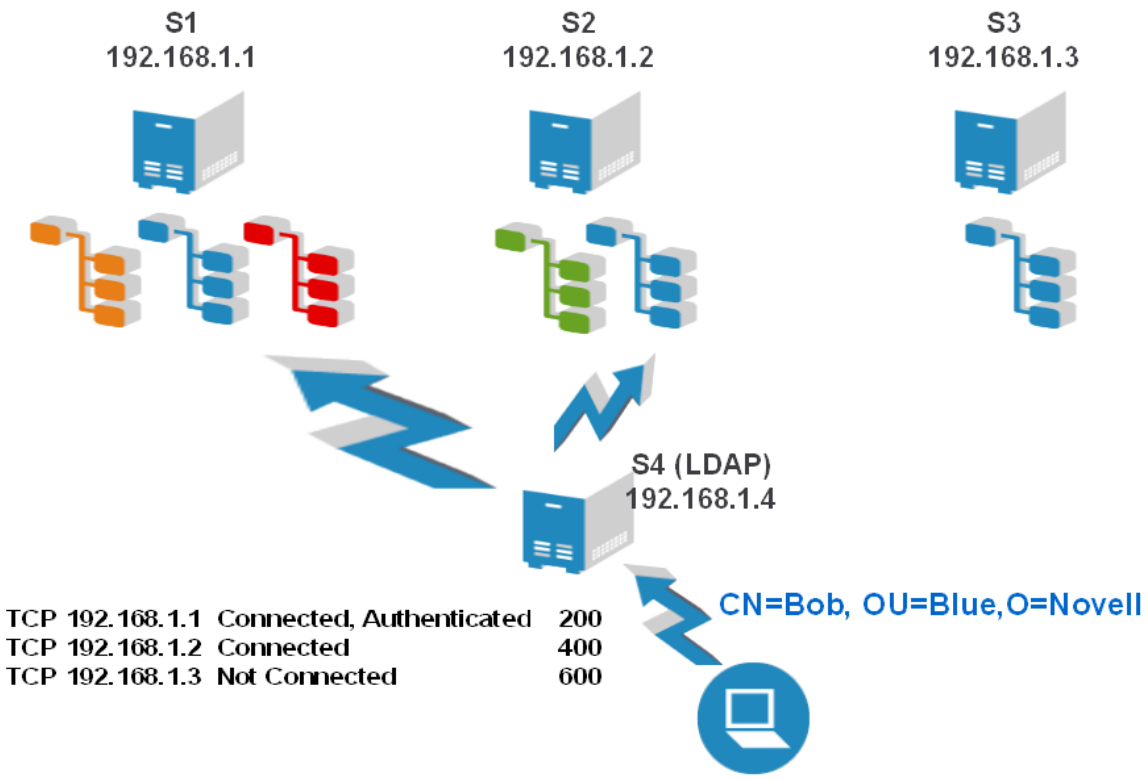
- Improved server performance and fault tolerance
- Better server-to-server communications
- Load distribution
- Remote server health monitoring
- Simplified isolation and identification of communication problems

Who Should Use ARC?

Servers that don't hold a local copy of an object or service need to walk the tree for information benefit from ARC, because they frequently communicate with the other servers. ARC is very effective in an LDAP environment, especially during prefer chaining.

For example, a server is sometimes overwhelmed by other servers that always make requests to that server, as illustrated in Figure 18-2.

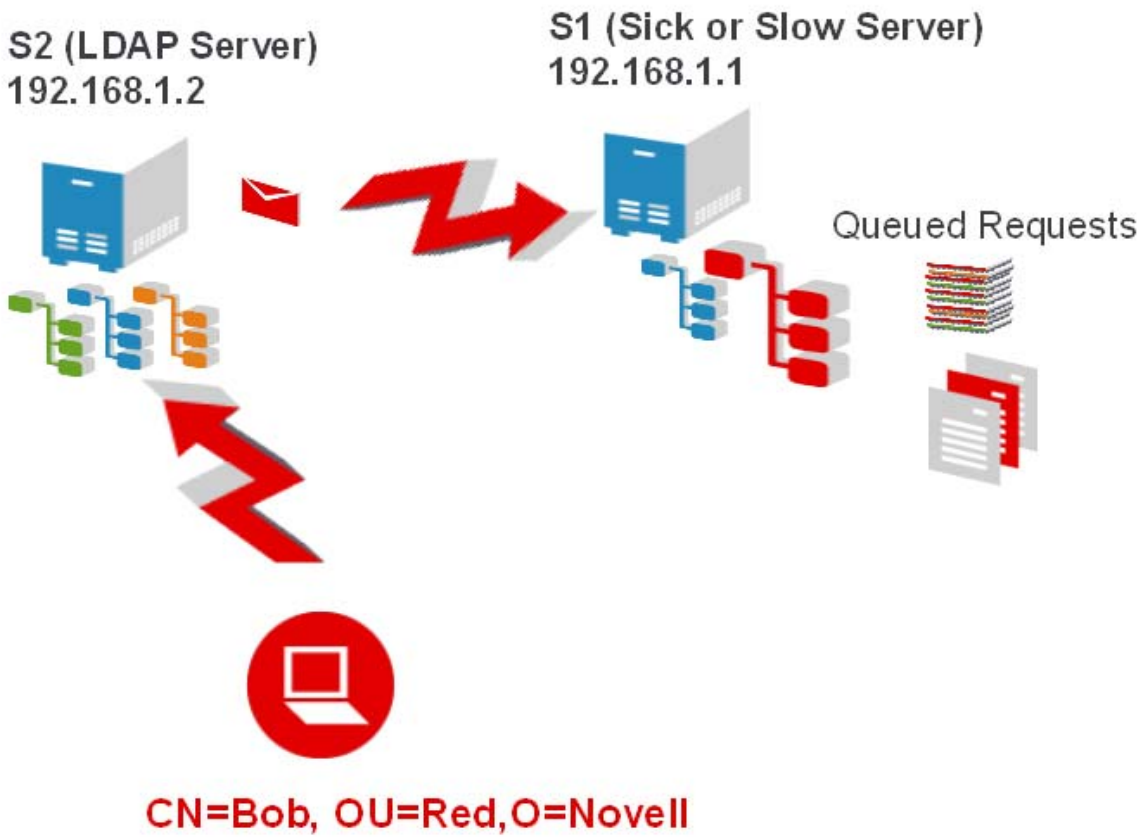
Figure 18-2 *One Stop Server Effect*



Although there are other available servers with replicas of the needed objects, servers still seem to prefer this server. This is because the servers making requests for a service or replica are are already connected to this server, so they tend to send all the requests that the server can handle. Figure 18-2 shows that all requests from S4 are going to S1. This is because S4 was already connected and authenticated to S1, so it continues to send all the requests for the blue partition to S1, even though S2 and S3 could service those requests. ARC helps to eliminate these situations by distributing the load to the servers that respond faster. You should enable ARC on remote servers (S4) that request this server, or you can enable ARC on all servers.

Figure 18-3 shows another scenario, illustrating the "cascading server" effect. Here, server S1 is often not responding, but it is not down. If the S1 were down, the requests would time out and communication would stop. If the server is still up at the transport level, but the database is slow or busy, the server continues to accept and queue new requests from other servers. This can cause the additional servers (S2) to eventually run out of threads. Each outstanding request takes a thread on the remote server, and when they run out of threads the server becomes non-responsive. ARC resolves this issue by distributing requests across the fastest servers, because a server that is slow or sick incurs a higher cost in servicing requests.

Figure 18-3 *Cascading Server Effect*



In addition, ARC is a good choice for improving fault tolerance. It has the ability to easily identify server communication problems.

18.1.2 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.

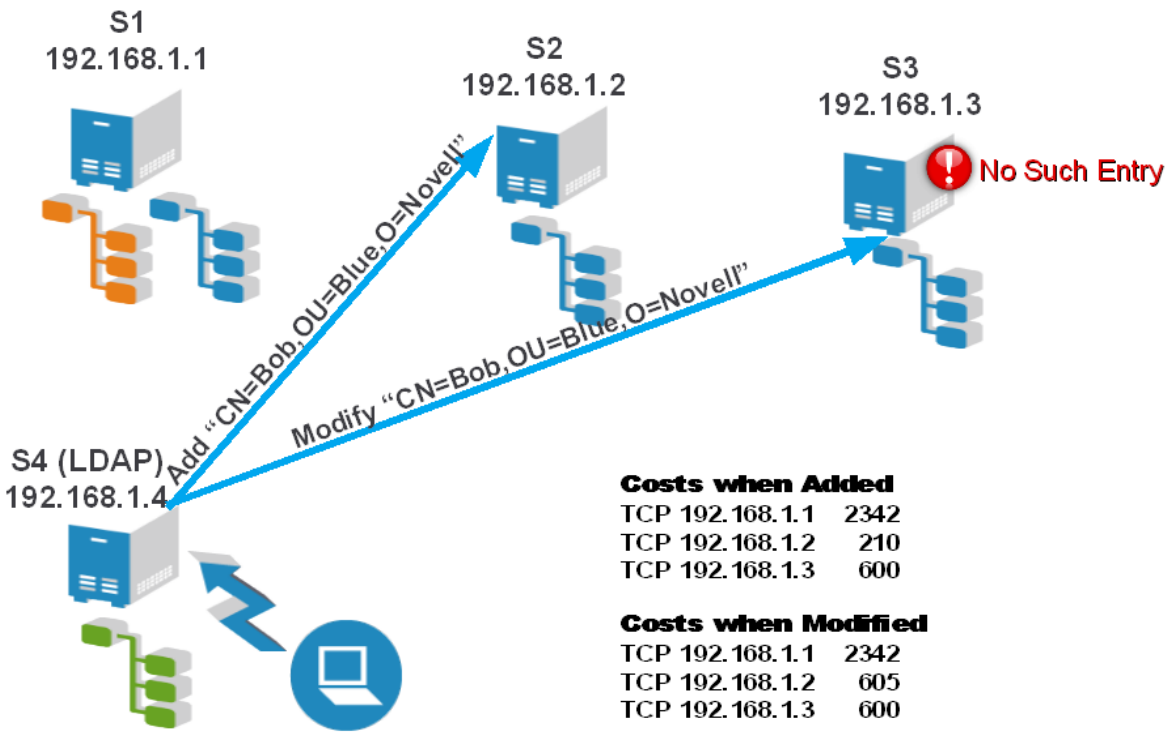
18.1.3 Deploying ARC

ARC is usually deployed on a server-to-server basis. Those servers that are ARC - enabled can know the new costing information. You should patch all the servers to 8.7.3.9 ftf3 and eDirectory 8.8 servers to 8.8.2 version and then enable ARC on each server in the environment.

Deployment Considerations

It is not useful to enable ARC on all servers. Figure 18-4 shows a situation that could impact the efficiency of LDAP servers. In the figure, S4 holds a copy of the green partition, but not of the blue partition. Any chaining LDAP request that requires information from the blue partition needs to walk to the S1, S2, or S3 servers to be fulfilled. This works in most cases, and ARC is designed for just such situations.

Figure 18-4 *ARC Deployment Considerations*



However, performing specific LDAP operations could be difficult. Although it is possible to add a user, for example, Bob.Blue.Novell, the operation might fail when you try to immediately return to modify Bob. The figure shows Bob added on S2, but modifying Bob on S3 has failed because S3 has not yet synchronized with S2, so S3 has not yet received Bob. ARC has the capability to direct you to a different server, because ARC is more dynamic than the original costing method.

This configuration works well in scenarios where the server costs don't vary much and they don't have problems synchronizing. Disabling ARC on S4 resolves this issue.

18.1.4 Enabling Advanced Referral Costing

NDSTrace

Use the NDSTrace tool to enable ARC on all UNIX platforms.

Table 18-1 *Enabling ARC on UNIX Platform*

`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://support.novell.com/support_options.html)

18.1.5 Tuning Advanced Referral Costing

ARC requires no tuning by default. However, there are tunable parameters in ARC that can be used to change how ARC functions, or to disable or enable certain features. There are 3 major components to ARC.

Advanced Costing

When asked to cost a given address, ARC uses the information known about the connection to calculate the cost of the given referral. If ARC is on, Advanced Costing is always used when costing a referral.

Background Monitoring

A background thread periodically checks the timer information to ensure that it is current. When a server is slow, its cost rises and there is a good chance that communication will cease. The background thread periodically (once a minute by default) checks to see if a server in the table has not been updated. If the server is has not been updated in the last three minutes, the server makes a resolve name request on its behalf to check the server's health. This creates current costing for the server, and also detects if a server is now less busy, or is healthy, so a client doesn't need to suffer adverse effects to check the server's health. There are two permanent configuration parameters that can be changed for the background thread:

- **ARC_MAX_WAIT:** How stale a timer is before a request to the server to check its health (180 seconds by default).
- **ARC_BG_INTERVAL:** How often the background thread runs (60 seconds by default,: 0 means disabled and the thread doesn't run).

For additional information, see section 8.4.24 setting permanent configuration parameters.

Remote Health Information

Servers using ARC periodically request health information from a remote server. These are not additional requests on the wire, but additional health information that is returned in standard resolve name requests that servers frequently make. This information is then used in the costing algorithm to further enhance reactions to servers that are under heavy loads. When a resolve name request is being made to a remote server, if it has been more than 15 seconds since the last update, health information is requested from the remote server and is added to the reply of the resolve name request.

There is one tunable parameter for Remote Health Monitoring:

- **ARC_DS_INFO_INTERVAL:** This is how often to request lock (health) information in ARC (15 seconds by default).

18.1.6 Monitoring Advanced Referral Costing

You can print the ResolveTimes table to observe Advanced Referral Costing in action.

Use the following commands to print the ResolveTimes table:

- `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.

Using ARC for Troubleshooting

One of the most useful features of ARC is the ability to quickly identify communication problems with servers.

The following is an example of a ResolveTimesTable printout:

ARC is currently enabled.

Table 18-2 *Resolve Time Costs*

Slot	Transport Address	Cost	LastUse	Checked	#Req	waiters	LockTime
1	tcp:151.155.134.27:524	214	14	14	0	0	0
2	tcp:151.155.134.11:524	0	0	0	0	0	0
3	udp:151.155.134.11:524	0	0	0	0	0	0
4	cp:151.155.134.13:524	554759	280	0	0	27	582
5	tcp:151.155.134.59:524	0	179	179	0	0	0
6	udp:151.155.134.59:524	0	119	119	0	0	0
7	tcp:151.155.134.28:524	1543	119	119	0	0	0
8	tcp:151.155.134.15:524	124	14	14	0	0	0

The printout shows that from this server's perspective, 151.155.134.13 is having difficulties. You can also see that the problem is most likely the server, not the transport. The server has 27 requests waiting for access to the database, and the requests are taking a long time to acquire the database lock. This server has two requests that have never received replies from the remote server.

You can also see that 151.155.134.11 and 151.155.134.59 are either very fast servers, or are not very busy, or both. You can see that 151.155.134.59 and 151.155.134.11 have both had problems communicating via TCP at one time, but are both healthy now, because they both have UDP connections. UDP connections to a server are tried only if there is a problem talking to the server via TCP.

The following is a summary of what each number means:

Transport Address: The address of the remote server.

Cost: The current cost of the remote server.

Last Use: The duration in seconds since last communication with the server.

Checked: The duration in seconds since last health information from the remote server.

#Req: The number of outstanding requests to the remote server.

Waiters: The number of requests to the remote server waiting for the database lock.

LockTime: Duration that a process has held the database lock on the remote server.

The following printout has another example of quickly identifying a communications problem, because you can see that the server currently cannot communicate to 151.155.134.13 via TCP.

ARC is currently enabled.

Table 18-3 *Resolve Time Costs*

Slot	Transport Address	Cost	LastUse	Checked	#Req	waiters	LockTime
1	tcp:151.155.134.27:524	394	92	14	0	0	0
2	tcp:151.155.134.11:524	0	0	0	0	0	0
3	udp:151.155.134.11:524	0	0	0	0	0	0
4	tcp:151.155.134.13:524	5000000	180	180 is in BAD ADDRESS CACHE			

There are a few things to keep in mind when looking at these tables:

- Outstanding requests are not necessarily bad, because the server might just be servicing many requests. Outstanding requests on servers where costing is high are a problem.
- Your first indicator of a server's health is the current cost, making it easy to see what server is causing you problems.

NOTE: All requests are timing round trip time, and how long requests are outstanding. This means transport times are also a component of the cost. If a server shows up as having problems in this table, but is working well from other servers, and doesn't appear to have a problem, this might indicate a transport issue.

Background Thread Traces

The following is a trace showing the ARCBackgroundResolveTimerThread running:

```
ARCBackGroundResolveTimerThread started Interval = 60 MaxWait = 180000

Updating timer info for tcp:151.155.134.11:524

Updating timer info for udp:151.155.134.11:524

Updating timer info for tcp:151.155.134.13:524 ARCBackGroundResolveTimerThread
error -635 in DCConnectToAddress for tcp:151.155.134.59:524

ARCBackGroundResolveTimerThread completed in 0 seconds

8-total timers 4-stale timers 3-timers updated
```

From the above message you can see the following:

- TCP:151.155.134.11 has not been used for more than 3 minutes
- UDP:151.155.134.11 has not been used for more than 3 minutes
- TCP: 151.155.134.13 has not been used for more than 3 minutes

The timer information was updated for all of the above servers, with the following results:

- TCP: 151.155.134.59 is still not reachable from this server.

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 by running the command `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. The remaining tags are turned off using the `set NDSTrace =nodebug` command.

```
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.

18.2 Improving Bulkload Performance

eDirectory 8.8 provides you with new options to increase the bulkload performance.

The following are the tunable parameters for bulkload performance using the Novell Import Convert Export (ICE) utility.

- Section 18.2.1, "eDirectory Cache Settings," on page 490
- Section 18.2.2, "LBURP Transaction Size Setting," on page 491
- Section 18.2.3, "Increasing the Number of Asynchronous Requests in ICE," on page 491
- Section 18.2.4, "Increased Number of LDAP Writer Threads," on page 492
- Section 18.2.5, "Disabling ACL Templates," on page 492
- Section 18.2.6, "Disabling Schema Validation in ICE," on page 492
- Section 18.2.7, "Disabling ACL Templates," on page 492
- Section 18.2.8, "Backlinker," on page 494
- Section 18.2.9, "Enabling/Disabling Inline Cache," on page 494
- Section 18.2.10, "Increasing the LBURP Time Out Period," on page 494

Also refer to the various operating system tunable parameters.

18.2.1 eDirectory Cache Settings

To optimize the bulkload performance, allocate a higher percentage of the eDirectory cache for block cache.

For more details refer to "Tuning eDirectory Subsystems" (http://www.novell.com/documentation/edir88/edir88tuning/data/bqebx8t.html) in the *Novell eDirectory 8.8 SP7 Tuning Guide for UNIX Platforms* (http://www.novell.com/documentation/edir88/edir88tuning/data/front.html).

18.2.2 LBURP Transaction Size Setting

The LBURP transaction size sets the number of records that are 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 between 1 and 350.

Modifying the Transaction Size

To modify the transaction size, modify the required value for the n4u.ldap.lburp.transize parameter in `/etc/opt/novell/eDirectory/conf/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.
- The LDIF file should be free of errors and any entries already existing in eDirectory should be 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 ignores the LBURP transaction setting and performs a commit after each operation to ensure data integrity.

  See "Debugging LDIF Files" for more information.
- LBURP optimization works only for leaf objects. If the transaction contains both a container and its subordinate objects, eDirectory treats this as an error. To avoid this, we recommend loading the container objects first using a separate LDIF file or enables the use of forward references.

  See "Enabling Forward References" in the *Novell eDirectory 8.8 SP7 Troubleshooting Guide* for more information.

18.2.3 Increasing the Number of Asynchronous Requests in ICE

This refers to the number of entries the ICE client can send to the LDAP server asynchronously before waiting for any result back from the server.

The number of asynchronous requests can be set between 10 and 200. The default value is 100. Any value less than the minimum value (10) would fallback to the default. The minimum value is appropriate for small LDIF files.

In ideal scenarios, a higher window size ensures faster performance. However, the window size must not be set to arbitrarily high values because a larger window size requires the client to allocate more memory to process the entries in the LDIF file. If the system is running low on memory, this can cause a slowdown due to swapping.

You can modify the number of asynchronous requests in ICE using either the ICE command line option or iManager.

Using ICE Command Line Option

The number of asynchronous requests can be specified using the ICE command line option `-z`. This is available as part of the LDAP destination handler.

To set the number of asynchronous requests sent by the ICE client to 50, you would enter the following command:

```
ice -SLDIF -f LDIF_file -a -c -DLDAP -d cn=admin,o=novell -Z50 -w password
```

Using iManager ICE Wizard

To set the number of asynchronous requests sent by the ICE client through iManager:

1. Click the *Roles and Tasks* button .
2. Click *eDirectory Maintenance* > *Import Convert Export Wizard*.
3. Type the value in the *LBURP Window Size* field in the LDAP Destination Handler screens in both the Importing Data from a File and Migrating Data between LDAP Servers tasks.
4. Click *Next*.

   For more information, refer to the help provided in the Wizard.

18.2.4 Increased Number of LDAP Writer Threads

The LDAP server now has multiple writer threads. Use the `-F` ICE command line option for enabling forward referencing to avoid any possible errors due to concurrent processing as follows:

```
ice -SLDIF -f LDIF_file -a -c -DLDAP -d cn=admin,o=novell -w password  -F
```

18.2.5 Disabling ACL Templates

See the *Novell eDirectory 8.8 SP7 Tuning Guide for UNIX Platforms* (http://www.novell.com/documentation/edir88/edir88tuning/data/front.html) for information on how to disable ACL templates.

18.2.6 Disabling Schema Validation in ICE

Use the `-C` and `-n` ICE command line options to disable schema validation at the ICE client as follows:

```
ice -C -n -SLDIF -f LDIF_file -a -c -DLDAP -d cn=admin,o=novell -w password
```

18.2.7 Disabling ACL Templates

You can disable the Access Control List (ACL) templates to increase the bulkload performance. The implication of this is that some of the ACLs will be missing. However, you can resolve this by adding the required ACLs to the LDIF file or applying them later.

1. Run the following command:

```
ldapsearch -D cn_of_admin -w password -b cn=schema -s base
objectclasses=inetorgperson
```

   The output of this command would be similar to the following:

```
dn: cn=schemaobjectClasses: ( 2.16.840.1.113730.3.2.2 NAME 'inetOrgPerson' SUP
 organizationalPerson STRUCTURAL MAY ( groupMembership $ ndsHomeDirectory
 $ loginAllowedTi meMap $ loginDisabled $ loginExpirationTime $
 loginGraceLimit $ loginGraceRem aining $ loginIntruderAddress $
 loginIntruderAttempts $ loginIntruderResetTim e $
 loginMaximumSimultaneous $ loginScript $ loginTime $
 networkAddressRestri ction $ networkAddress $ passwordsUsed $
 passwordAllowChange $ passwordExpirationInterval $
 passwordExpirationTime $passwordMinimumLength $ passwordRequired $
 passwordUniqueRequired $ printJobConfiguration $ privateKey $ Profile $
 publicKey $ securityEquals $ accountBalance $ allowUnlimitedCredit $
 minimum AccountBalance $ messageServer $ Language $ UID $
 lockedByIntruder $ serverHolds $ lastLoginTime $ typeCreatorMap $
 higherPrivileges $ printerControl $ securityFlags $ profileMembership $
 Timezone $ sASServiceDN $ sASSecretStore $ sASSecretStoreKey $
 sASSecretStoreData $ sASPKIStoreKeys $ userCertificate
 $nDSPKIUserCertificateInfo $ nDSPKIKeystore $ rADIUSActiveConnections $
 rADIUS AttributeLists $ rADIUSConcurrentLimit $ rADIUSConnectionHistory
 $ rADIUSDefa ultProfile $ rADIUSDialAccessGroup $ rADIUSEnableDialAccess
 $ rADIUSPassword $ rADIUSServiceList $ audio $ businessCategory $
 carLicense $ departmentNumbe r $ employeeNumber $ employeeType $
 givenName $ homePhone $ homePostalAddress  $ initials $ jpegPhoto $
 labeledUri $ mail $ manager $ mobile $ pager $ ldap Photo $
 preferredLanguage $ roomNumber $ secretary $ uid $ userSMIMECertifica te
 $ x500UniqueIdentifier $ displayName $ userPKCS12 ) X-NDS_NAME 'User' X
-NDS_NOT_CONTAINER '1' X-NDS_NONREMOVABLE '1' X-NDS_ACL_TEMPLATES (
'2#subtree#[Self]#[All Attributes Rights]' '6#entry#[Self]#loginScript'
'1#subtree#[Root Template]#[Entry Rights]' '2#entry#[Public]#messageServer'
'2#entry#[Root Template]#groupMembership'
'6#entry#[Self]#printJobConfiguration' '2#entry#[Root
Template]#networkAddress') )
```

2. In the output noted in the previous step, delete the information marked in bold.
3. Save the revised output as an LDIF file.
4. Add the following information to the newly saved LDIF file:

```
dn: cn=schemachangetype: modifydelete: objectclassesobjectclasses: (
2.16.840.1.113730.3.2.2 )-add:objectclasses
```

Therefore, your LDIF should now be similar to the following:

```
dn: cn=schemachangetype: modifydelete: objectclassesobjectclasses: (
2.16.840.1.113730.3.2.2 )-add:objectclassesobjectClasses: (
2.16.840.1.113730.3.2.2 NAME 'inetOrgPerson' SUP
 organization alPerson STRUCTURAL MAY ( groupMembership $ ndsHomeDirectory
 $ loginAllowedTimeMap $ loginDisabled $ loginExpirationTime $
 loginGraceLimit $ loginGraceRem aining $ loginIntruderAddress $
 loginIntruderAttempts $ loginIntruderResetTime $
 loginMaximumSimultaneous $ loginScript $ loginTime $
 networkAddressRestri ction $ networkAddress $ passwordsUsed $
 passwordAllowChange $ passwordExpirationInterval $
 passwordExpirationTime $ passwordMinimumLength $ passwordRequired
 $passwordUniqueRequired $ printJobConfiguration $ privateKey $ Profile $
 publicKey $ securityEquals $ accountBalance $ allowUnlimitedCredit $
 minimum AccountBalance $ messageServer $ Language $ UID $
 lockedByIntruder $ serverHolds $ lastLoginTime $ typeCreatorMap $
```

```
higherPrivileges $ printerControl $ securityFlags $ profileMembership $
Timezone $ sASServiceDN $ sASSecretStore $ sASSecretStoreKey $
sASSecretStoreData $ sASPKIStoreKeys $ userCertificate $
nDSPKIUserCertificateInfo $ nDSPKIKeystore $ rADIUSActiveConnections $
rADIUSAttributeLists $ rADIUSConcurrentLimit $ rADIUSConnectionHistory $
rADIUSDefa ultProfile $ rADIUSDialAccessGroup $ rADIUSEnableDialAccess
$rADIUSPassword $ rADIUSServiceList $ audio $ businessCategory $
carLicense
$ departmentNumbe r $ employeeNumber $ employeeType $ givenName $
homePhone $ homePostalAddress  $ initials $ jpegPhoto $ labeledUri $ mail
$ manager $ mobile $ pager $ ldap Photo $ preferredLanguage $ roomNumber
$ secretary $ uid $ userSMIMECertifica te $ x500UniqueIdentifier $
displayName $ userPKCS12 ) X-NDS_NAME 'User' X-ND S_NOT_CONTAINER '1' X
-NDS_NONREMOVABLE '1' )
```

5 Enter the following command:

```
ldapmodify -D cn_of_admin -w password -f LDIF_file_name
```

See the *Novell eDirectory 8.8 SP7 Tuning Guide for UNIX Platforms* (http://www.novell.com/documentation/edir88/edir88tuning/data/front.html) for more information on working with ACLs.

18.2.8 Backlinker

Backlinker is a background process that checks the referential integrity among other checks runs 50 minutes after the eDirectory server comes up. The subsequent time it runs is after 13 hours. Ensure that backlinker does not run during the bulkload process. In case backlinker runs, depending on the time and the number of objects loaded, backlinker can hinder the bulkload.

18.2.9 Enabling/Disabling Inline Cache

You can enable or disable the Inline Change Cache for a server. You can disable Inline Change Cache only when Outbound Synchronization is disabled. Enabling Outbound Synchronization also enables Inline Change Cache.

Disabling Inline Change Cache marks the change cache as invalid for this replica and tags it with an invalid flag in *Agent Configuration* > *Partitions*. Enabling Inline Change Cache removes the invalid change cache flag when the change cache is rebuilt.

18.2.10 Increasing the LBURP Time Out Period

By default, the time out period for a client is 20 minutes (1200 seconds). But during bulkload, with the LBURP transaction size as high as 250, objects with large number of attributes with huge values for these attributes, and with LBURP concurrent processing enabled at the server, the server gets busy processing data pumped in by the ICE client without responding to the client in the stipulated time. This times out the ICE client.

Therefore, we recommend you to increase the time out period. You can do this by exporting the environment variable LBURP_TIMEOUT with high values (in seconds).

For example, to export the LBURP_TIMEOUT variable with 1200 seconds, enter the following:

```
export ICE_LBURP_TIMEOUT=1200
```

18.3 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.

18.3.1 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.

18.3.2 Health Check Overview

A complete health check includes checking the following:

- eDirectory version

  Running different versions of NDS or eDirectory on the same server can cause synchronization problems. If your version of NDS or eDirectory is outdated, download the latest software patch from the Patches & Security Web site (http://support.novell.com/patches.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, Section 18.3.3, "Checking eDirectory Health Using iMonitor," on page 496.

18.3.3 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 Section 8.2, "Accessing iMonitor," on page 201.

2. In the Navigator frame, click the Reports icon .
3. In the Assistant frame, click the *Report Config* link.

   A Runable 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 Section 8.2, "Accessing iMonitor," on page 201.

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.

18.3.4 For More Information

The tools and techniques used to keep eDirectory healthy are documented in the Novell eDirectory Tools & Diagnostics Course 3007. 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).

18.4 Resources for Monitoring

The Novell DSTrace utility runs on Windows, Linux, Solaris, and AIX. This tool helps you monitor the vast resources of eDirectory. For more information on DSTrace, see the following:

- "Configuring Trace Settings" on page 210
- "Looking Into the Directory Services Trace (DSTrace) Options" (http://support.novell.com/techcenter/articles/anp20010801.html)
- "More on Using the DSTrace Command" (http://support.novell.com/techcenter/articles/anp20010901.html)

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:

- Symantec (http://www.symantec.com/compliance/)
- Blue Lance (http://www.bluelance.com)
- Quest (http://www.quest.com/active-directory/)

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.

18.5 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 391.

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 498
- "Planned Replacement of a Server" on page 500

18.5.1 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 501.

2. Use a Client command like the following to do a cold backup of the eDirectory database and keep the database closed and locked when finished. If you use NICI, make sure to back up the security files too.

   ```
   backup -f backup_filename_and_path
   -l log_filename_and_path -t -c -o -d
   ```

If you use NICI, make sure you back up the NICI files. See "Backing Up Manually with DSBK" on page 415 and "Backup and Restore Command Line Options" on page 419 for more information about using the 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.

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	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	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: `restore -r -f` *`backup_filename_and_path`* `-l` *`log_filename_and_path`*. Add the `-u` option if you backed up the files listed in an include file and restore NICI files seperately. 5. Unlock the eDirectory database. 6. If you restored NICI 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 ConsoleOne® 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 500.

18.5.2 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 501.
2. Do the eDirectory and file system backups on Server A as described in "2. Creating a Backup of eDirectory" on page 502.
3. Transfer the information to Server B as described in "3. Restoring eDirectory Information for a Server Replacement" on page 502.

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 502 will put Server B into the original tree that Server A was in.)

Continue with the steps in the next section, "2. Creating a Backup of eDirectory" on page 502.

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 501, use the 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 501.
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 Client with the `-c`, `-o`, and `-d` switches:

```
backup -f backup_filename_and_path -l log_filename_and_path -t -c -o -d
```

   If you use NICI, make sure you back up the NICI files. See "Backing Up Manually with DSBK" on page 415 and "Backup and Restore Command Line Options" on page 419 for more information about using the 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.

   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/oes/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 502.

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 501 and "2. Creating a Backup of eDirectory" on page 502.
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 command line client, use a command like the following:

```
restore -r -f backup_filename_and_path -l log_filename_and_path
```

If you use NICI, make sure you restore the NICI files. Add the `-u` option if you backed up files listed in an include file. See "Restoring from Backup Files with DSBK" on page 417 and "Backup and Restore Command Line Options" on page 419 for more information about using the 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.

3c Transfer Server A's file system data onto Server B, from backup.

4 If you use NICI, restart the server to reinitialize NICI so it will use the restored NICI security files.

5 Unlock the eDirectory database.

6 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 NICI 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.

7 (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.

18.5.3 Server IP Address Changes

Usually the server's IP address is static. When it changes you need to update the `nds.conf` file for all the eDirectory instances with the new IP address. `nds.conf` should use the interface name instead of IP address if the IP address changes frequently.

For example: `n4u.server.interfaces=eth0@1524`

After an IP address change, a server's IP-based Key Material Objects (KMO) will not be automatically updated. Though deleting the old KMOs (with IP in their name) is not necessary, it helps to keep the tree clean. Run the `ndsconfig upgrade` command to recreate your KMOs and link them with the NCP Server and LDAP Server objects.

NOTE: Running `ndsconfig upgrade` restarts your eDirectory instance.

Now the server continues to listen on the new address. Run DSRepair network repair options if there are multiple servers in the tree:

```
ndsrepair -N
```

After running the repair options, restart the eDirectory server.

For more information on server IP address changes, refer to TID# 3201067 (http://www.novell.com/support/search.do?cmd=displayKC&docType=kc&externalId=3201067&sliceId=SAL_Public&dialogID=36008849&stateId=0%200%2036014447)

18.6 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 Section 16.4, "Preparing for a Restore," on page 408 and "Restoring from Backup Files with iManager" on page 537 (or "Restoring from Backup Files with DSBK" on page 417).

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

- We recommend you exclude the DIB directory on your eDirectory server from any antivirus or backup software processes. Use the eDirectory Backup Tool to back up your DIB directory. For more information about backing up eDirectory, see "Backing Up and Restoring Novell eDirectory" on page 391.
- 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 is available from the Novell Support Web site (http://support.novell.com/docs/Readmes/InfoDocument//2960653.html).

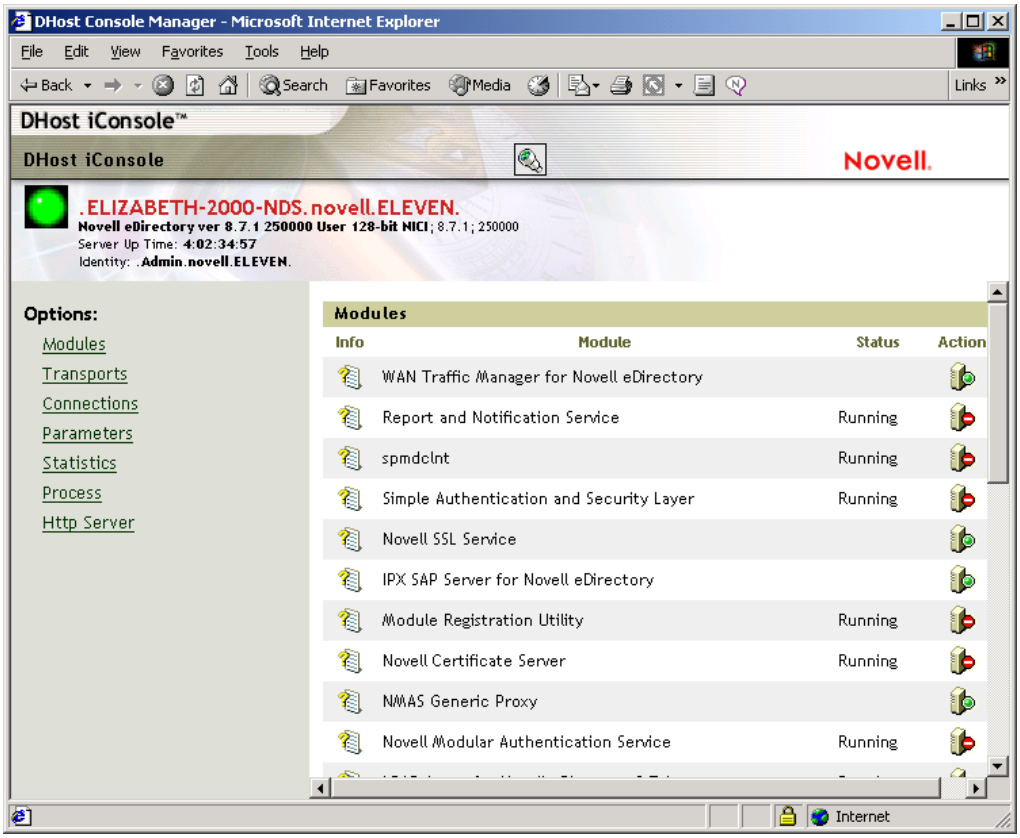
---

19 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 19-1 *DHost iConsole Manager*



This chapter contains the following information:

- Section 19.1, “What is DHost?,” on page 506
- Section 19.2, “Running DHost iConsole,” on page 506
- Section 19.3, “Managing eDirectory Modules,” on page 507

- Section 19.4, “Querying for DHost Information,” on page 509
- Section 19.5, “Process Stack,” on page 511

19.1 What is DHost?

Novell eDirectory software for Windows, Solaris, Linux, and AIX is all built upon the same core code. In order for eDirectory for Windows, Linux, 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 that the NCP provides naturally.

DHost provides the following services:

Service	Description
NCP Engine	A packet-based protocol that enables a client to send requests to and receive replies from an eDirectory server. For more information, see the NetWare Core Protocols NDK (http://developer.novell.com/documentation/ncp/index.html).
Watchdog	Packets used to make sure workstations are still connected to the eDirectory server. For more information, see “Watchdog Packet Spoofing”.
Connection Table	A unique number assigned to any process, print server, application, workstation, or other entity that attaches to an eDirectory 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.
Event System	Provides a way for applications to monitor the activity of an individual server.
Thread Pool	A sequence of instructions executed as an independent entity and scheduled by system software.
NCP Extensions	Allows server application developers to write NLM™ software to be implemented as NCPs. For more information, see “NCP Extension” (http://developer.novell.com/documentation/ncp/ncp__enu/data/alne6tm.html) in the NCP NDK.
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.

19.2 Running DHost iConsole

- Section 19.2.1, “Running DHost iConsole on Windows,” on page 507
- Section 19.2.2, “Running DHost iConsole on Linux, Solaris, and AIX,” on page 507

19.2.1 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 user name, context, and password.

19.2.2 Running DHost iConsole on Linux, Solaris, and AIX

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 user name, context, and password.

19.3 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 Windows" on page 508
- "Loading or Unloading Modules on Linux, Solaris, and AIX" on page 508

For more information on using Novell iManager to load and unload eDirectory services, see Section 6.4, "eDirectory Service Manager," on page 189.

19.3.1 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 user name, context, and password.
4. Click *Modules*.
5. Click to load a module, or to unload a module.

19.3.2 Loading or Unloading Modules on Linux, Solaris, and AIX

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 user name, context, and password.

4 Click *Modules*.

5 Click to load a module, or to unload a module.

19.4 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

19.4.1 Viewing the Configuration Parameters

Configuration parameters are specific only to Linux and UNIX platforms.

In the DHost iConsole Manager, click *Parameters*. See "Running DHost iConsole on Linux, Solaris, and AIX" on page 507 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 8.8 SP7 Installation Guide*.

19.4.2 Viewing Protocol Information

In the DHost iConsole Manager, click *Transports*.

The following protocol information is displayed:

- ID

- Protocol
- Transports

19.4.3 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

19.4.4 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

19.5 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 user name, context, and password.
4. Click *Process*.
5. To view the call stack for a thread, click the thread ID.

20 Setting the sadmin Password

You can set up a preconfigured admin user that 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.

Use the ndspassstore utility to set the sadmin password on Windows, Linux, Solaris, and AIX systems.

Enter the following at the server console:

```
ndspassstore -a sadmin -w <password>
```

where `sadmin` (admin context) is the fully distinguished name of a user having administrative rights and `password` is the password for authentication. Select an appropriate instance in case of a multi-instance scenario.

Example: `ndspassstore -a sadmin -w pass`

ndspassstore is available by default at *C:\Novell\NDS* in Windows and at */opt/novell/eDirectory/bin* in UNIX.

21 The eDirectory Management Toolbox

The Novell eDirectory Management Toolbox (eMBox) lets you access all of the eDirectory back-end utilities remotely, as well as on the server.

The eMBox works with Novell iManager to provide Web-based access to eDirectory utilities such as DSRepair, DSMerge, Backup and Restore, and Service Manager.

IMPORTANT: For all the users including the administrator, Role Based Services must be configured through iManager to the tree that is to be administered for tasks to be run.

Role Based Services must be configured for the following tasks that are under eDirectory maintenance menu in the iManager:

- Backup Configuration
- Graft Tree
- Repair eDirectory
- Repair Server
- Repair Sync
- Replica Repair
- Replica Ring Repair
- Restore
- Schema Maintenance
- Service Manager
- Merge Tree
- Rename Tree

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 Client.

For all eDirectory Management Tools (eMTools), to run, including Backup, DSRepair, DSMerge, Schema Operations, and eDirectory Service Manager, eMBox must be loaded and running on the eDirectory server.

In this section:

- Section 21.1, “Using the Command Line Client,” on page 516
- Section 21.2, “Using the Logger,” on page 523
- Section 21.3, “Using the eMBox Client for Backup and Restore,” on page 525
- Section 21.4, “Using Novell iManager for Backup and Restore,” on page 533

21.1 Using the Command Line Client

One way to access 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 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.

NOTE: The eDirectory Management Toolbox only supports English, both in the command line client and command line help.

In this section:

- "Displaying the Command Line Help" on page 516
- "Running the Command Line Client in Interactive Mode" on page 516
- "Running the Command Line Client in Batch Mode" on page 520
- "eMBox Command Line Client Options" on page 522
- "Establishing a Secure Connection with the Client" on page 522
- "Finding Out eDirectory Port Numbers" on page 523

21.1.1 Displaying the Command Line Help

To display the general command line help before going in to the Client, do the following:

- Linux and UNIX: At the command line, enter `edirutil -?`.
- Windows: Run *drive*`\novell\nds\\edirutil.exe -?`

To display the interactive command line help while you are in the interactive mode, at the Client prompt enter a question mark (`?`). For example, `Client> ?`

The help displays information on the command line options like the information in "eMBox Command Line Client Options" on page 522.

21.1.2 Running the Command Line Client in Interactive Mode

Interactive mode lets you run commands one at a time.

In this section:

- "Running the Client on an eDirectory Server" on page 517
- "Running the Client on a Workstation" on page 517
- "Setting Up the Path and Classpath for Client" on page 517
- "Logging In to a Server" on page 518
- "Setting Preferred Languages, Timeout, and Log File" on page 518
- "Listing eMTools and Their Services" on page 519

- "Running a Particular Service" on page 519
- "Logging Out From the Current Server" on page 520
- "Exiting the Client" on page 520

Running the Client on an eDirectory Server

The Client and Sun JVM 1.3.1 are installed with eDirectory. To open the Client in interactive mode on an eDirectory server, do the following:

- Linux 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 Client. It points to the Java executable and the default location where the Client is installed with eDirectory. You can also enter the information manually, as described in "Setting Up the Path and Classpath for Client" on page 517.

You must have access behind the firewall to use the command line client for the servers you want to manage—so if you are remote, you'll need VPN access.

Running the Client on a Workstation

To use the Client on a machine other than an eDirectory server:

- Copy the `eMBoxClient.jar` file from an eDirectory server to your machine.
  - Windows: `\novell\nds\\eMBoxClient.jar`
  - Linux and UNIX: `/opt/novell/eDirectory/lib/nds-modules//eMBoxClient.jar`
- Make sure the machine has Sun JVM 1.3.1 installed.
- Make sure you have access behind the firewall to use the 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 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 Client" on page 517.

Setting Up the Path and Classpath for Client

If you are running the 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 Client. See "Running the Client on an eDirectory Server" on page 517.

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 Client as explained in this section.

You can run the 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, Linux, and UNIX servers, the directory needs to be in your path.

On a workstation, you might need to set it up yourself. For example, in 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. For example, in Windows enter `cd c:\novell\nds\\jre\bin`

- Add the path to the `eMBoxClient.jar` file to your classpath.

  Windows server or workstation: `set CLASSPATH=path\eMBoxClient.jar`

  Linux and 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:

  ```
  java -cp path/eMBoxClient.jar -i
  ```

  For example, in Windows enter `java -cp c:\novell\nds\\eMBoxClient.jar -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  -i
```

For information on Java commands, see the Java documentation on the Oracle Web site (http://www.oracle.com/technetwork/java/).

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 user name and password are not needed for public logins.

For example, after opening the 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 523.

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
`set -L en,de`	Sets the language preference to English and German (in that order).
`set -T 100`	Sets the timeout to 100 seconds. The timeout setting specifies how long to wait for responses from the server.
`set -l mylog.txt -o`	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
`list`	Lists the eMTools available on the server.
`list -r`	Refreshes the eMTool list.
`list -t backup`	Lists Backup services with details.
`list -t dsrepair`	Lists DSRepair services with details.
`list -t dsmerge -f`	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
`dsrepair.rld`	Repair local database.
`backup.getconfig`	Get backup configuration information.

For more information, see the following:

- Section 21.3, “Using the eMBox Client for Backup and Restore,” on page 525
- Section 10.4, “Using the Client to Merge Trees,” on page 241
- Section 12.10, “Using the Client to Repair a Database,” on page 289
- Section 6.4.1, “Using the Client Service Manager eMTool,” on page 189

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
```

21.1.3 Running the Command Line Client in Batch Mode

There are three ways you can run the Client in batch mode:

- "Single Tasks" on page 520
- "Internal Batch File" on page 520
- "System Batch File" on page 521

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 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 -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" on page 520.

Internal Batch File

To run the Client in batch mode using a Client internal batch file, you need to create a file which contains a group of commands you would run in the interactive mode.

A Client internal batch file lets you run all the commands in the batch file without your attention. You can perform multiple tasks with multiple 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 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 a Client command. For example, this command logs in to a server and runs the commands listed in the `mybatch.mbx` file:

```
java -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 521.

Here is an example of an 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 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 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 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 528.

From one server, you can perform tasks with multiple tools on multiple servers.

In a system batch file, you can use a combination of 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 520 above.

Consult the documentation for your operating system or third-party scheduling software for instructions on how to run batch files unattended.

21.1.4 eMBox Command Line Client Options

Option	Description
`-?` or `-h`	Display help information
`-i`	Interactively run commands one at a time.
`-s server`	Name or IP address of the server. Default=`127.0.0.1`
`-p port`	Port number of the server. Default=`8008`
`-u user`	User DN. For example, `admin.mycompany`. Default=`anonymous`
`-w password`	Password associated with the user specified with `-u`.
`-m mode`	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 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 Client" on page 522 for more information.
`-l log file`	Name of the log file.
`-o`	Overwrite the log file when opening it.
`-T timeout`	How long (in seconds) to wait for responses from the server.
`-L language`	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 [tool.]task options`	Perform a single service with this connection. The string following `-t` should be a valid command.
`-b batch file`	Perform a group of services as specified in the batch file. The commands in the batch file should be put on separate lines. Lines preceded by # are comments.

21.1.5 Establishing a Secure Connection with the 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`
- `jcert.jar`

If you don't, the Client will return an error saying that it cannot establish a secure connection.

You can get these files and information about JSSE from the Oracle Web site (http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136007.html).

21.1.6 Finding Out eDirectory Port Numbers

When logging in to a server in the Client, you must specify a port number.

If you specified a port number when you installed eDirectory, use that number.

For all platforms, the default nonsecure port is 8008, and the default secure port is 8030.

The following sections give some additional tips for finding out the port that is assigned to eDirectory:

- "On Windows" on page 523
- "On Linux and UNIX" on page 523

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 Linux and 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.

21.2 Using the Logger

The 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 Logger is different than the client logging service, which is provided through the log files that you specify when you run the client. For example, when you specify `-l mylogfile.txt` in a Client command or when you enter `mylogfile.txt` as a log file name in iManager. The Logger currently records all server messages for tasks that are performed by the eMTools, 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 Logger provides the following features:

- The ability to change the log file name and location.

  By default, log files are created in the `\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 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 Logger Command Line Client" on page 524
- "Using the Logger Feature in Novell iManager" on page 525

21.2.1 Using the Logger Command Line Client

The following table lists the Logger command line client options:

Option	Description
`logstart`	Starts the logger.
`logstop`	Stops the logger.
`readlog`	Displays the current log file.
`getlogstate`	Displays the current state of the logger (Start/Stop).
`getloginfo`	Displays the name, logging mode (Append/Overwrite), maximum size, and the current size of the log file.

`setloginfo [-f filename] [-s size in Kilo bytes] [-a | -o]`	Lets you set the name, size, and logging mode (Append/Overwrite) of the log file using the following parameters:
	`-f filename`
	The log file name.
	`-s size in KB`
	The maximum size of the log file.
	◆ `-a`
	New log messages will be appended to the current one.
	◆ `-o`
	The log file will be overwritten.

`emptylog`	Clears the contents of the server log file.

21.2.2 Using the Logger Feature in Novell iManager

1. In Novell iManager, click the *Roles and Tasks* button .
2. Click *eDirectory Maintenance* > *Log File*.
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.

21.3 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. It enables you to perform most backup and restore tasks tasks remotely 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.

In iManager, you can use all the features except cold backup, unattended backup, and advanced restore options, as explained in Section 21.4, "Using Novell iManager for Backup and Restore," on page 533.

The eDirectory Backup Tool is part of the eMBox tool set. The eMBox is a service that is installed on the server as part of eDirectory.

The Backup Tool 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 backuptl program.
`backuptl`	Tool interface to the backupcr library. Provides backup and restore functionality through the DSBK architecture. This can be accessed via the iManager plug-in or DSBK, the Java command line client.
`dsbackup_en.xlf`	Language file containing messages returned by the Backup Tool.

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.

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 515 and "Running the Client on a Workstation" on page 517.

Before performing backup and restore tasks, review Section 16.1, "Checklist for Backing Up eDirectory," on page 392 for an overview of the issues involved in planning an effective eDirectory backup strategy.

- Section 21.3.1, "Prerequisites," on page 526
- Section 21.3.2, "Backing Up Manually with the eMBox Client," on page 527
- Section 21.3.3, "Doing Unattended Backups, Using a Batch File with the eMBox Client," on page 528
- Section 21.3.4, "Configuring Roll-Forward Logs with the eMBox Client," on page 529
- Section 21.3.5, "Restoring from Backup Files with the eMBox Client," on page 531

21.3.1 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 installation. 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 Section 21.1, "Using the Command Line Client," on page 516.

- ❒ 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 Section 16.3, "Using Roll-Forward Logs," on page 404. For how to turn them on, see "Configuring Roll-Forward Logs with the eMBox Client" on page 529.

- ❒ Review the description of the command line options in "Backup and Restore Command Line Options" on page 419.
- ❒ For multiple-server trees, you should upgrade all the servers that share replicas with this server to eDirectory 8.5 or later.

21.3.2 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.

To back up the eDirectory database on a server using the eMBox Client:

1. Run the eMBox Client in interactive mode.
   - Linux 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.You can also enter the information manually, as described in "Setting Up the Path and Classpath for Client" on page 517.

   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, in 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 Client" on page 522.

   For help finding out which port number to use, see "Finding Out eDirectory Port Numbers" on page 523.

   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, in Windowsin 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.

For more information on manual backup, refer to Section 16.6.3, "Backing Up Manually with DSBK," on page 415.

21.3.3 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 Command Line Client in Batch Mode" on page 520.

This procedure describes using a system batch file:

1 Create a system batch file to back up the servers, following these general patterns, with one line per server.

In Windows and UNIX environments, 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
```

For examples and more explanation, see "Example of System Batch Files for Unattended Backups" on page 529.

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 523. If you want to use a secure connection, see "Establishing a Secure Connection with the Client" on page 522. For information on using an eMBox Client internal batch file as well, see "Running the Command Line Client in Batch Mode" on page 520.

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.

Example of System Batch Files for Unattended Backups

Below is an example system batch file:

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`).

21.3.4 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 Section 16.3, "Using Roll-Forward Logs," on page 404.

1. Run the eMBox Client in interactive mode:
   - Linux 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. it includes the necessary `-ns` option. You can also enter the options manually, as described in "Running the Client on a Workstation" on page 517.

   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, in 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 Client" on page 522.

   For help finding out which port number to use, see "Finding Out eDirectory Port Numbers" on page 523.

   The eMBox Client indicates whether the login is successful.
3. (Optional) Find out the current settings by entering the following command:

   ```
   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 C:\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.

   Ideally, you would have a separate disk partition/volume dedicated to roll-forward logs to make it easier to monitor disk space and rights.

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 407.

5 Log out from the server by entering the following command:

```
logout
```

6 Exit the eMBox Client by entering the following command:

```
exit
```

21.3.5 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 419, under restore and restadv.

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 Section 16.4, "Preparing for a Restore," on page 408.

2 Run the eMBox Client in interactive mode:

- Linux 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, it includes the necessary -ns option. You can also enter the information manually, as described in "Running the Client on a Workstation" on page 517.

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, in 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 Client" on page 522.

For help finding out which port number to use, see "Finding Out eDirectory Port Numbers" on page 523.

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 406.

For example:

```
restore -r -a -o -f sys:/backup/nds.bak -d $HOME/rfldir/nds.rfl -l $HOME/
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 Section 16.7, "Recovering the Database If Restore Verification Fails," on page 426.

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.

---

7 Log out from the server by entering the following command:

```
logout
```

8 Exit the eMBox Client by entering the following command:

```
exit
```

9 (Conditional) If you restored NICI security files, after completing the restore, restart the server to reinitialize NICI and then restore DIB.

10 Make sure the server is responding as usual.

11 (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 Section 16.3, "Using Roll-Forward Logs," on page 404.

Your restore should now be complete, and NICI reinitialized with the restored NICI files so you can access encrypted information. 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.

21.4 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 Tool, 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 2.7 Administration Guide* (http://www.novell.com/documentation/imanager27/imanager_admin_275/data/hk42s9ot.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 DSBK, as described in Section 16.6, "Using DSBK," on page 412.

Before performing backup and restore tasks, review Section 16.1, "Checklist for Backing Up eDirectory," on page 392 for an overview of the issues involved in planning an effective eDirectory backup strategy.

In this section:

- "Backing Up Manually with iManager" on page 533
- "Configuring Roll-Forward Logs with iManager" on page 536
- "Restoring from Backup Files with iManager" on page 537

21.4.1 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 DSBK. See "Backing Up Manually with DSBK" on page 415.

Before performing backup and restore tasks, review Section 16.1, "Checklist for Backing Up eDirectory," on page 392 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.

  You can back up NICI files and stream files by checking the check boxes for those options in iManager. We recommend that you always back up NICI files.

  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 Tool 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 Section 16.3, "Using Roll-Forward Logs," on page 404. For how to turn them on, see "Configuring Roll-Forward Logs with iManager" on page 536.

- ❒ For multiple-server trees, you should upgrade all the servers that share replicas with this server to eDirectory 8.5 or later.

Procedure

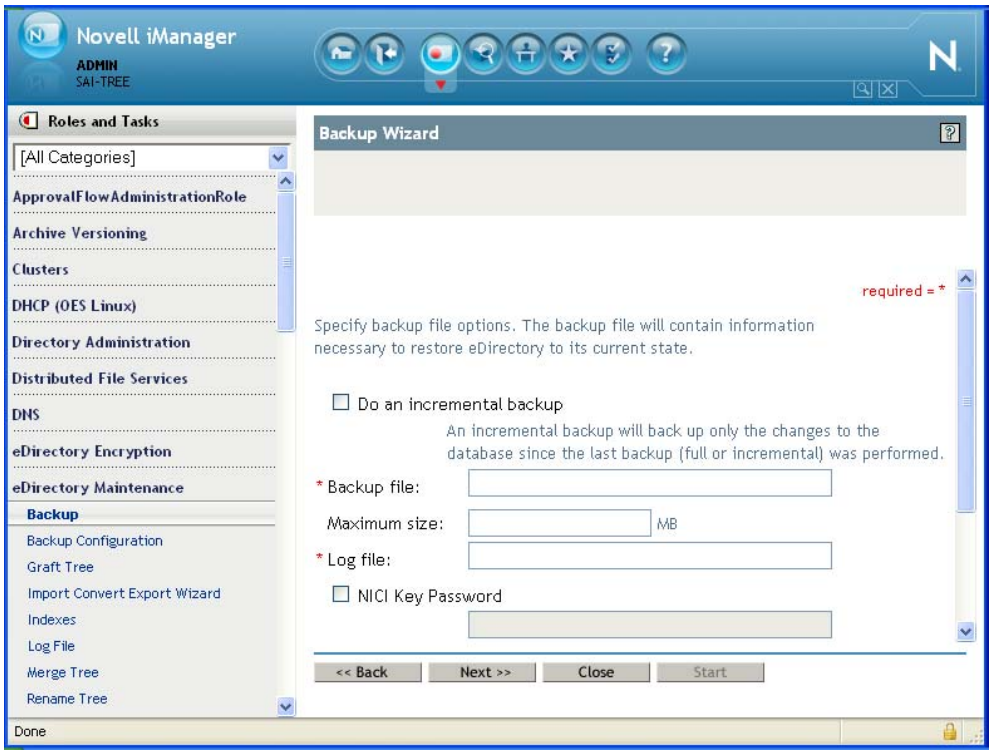
To back up the eDirectory database on a server, using iManager:

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 user name, 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.

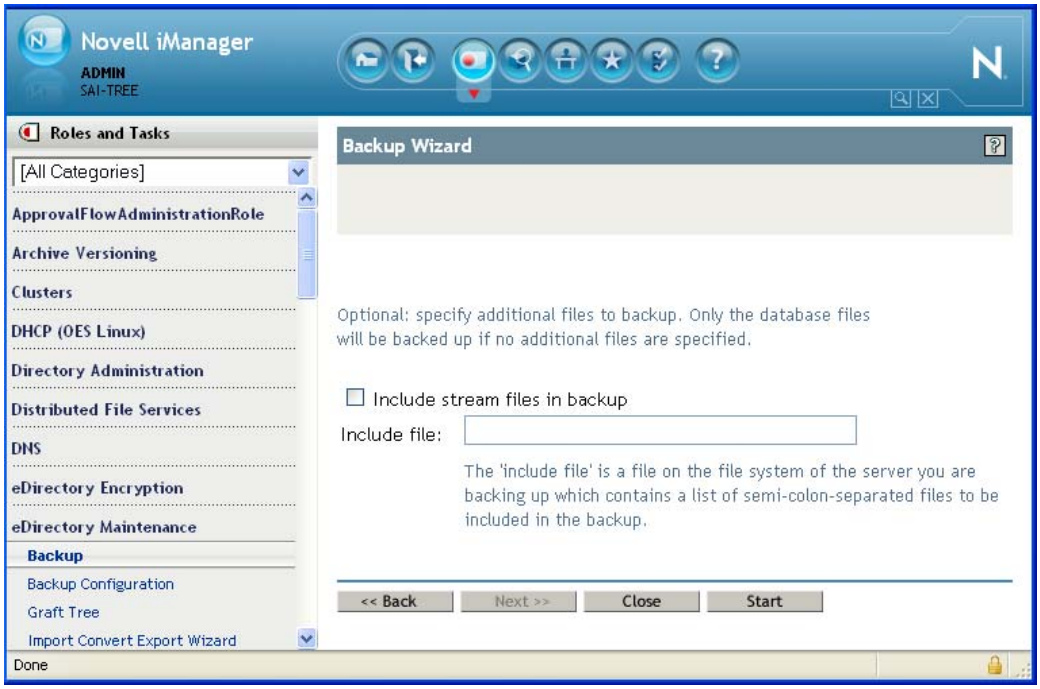


6 Specify additional files to back up.

If no additional files are specified, only the eDirectory database is backed up.

We recommend that you always back up NICI security files.

The following is an example of the screen.



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 Tool only places them on the server.

21.4.2 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 Section 16.3, "Using Roll-Forward Logs," on page 404.

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*.

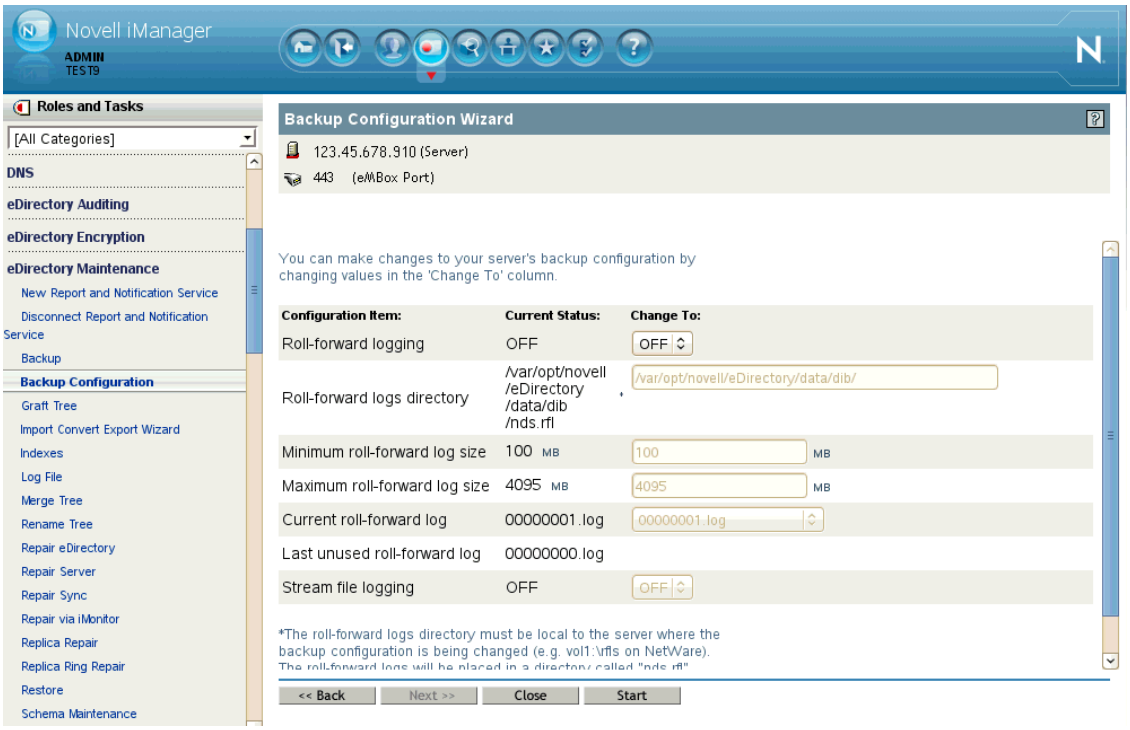
4 Specify a user name, password, and context for the server where you want to change configuration, then click *Next*.

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 407.

The following is an example of the screen.



6 Follow the online instructions to complete the operation.

21.4.3 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 Tool Does a Restore" on page 397.

Keep in mind that for advanced restore options you must use DSBK, as described in Section 16.6, "Using DSBK," on page 412.

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 Section 16.4, "Preparing for a Restore," on page 408 and "Locating the Right Backup Files for a Restore" on page 409.

- ❒ 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 Tool Does a Restore" on page 397.

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 Section 16.4, "Preparing for a Restore," on page 408.
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 user name, 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*.

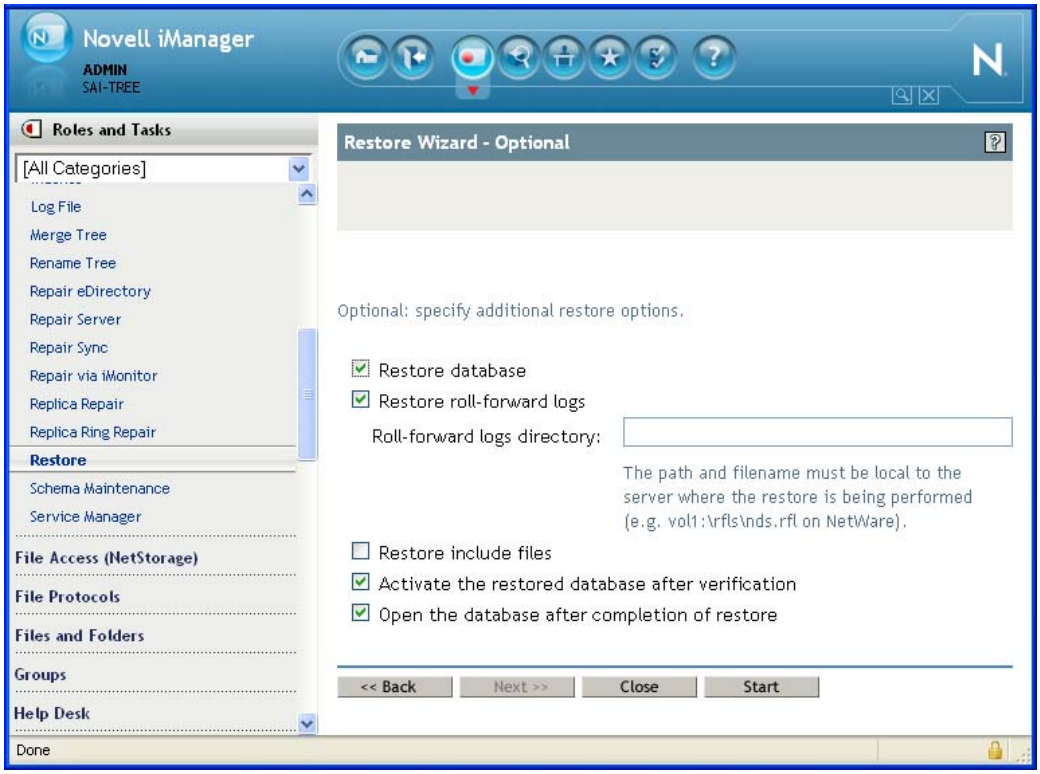
   In most cases you should at least check the check boxes for:

   - *Restore database*
   - *Activate the restored database after verification*
   - *Open the database after completion of restore*
   - *Restore security files* (meaning NICI files)

   We recommend that you always back up NICI files so you can read encrypted information after the 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 406.

The following is an example of the screen.



8 Follow the online instructions to complete the restore.

If the restore verification fails, see Section 16.7, “Recovering the Database If Restore Verification Fails,” on page 426.

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.

---

9 If you restored NICI security files, after completing the restore, restart the server to reinitialize NICI.

10 Make sure the server is responding as usual.

11 (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 Section 16.3, “Using Roll-Forward Logs,” on page 404.

Your restore should now be complete, and NICI reinitialized with the restored NICI files so you can access encrypted information. 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.

A NMAS Considerations

This appendix contains the following topics:

- Section A.1, “Setting Up a Security Container As a Separate Partition,” on page 541
- Section A.2, “Merging Trees with Multiple Security Containers,” on page 541

A.1 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 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.

A.2 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 2.2.1 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.

A.2.1 Product-Specific Operations to Perform prior to Tree Merge

This section contains the following information:

- "Novell Certificate Server" on page 542
- "Novell Single Sign-on" on page 543
- "NMAS" on page 544
- "Novell Security Domain Infrastructure" on page 544
- "Other Security-Specific Operations" on page 545

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 and 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.

   NOTE: To delete the Authorized Login Methods, use ldapdelete/ConsoleOne.

Novell Security Domain Infrastructure

If Novell Certificate Server 2.*x* or later, Novell Single Sign-on, NMAS, 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.

A.2.2 Performing the Tree Merge

eDirectory trees are merged using the DSMerge utility. For more information, see Chapter 10, "Merging Novell eDirectory Trees," on page 229 and Appendix B, "Novell eDirectory Linux and UNIX Commands and Usage," on page 547.

A.2.3 Product-Specific Operations to Perform after the Tree Merge

This section contains the following information:

- "Novell Security Domain Infrastructure" on page 545
- "Novell Certificate Server" on page 546
- "Novell Single Sign-On" on page 546
- "NMAS" on page 546

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 2.5.2 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 2.5.2 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 3.3 Administration Guide* (http://www.novell.com/documentation/crt33/crtadmin/data/a2ebomw.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 2.5.2 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 2.5.2 or later must be installed.

Novell Certificate Server 2.5.2 or later must be installed on the server that hosts the Organizational CA. For more information, see the *Novell Certificate Server 3.3 Administration Guide* (http://www.novell.com/documentation/crt33/crtadmin/data/a2ebomw.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 Services 3.3 Administration Guide* (http://www.novell.com/documentation/nmas33/admin/data/a20gkue.html).

B Novell eDirectory Linux and UNIX Commands and Usage

This chapter lists the utilities for Novell eDirectory 8.8 on Linux, Solaris, and AIX and their usage:

- Section B.1, “General Utilities,” on page 547
- Section B.2, “LDAP-Specific Commands,” on page 552

B.1 General Utilities

This section gives a list of the eDirectory utilities on Linux and UNIX and their usage.

NOTE: After installation, ensure that you run the ndsconfig, ndscheck, and ndslogin utilities from the installed location on the server, which is `/opt/novell/eDirectory/bin` by default. Do not run ndsconfig from the installation package.

For more information on the usage of utilities, see the utilities man pages.

Command	Description	Usage
`nds-install`	Utility that installs Novell eDirectory.	`nds-install [-h] [--help] [-i] [-j] [-u]`

Command	Description	Usage
ndsconfig	Configures Novell eDirectory	ndsconfig new [-m <*modulename*>] [-i] [-S <*server name*>] [-t <*tree_name*>] [-n <*server context*>] [-d <*path_for_dib*>][-P <*LDAP URL(s)*>] [-L <*ldap_port*>] [-l <*ssl_port*>] [-o <*http port*>] [-O <*https port*>] [-p <*IP address:[port]*>] [-c] [-w <*admin password*>] [-W <*obfuscated_password_file*>] [-e] [-a <*admin FDN*>] [-b <*port to bind*>] [-B <*interface1@port1*>, <*interface2@port2*>,..] [-D <*custom_location*>] [--config-file <*configuration file*>] ndsconfig def [-m <*modulename*>] [-i] [-S <*server name*>] [-t <*tree_name*>] [-n <*server context*>] [-d <*path_for_dib*>][-P <*LDAP URL(s)*>] [-L <*ldap_port*>] [-l <*ssl_port*>] [-o <*http port*>] [-O <*https port*>] [-e] [-a <*admin FDN*>] [-w <*admin password*>] [-W <*obfuscated_password_file*>] [-c] [-D <*custom_location*>] [--config-file <*configuration file*>] ndsconfig add [-m <*modulename*>] [-S <*server name*>] [-t <*tree_name*>] [-p <*IP_address:port*>] [-n <*server context*>] [-d <*path for dib*>][-P <*LDAP URL(s)*>] [-L <*ldap_port*>] [-l <*ssl_port*>] [-o <*http port*>] [-O <*https port*>] [-e] [-a <*admin FDN*>] [-w <*admin password*>] [-W <*obfuscated_password_file*>] [-p <*IP address:[port]*>] [-R] [-c] [-b <*port to bind*>] [-B <*interface1@port1, interface2@port2,..*>] [-D <*custom_location*>] [-E] [--config-file <*configuration file*>] ndsconfig rm [-a <*admin FDN*>] [-w <*admin password*>] [-W <*obfuscated_password_file*>] [-c] [-b <*port to bind*>] [--config-file <*configuration file*>] ndsconfig upgrade [-a <*admin FDN*>] [-w <*admin password*>] [-W <*obfuscated_password_file*>] [-c] [-j] [--config-file <*configuration file*>] ndsconfig {set <*valuelist*> | get [<*paramlist*>] | get help [<*paramlist*>]}

Command	Description	Usage
ndscheck	Utility that checks the health of the tree.	ndscheck [--help | -?] Display command usage ndscheck [--version | -v] Display version information ndscheck [-h <*hostname port*>] [-a <*admin FDN*>] [-F <*log file*>] [-D] [-q] [-w <*admin password*>] [-O <*obfuscated_password_file*>] [-W <*obfuscated_password_file*>] [--config-file <*file name*>] ndscheck [-a <*admin FDN*>] [-W] [--config-file <*file name*>] For example: ndscheck -a admin.novell -W --config-file /etc/opt/novell/eDirectory/conf-1/nds.conf
ndsmanage	Utility that lists the eDirectory instances.	ndsmanage [-a] ndsmanage [<*username*>]

Command	Description	Usage
ndsbackup	Creates eDirectory object archives and adds or extracts eDirectory objects	ndsbackup c [f <*ndsbackupfile*>] [e] [v] [w] [X<*exclude-file*>] [R] [Replica-server-name] [-a <*admin-user*>] [-I <*include-file*>] [-E <*password*>] [--config-file <*configuration_file_path*>]... [eDirectoryobject] ndsbackup r [f <*ndsbackupfile*>] [e] [v] [w] [X<*exclude-file*>] [R] [Replica-server-name] [-a <admin-user>] [-I <*include-file*>] [-E <*password*>] [--config-file <*configuration_file_path*>]... [eDirectoryobject] ndsbackup t [f <*ndsbackupfile*>] [e] [v] [w] [X<*exclude-file*>] [R] [Replica-server-name] [-a <admin-user>] [-I <*include-file*>] [-E <*password*>] [--config-file <configuration_file_path>]... [eDirectoryobject] ndsbackup x [f <*ndsbackupfile*>] [e] [v] [w] [X<*exclude-file*>] [R] [Replica-server-name] [-a <admin-user>] [-I <*include-file*>] [-E <*password*>] [--config-file <*configuration_file_path*>]... [eDirectoryobject] ndsbackup s [e] [v] [w] [X<*exclude-file*>] [R] [Replica-server-name] [-a <*admin-user*>] [-I <*include-file*>] [-E <*password*>] [--config-file <*configuration_file_path*>]... [eDirectoryobject] ndsbackup --version ndsbackup [option] [file] [-a <*admin FDN*>] [-p passstore] [--config-file <*file name*>] For example: ndsbackup cvf /tmp/test.bak -a admin.novell -p passstore --config-file /etc/opt/novell/eDirectory/conf-1/nds.conf
ndslogin	Diagnostic utility to verify Novell eDirectory authentication	ndslogin [-t <*treename*>] [-h <*hostname[:port]*>] [-p <*password*>] [-s] <*userFDN*> [--config-file <*configuration_file_path*>]
ndsd	NDS® daemon	/opt/novell/eDirectory/sbin/ndsd [--config-file *configfile*] **NOTE**: Before rebooting Solaris, ndsd needs to be stopped. Enter /etc/init.d/ndsd stop. For nonroot or custom location installation, use ndsmanage to stop the instance.

Command	Description	Usage
ndsimonitor	Monitors and diagnoses the servers in the Novell eDirectory tree using HTTP	/opt/novell/eDirectory/bin/ndsimonitor [-l [-d *<path of ndsimonitor conf files>*] | u] [-h *<local_interface:port>*] [--config-file *<configuration_file_path>*]
ndsmerge	Utility to merge two Novell eDirectory trees	ndsmerge [-m target-tree target-admin source-admin [target-container]] [-c] [-t] [-r target-tree source-admin] [-h *<local_interface:port>*] [--config-file *<configuration_file_path>*]
ndsrepair	Utility to repair and correct problems with the Novell eDirectory database, such as records, schema, bindery objects, and external references.	ndsrepair {-U |-E |-C |-P [Ad] |-S [Ad]|-N |-T |-J *<entry_id>*} [-A *<yes/no>*] [-O *<yes/no>*][-F *<filename>*] [-h *<local_interface:port>*] [--config-file *<configuration_file_path>*] 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>*] [-h *<local_interface>*] [--config-file *<configuration_file_path>*]
ndssch	Novell eDirectory schema extension utility	ndssch [-h *<hostname>*[:<port>]][-t *<treename>*][-F *<logfile>*] *<admin-FDN>* *<schemafile>* ... ndssch [-h *<hostname>*[:<port>]][-t *<treename>*] [-d] *<admin-FDN>* *<schemafile>* [schema description] ...
ndssnmp	SNMP services module for Novell eDirectory.	/opt/novell/eDirectory/bin/ndssnmp
ndssnmpconfig	SNMP trap configuration utility	ndssnmpconfig [-h *<hostname[:port]>*] [-p *<password>*] [-a *<userFDN>*] [-c *<command>*]
ndssnmpsa	eDirectory SNMP subagent daemon	/opt/novell/eDirectory/bin/ndssnmpsa
ndsstat	Utility that displays the server information	ndsstat { -r -s -p *<partitionname>*} [-n] [[-h *<hostname | IP address>:<port>*] | [--config-file *<configuration file>*]]
ndstrace	Utility that displays the server debug messages	ndstrace [-l|-u|-c "command1;......"|--version] [-h *<local_interface:port>*] [--config-file *<configuration_file_path>*]
nds-uninstall	Utility to uninstall Novell eDirectory	nds-uninstall [-s][-h]
nldap	LDAP services for NDS daemon	/opt/novell/eDirectory/sbin/nldap

Command	Description	Usage
nmasinst	NMAS configuration utility	nmasinst -i <*admin-FDN*> <*treename*> [-h <*hostname*>[:port]] nmasinst -addmethod <*admin-FDN*> <*treename*> <*config.txt file*> [-h <*hostname*>[:port]]
npki	Novell Public Key Infrastructure Services	/opt/novell/eDirectory/sbin/npki

B.2 LDAP-Specific Commands

Command	Description	Usage
ldapconfig	Utility to configure LDAP Server and LDAP Group objects	ldapconfig get [...] | set <*attribute-value-list*> [-t <*treename*> | -p <*hostname*>[:port] | --config-file <*configuration file*>] [-w <*password*>] [-a <*user FDN*>] [-f] ldapconfig [-t <*treename*> | -p <*hostname*>[:port]] [-w <*password*> | --config-file <*configuration file*>] [-a <*user FDN*>] [-V] [-R] [-H] [-f] -v <*attribute*>,<*attribute2*>... ldapconfig [-t <*treename*> | -p hostname[:port] | --config-file <*configuration file*>] [-w <*password*>] [-a <*admin FDN*>] [-V] [-R] [-H] [-f] -s <*attribute*>=<*value*>,...
ldapadd ldapmodify	Add or modify entries from an LDAP server	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*>]
ldapdelete	Delete entries from an LDAP server	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]...

Command	Description	Usage
ldapmodrdn	LDAP modify entry Relative Distinguished Name (RDN) tool.	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*>]
ldapsearch	The LDAP search tool	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....]
ndsindex	Utility to create, list, suspend, resume, or delete Novell eDirectory database indexes.	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*>.....]

Special Characters in User Name and Password

Using special characters in user names and passwords can create problems when the values are passed during an eDirectory installation or schema extension. If the user name or password contains special characters, such as $, # and so on, escape the character by preceding it with a backslash (\) .

For example, an administrator user name of `cn=admin$name.o=container` must be passed as `cn=admin\$name.o=container`.

When entering parameter values at the command line, you can escape the character, or place single quotes around the value.

For example,

```
cn=admin\$name.o=container
```

or

```
'cn=admin$name.o=container'
```

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.

- Section C.1, “Service Location Protocol,” on page 555
- Section C.2, “SLP Fundamentals,” on page 555
- Section C.3, “Configuration Parameters,” on page 558

C.1 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).

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).

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) Web site and the SourceForge (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.

C.2 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.

C.2.1 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.

C.2.2 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.

C.2.3 Service Agents

Service agents take the physical form of a separate process on the host machine. In the case of Windows, `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 557.).

C.3 Configuration Parameters

The SLP configuration parameters are stored in the `slp.conf` file, located in `/etc` on UNIX and Linux platforms and `%systemroot%/slp.conf` on Windows platforms. These parameters can be modified to tune the network operations.For example, the following parameters control the DA discovery:

```
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.

slptool Utility

This is a command line utility provided by OpenSLP. You can use slptool to register or de-register the services, query the scopes, service types, attributes, and the services available.

For example:

- To register the services,

  Syntax: `slptool register url [attrs]`

  ```
  slptool register service:myserv.x://myhost.com "(attr1=val1),(attr2=val2)"
  ```

- To de-register a service,

  Syntax: `slptool deregister url`

  ```
  slptool deregister service:myserv.x://myhost.com
  ```

- To find the available services,

  Syntax: `slptool findsrvs` *`service-type [filter]`*

  ```
  slptool findsrvs service:myserv.x
  ```

  ```
  slptool findsrvs service:myserv.x "(attr1=val1)"
  ```

- To find the configured scopes,

  Syntax: `slptool findscopes`

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 Linux and 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 server 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://en.wikipedia.org/wiki/Resource_record#DNS_resource_records).

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.

E Configuring GSSAPI with eDirectory

The SASL-GSSAPI mechanism for Novell eDirectory enables you to authenticate to eDirectory through LDAP using a Kerberos ticket. You are not required to enter the eDirectory user password. The Kerberos ticket should be obtained by authenticating to a Kerberos server.

For SASL-GSSAPI conceptual information, refer to the *Novell eDirectory 8.8 SP7 What's New Guide* (http://www.novell.com/documentation/edir88/edir88new/data/front.html).

NOTE: The SASL-GSSAPI mechanism is supported with eDirectory 8.8.3 or later on Linux platforms.

The following sections explain how to configure GSSAPI and describe the various tasks you can perform with Kerberos in eDirectory and give some useful additional information:

- Section E.1, "Prerequisites," on page 563
- Section E.2, "Configuring the SASL-GSSAPI Method," on page 567
- Section E.3, "Managing the SASL-GSSAPI Method," on page 568
- Section E.4, "Creating a Login Sequence," on page 575
- Section E.5, "How Does LDAP Use SASL-GSSAPI?," on page 575
- Section E.6, "Error Messages," on page 575

E.1 Prerequisites

To configure GSSAPI, you must first do the following:

- **SASL-GSSAPI method:** Install the SASL-GSSAPI method. Refer to the Installing a Login Method section in the *Novell Modular Authentication Services 3.3 Administration Guide* (http://www.novell.com/documentation/nmas33/admin/data/a20gkue.html).

  NOTE: The eDirectory SASL-GSSAPI method does not work on installations of Open Enterprise Server versions 2 or 11 that have Domain Services for Windows installed.

  To verify whether SASL-GSSAPI is installed on your machine, enter the following:

  ```
  ldapsearch -x -h osg-dt-srv9 -b " " -s base | grep -i sasl
  ```

  If SASL-GSSAPI is installed, the output of the command is similar to the following:

  ```
  supportedSASLMechanisms: NMAS_LOGIN

  supportedSASLMechanisms: GSSAPI
  ```

- **Kerberos plug-in for iManager:** Install the Kerberos plug-in for iManager. Refer to Section E.1.2, "Installing the Kerberos Plug-in for iManager," on page 564 for more information.

- **Key distribution center (KDC):** Install Kerberos KDC (MIT; Active Directory) on the network.

  For Microsoft KDC (Active Directory), you must have the Kerberos tools installed. These tools are part of the Windows installation and can be installed from `\support\tools\setup.exe` (Windows XP) and `\support\tools\suptools.msi` (Windows 2003) on the Windows installation CD.

- **Time Synchronization:** Synchronize the time on the NMAS client machine, the NMAS server machine, and the KDC machine for this method to work. For more information on synchronizing network time, refer to Section 2.8, "Synchronizing Network Time," on page 90.

- **Kerberos LDAP Extensions:** Add the Kerberos LDAP extensions. For more information, see Section E.1.3, "Adding Kerberos LDAP Extensions," on page 565.

IMPORTANT

- On Open Enterprise Server, do not add the Kerberos LDAP extensions on servers where Domain Services for Windows or DNS services are configured.
- All Kerberos information collected from your Kerberos administration is case-sensitive and must be specified exactly in the same case.

E.1.1 Assumptions on Network Characteristics

The SASL-GSSAPI mechanism is based on the following assumptions:

- All the machines in the network have loosely synchronized time. This means that no two machines in the network have their system time differing by more than five minutes.
- The SASL-GSSAPI mechanism is expected to be used mostly in LAN as it is difficult to obtain the time synchronization requirement mentioned above in MAN and/or WAN environments. However, this mechanism is not limited to LAN.
- You trust the Kerberos servers and Kerberos administrators unconditionally and unverifiably.
- Denial-of-Service attack is not countered. For more information, refer to RFC 1510 (http://www.ietf.org/rfc/rfc1510.txt?number=1510).

E.1.2 Installing the Kerberos Plug-in for iManager

1. Open the browser.
2. Enter the following URL in the address field of the browser window:

   `http://hostname/nps/`

   where *`hostname`* is the server name or IP address of the iManager server where you want to install the iManager plug-in for SASL-GSSAPI.

   NOTE: In case of problems, ensure that the Tomcat and Web server are configured properly. For information, refer to the *Novell iManager 2.7 Administration Guide* (http://www.novell.com/documentation/imanager27/imanager_admin_275/data/hk42s9ot.html).

   ---

3. Specify the user name and password to log in to eDirectory, then click *Login*.
4. Click *Configure* on the iManager toolbar.
5. In the left pane, click *Plug-in Installation* > *Available Novell Plug-in Modules*.

6 Click *Add*.

7 Specify the location of the `kerberosPlugin.npm` file or click *Browse* to select it.

The Kerberos Management plug-in is available as part of the eDirectory 88 single NPM (`eDir_88_iMan27_Plugins.npm`) and can be downloaded from Novell Download Site (http://download.novell.com/index.jsp).

If you have moved the `kerberosPlugin.npm` file to a different location, browse to the location and select it.

8 Click *Open*, then click *OK*.

9 Click *Install*.

This installation will take a few minutes.

10 Restart the iManager server after the Successfully saved module message appears.

If you are running iManager in an Unrestricted Access mode (no RBS collection in the tree), skip Step 11 through Step 17.

NOTE: For information on restarting the iManager server, refer to the *Novell iManager 2.7 Administration Guide* (http://www.novell.com/documentation/imanager27/imanager_admin_275/data/hk42s9ot.html).

---

11 Log in to iManager, then click the *Configure* button.

12 In the left pane, click *Role Based Services* > *RBS Configuration*.

13 (Conditional) If you do not have an RBS collection, do the following:

13a Click *New* > *Collection*.

13b Specify the name you want to use for the collection.

13c Select the container under which you want to create the Role Based services, then click *OK*.

13d Click *OK* again.

14 In the *iManager 2.x collections* tab, click the number in the *Modules* column for the collection you want to use.

15 Select *Kerberos Module* and click *Install*.

16 Click *OK* to continue.

17 When iManager finishes installing the module, click *OK*.

18 In the iManager toolbar, click *Roles and Tasks*.

The Kerberos Management role is displayed on the left pane.

If the Kerberos Management role is not displayed, restart the iManager server.

E.1.3 Adding Kerberos LDAP Extensions

Kerberos LDAP Extensions provide the functionality to manage Kerberos keys.

To use the Kerberos LDAP extensions, you must install the LDAP libraries for C. For more information, refer to LDAP Libraries for C (http://www.novell.com/developer/ndk/ldap_libraries_for_c.html).

To add or remove the Kerberos LDAP extensions, use the krbLdapConfig utility. When standalone eDirectory package is extracted to a directory, the path of this file is `extracted_folder/nmas/NmasMethods/Novell/GSSAPI/Kerberos_ldap_extensions/Linux/krbLdapConfig`.

For example, `/misc/eDir88/Linux/nmas/NmasMethods/Novell/GSSAPI/Kerberos_ldap_extensions/Linux/krbLdapConfig`.

To add the Kerberos LDAP extensions, use the following syntax:

```
krbldapconfig {-i | -u} -D bind_DN [-w bind_DN_password] [-h ldap_host] [-p ldap_port] [-e trusted_root_cert]
```

The following table explains the krbldapconfig utility parameters:

Parameter	Description
`-i`	Adds the Kerberos LDAP extensions to eDirectory.
`-u`	Removes the Kerberos LDAP extensions from eDirectory.
`-D bind_fdn`	Specifies the FDN of the administrator or the user with administrator-equivalent rights. This must be in the format `cn=admin,o=org`.
`-w bind_fdn_password`	Specifies the password of the bind FDN (`bind_fdn`).
`-h ldap_server`	Specifies the hostname or IP address of the LDAP server where Kerberos LDAP extensions must be installed.
`-p port`	Specifies the port where the LDAP server is running.
`-e trusted_root_file`	Specifies the trusted root certificate filename for the SSL bind. If you are using an SSL port, specify the `-e` option. For more information, refer to Section E.1.4, “Exporting the Trusted Root Certificate,” on page 567.

NOTE: If you do not specify the `-h` option, the name of the local host that krbldapconfig is invoked from is used as the default.

If you do not specify the LDAP server port and the trusted root certificate, the default port 389 is used.

If you do not specify the LDAP server port but specify the trusted root certificate, the default port 636 is used.

For example, enter the following to add the extensions:

```
krbldapconfig -i -D cn=admin,o=org -w password -h ldapserver -p 389
```

Or to remove, enter the following:

```
krbldapconfig -u -D cn=admin,o=org -w password -h ldapserver -p 389
```

IMPORTANT: You must manually refresh the LDAP server for the installation changes to take effect. For more information, refer to Section 15.5, “Refreshing the LDAP Server,” on page 362.

E.1.4 Exporting the Trusted Root Certificate

1. In iManager, click *Directory Administration* > *Modify Object* to open the Modify Object page.
2. Use the Object Selector to select the Server Certificate object of the server.
3. Click *OK*.
4. Click the *Certificates* tab, then select *Trusted Root Certificate* and view the details of the certificate.
5. Click *Export*.
6. Click the *Certificates* drop-down menu and select the certificate you want to export.
7. Specify whether you want to export the private key or not. If you want to export the private key, you might need to specify a password to protect the private key.
8. Click *Next*.
9. Click *Save the exported certificate.*
10. Click *Save File*.
11. Click *Close*.

E.2 Configuring the SASL-GSSAPI Method

1. The iManager plug-in for SASL-GSSAPI will not work if iManager is not configured to use SSL/TLS connection to eDirectory. A secure connection is mandated to protect the realm's master key and principal keys.

   By default, iManager is usually configured for SSL/TLS connection to eDirectory. You need to add the SSL trusted root certificates of the LDAP server that you use for Kerberos administration to iManager.

   For information on configuring iManager with SSL/TLS connection to eDirectory, refer to the *Novell iManager 2.7 Administration Guide* (http://www.novell.com/documentation/imanager27/imanager_admin_275/data/hk42s9ot.html).
2. Complete the following procedures in the order given:
   - **2a** Extend the Kerberos Schema.
   - **2b** Create a Realm Container.
   - **2c** Create the LDAP Service Principal.
   - **2d** Extract a Service Principal Key or Shared Key from KDC.
   - **2e** Creating a Service Principal Object in eDirectory.
   - **2f** Associate a Kerberos Principal Name with the User Object.

E.2.1 Merging eDirectory Trees Configured with SASL-GSSAPI Method

When you merge two trees, either or both configured with the SASL-GSSAPI method, you need to manually create all the Kerberos objects that are in the source tree in the target tree.

E.3 Managing the SASL-GSSAPI Method

You can perform the following Kerberos operations with iManager:

- Section E.3.1, "Extending the Kerberos Schema," on page 568
- Section E.3.2, "Managing the Kerberos Realm Object," on page 568
- Section E.3.3, "Managing a Service Principal," on page 570
- Section E.3.4, "Editing Foreign Principals," on page 574
- Section E.3.5, "Configuring SASL GSSAPI Authentication if MIT Kerberos KDC Uses eDirectory as Back End," on page 574

E.3.1 Extending the Kerberos Schema

This task allows you to extend your eDirectory schema with the Kerberos object class and attribute definitions.

1. If the schema has not already been extended, click *OK* to extend the schema.
2. In iManager, click *Kerberos Management* > *Extend Schema* to open the Extend Schema page.

   If the schema has been extended, a message is displayed with the status.
3. Click *Close*.

E.3.2 Managing the Kerberos Realm Object

A realm is the logical network served by a set of Key Distribution Centers (KDCs). In other words, a realm is a domain or grouping of principals served by a set of KDCs. By convention, realm names are generally all uppercase letters, to differentiate the realm from the internet domain. For more information, refer to RFC 1510 (http://www.ietf.org/rfc/rfc1510.txt?number=1510).

This section discusses the following:

- "Creating a New Realm Object" on page 568
- "Editing a Realm Object" on page 569
- "Deleting a Realm Object" on page 569

Creating a New Realm Object

The supported and the default encryption type is DES-CBC-CRC.

1. In iManager, click *Kerberos Management* > *New Realm* to open the New Realm page.
2. Specify a name for the Kerberos realm that is to be created.

   The realm name must be the same as the one that you want to configure this Login Method with and must conform to the RFC 1510 conventions.
3. Specify a master password for the realm, then confirm the password.

   NOTE: Ensure that you use a strong master password.
4. Specify the subtrees and Principal Container Reference you want the Kerberos realm to be configured with or use the *Object Selector* icon to select it.

This is the FDN of the subtree or the container that contains the eDirectory service principals of this realm. This subtree is not applicable to user principals.

5 Specify the scope of the subtree search:

- *One-level*: Searches the immediate subordinates of the realm subtree.
- *Subtree*: Searches the entire subtree starting with, and including, the realm subtree.

6 Click *OK*.

NOTE: The *KDC Services* box is not used in SASL-GSSAPI.

NOTE: If a Kerberos realm for LDAP SASL GSSAPI authentication has to be configured in the tree by an eDirectory container administrator, the tree administrator should perform the following:

1. Ensure that the security container object (cn=security) has objectclass krbContainerRefAux and the krbContainerReference attribute is set to the Kerberos container.
2. Grant Read access right to the container administrator over the krbContainerReference attribute.
3. Create a realm container under the Kerberos container. The name of the container should be same as the name of the new realm being created, and the objectclass should be krbRealmContainer.
4. Grant Supervisor right to the container administrator over the realm container.

Login to iManager as the container administrator, select *Kerberos Management* > *Set MasterKey* to open the Set Master Key page. Select the *MIT KDC realm* and specify a master password.

Editing a Realm Object

1 In iManager, click *Kerberos Management* > *Edit Realm* to open the Edit Realm page.

2 Specify a name for the Kerberos realm that is to be edited or use the *Object Selector* icon to select it.

3 Click *OK*.

4 Specify the subtree you want the Kerberos realm to be configured with or use the *Object Selector* icon to select it.

This is the FDN of the subtree or the container that contains the eDirectory service principals of this realm. This subtree is not applicable to user principals.

5 Specify the scope of the subtree search.

- *One-level*: Searches the immediate subordinates of the realm subtree.
- *Subtree*: Searches the entire subtree starting with, and including the realm subtree.

6 Click *OK*.

7 (Optional) To edit another realm, click *Repeat Task*.

NOTE: The *KDC Services* box is not used in SASL-GSSAPI.

Deleting a Realm Object

1 In iManager, click *Kerberos Management* > *Delete Realm* to open the Delete Realm page.

2 Select the realms that are to be deleted.

To select multiple realms, press Shift and select the realms or press Shift+Arrow keys.

3 Click *OK*.

4 Click *OK* again to confirm the delete operation or click *Cancel* to cancel the delete operation.

IMPORTANT: Deleting a Realm object deletes all service principal objects under that Realm.

E.3.3 Managing a Service Principal

This section discusses the following:

- "Creating a Service Principal for an LDAP Server" on page 570
- "Extracting the Key of the Service Principal for eDirectory" on page 571
- "Creating a Service Principal Object in eDirectory" on page 571
- "Viewing the Kerberos Service Principal Keys" on page 572
- "Deleting a Kerberos Service Principal Object" on page 572
- "Setting a Password for the Kerberos Service Principal" on page 573

Creating a Service Principal for an LDAP Server

Use the Kerberos Administration tool that is available with your KDC to create the eDirectory service principal with the encryption type and salt type as AES256-CTS and Normal, respectively.

The name of the principal must be ldap/*MYHOST.MYDNSDOMAIN@REALMNAME*.

For example, if you are using MIT KDC, execute the following command:

```
kadmin:addprinc -randkey -e aes256-cts:normal ldap/server.novell.com@MITREALM
```

For example, if you are using Heimdal KDC, execute the following command:

```
kadmin -l
```

```
kadmin> add --random-key ldap/server.novell.com@MITREALM
```

To delete the unsupported encryption types for the service principal, execute the following command:

```
kadmin> del_enctype ldap/MYHOST.MYDNSDOMAIN@MYREALM des-cbc-
```

```
md4
```

```
kadmin> del_enctype ldap/MYHOST.MYDNSDOMAIN@MYREALM des-cbc-
```

```
md5
```

```
kadmin> del_enctype ldap/MYHOST.MYDNSDOMAIN@MYREALM des3-cbc-sha1
```

where *MYHOST.MYDNSDOMAIN* is the host name and *MYREALM* is the Kerberos realm.

IMPORTANT: The hostname of service principal created must be in lowercase. Authentication fails if the hostname is in uppercase. For example, if the hostname is myHost.com, the hostname syntax of the LDAP service principal should look like `ldap/myhost.com@<realmname>`.

Best Practice

- All the keys should be preferably of type AES256.
- Change the LDAP service principal keys regularly. Whenever you change the LDAP service principal keys, ensure that you update the principal object in eDirectory.

Extracting the Key of the Service Principal for eDirectory

Use the Kerberos Administration tool that is available with your KDC to extract the key of the LDAP service principal created in "Creating a Service Principal for an LDAP Server" on page 570, then store it in the local file system. This can be done with the help of your Kerberos administrator.

For example, if you are using an MIT KDC, execute the following command:

```
kadmin: ktadd -k /directory_path/keytabfilename -e aes256-cts:normal ldap/
server.novell.com@MITREALM
```

For example, if you are using Microsoft KDC, create a user ldapMYHOST in Active Directory and then execute the following command:

```
ktpass -princ ldap/MYHOST.MYDNSDOMAIN@MYREALM -mapuser ldapMYHOST -pass mypassword
-out MYHOST.keytab
```

This command maps the principal (ldap/MYHOST.MYDNSDOMAIN@MYREALM) to the user account (ldapMYHOST), sets the host principal password to `mypassword`, and extracts the key into the `MYHOST.keytab` file.

For example, if you are using Heimdal KDC, execute the following command:

```
kadmin> ext_keytab -k /directory_path/keytabfilename ldap/
server.novell.com@MITREALM
```

where *`keytabfilename`* is the name of the file that contains the extracted key.

Creating a Service Principal Object in eDirectory

You must create a Kerberos service principal with the same name (ldap/*MYHOST.MYDNSDOMAIN@MYREALM*) as specified in "Creating a Service Principal for an LDAP Server" on page 570.

Best Practice

Service principals for eDirectory must be readily accessible to all servers enabled for the SASL GSSAPI mechanism. If these eDirectory service principals are not created under the Kerberos Realm container inside the Security container, we strongly recommend that you create the container that contains these eDirectory service principals as a separate partition, and that the container be widely replicated.

1. In iManager, click *Kerberos Management* > *New Principal* to open the New Principal page.
2. Specify the name of the principal to be created.

   The principal name must be in the format ldap/*MYDNSDOMAIN@REALMNAME*.
3. Specify the name of the container where the Principal object is to be created or use the *Object Selector* icon to select it.

4 Specify the name of the realm.

   If you have already specified the realm name in Step 2, leave this field blank.

5 Do either of the following:

   - Specify the keytab filename or click *Browse* to select the location where the keytab file is stored.

     This is the file that contains the key extracted in "Extracting the Key of the Service Principal for eDirectory" on page 571.

   - Specify the password, confirm the password, then select the encryption type and salt type combination.

     The password and encryption type/salt type combination must be the same as the those specified while creating the service principal in the KDC database.

6 Click *OK*.

Viewing the Kerberos Service Principal Keys

1 In iManager, click *Kerberos Management* > *View Key Information* to open the View Principal Keys page.

2 Specify the name of the principal key that is to be viewed or use the *Object Selector* icon to select it.

   The following information of the principal keys is displayed:

   - Principal name
   - Key Information
     - Number: Serial number of the key in the key table
     - Version: Version of the key
     - Key Type: Type of this principal key
     - Salt Type: Salt type of this principal key

3 Click *OK*.

Deleting a Kerberos Service Principal Object

You can delete a single object or multiple objects, or perform an advanced selection of the principal objects to be deleted.

To delete a single principal object:

1 In iManager, click *Kerberos Management* > *Delete Principal* to open the Delete Principal page.

2 Click *Select a Single Object*.

3 Specify the name of the Principal object to be deleted or use the *Object Selector* icon to select it.

4 Click *OK*.

5 Click *OK* again to confirm the delete operation or click *Cancel* to cancel the delete operation.

To delete multiple principal objects:

1 In iManager, click *Kerberos Management* > *Delete Principal* to open the Delete Principal page.

2 Click *Select Multiple Objects*.

3 Specify the name of the principal objects that are to be deleted or use the *Object Selector* icon to select them.

4 Select the principal to be deleted.

5 Click *OK*.

6 Click *OK* again to confirm the delete operation or click *Cancel* to cancel the delete operation.

To delete a principal using advanced selection:

1 In iManager, click *Kerberos Management* > *Delete Principal* to open the Delete Principal page.

2 Click *Advanced Selection*.

3 Select the object class.

4 Specify the container that contains the Principal object or use the *Object Selector* icon to select it.

5 Click *Include sub-containers* to include the subcontainers of the container specified in Step 3.

6 Click to open the Advanced Selection Criteria window.

7 Select the type of attribute and the operator from the drop-down list, then provide the corresponding values.

8 Click *Add Row* to include more Logic groups to the selection.

9 Click *OK* to set the filter.

10 Click *Show Preview* to display the preview of the advanced selection.

11 Click *OK*.

12 Click *OK* again to confirm the delete operation or click *Cancel* to cancel the delete operation.

Setting a Password for the Kerberos Service Principal

If the eDirectory service principal key has been reset in your KDC, you must update the key for this principal in eDirectory also.

For information on extracting the key, refer to "Extracting the Key of the Service Principal for eDirectory" on page 571.

1 In iManager, click *Kerberos Management* > *Set Principal Password* to open the Set Principal Password page.

2 Specify the name of the Principal object for which an individual password has to be set or use the *Object Selector* icon to select it.

3 Specify the keytab filename or click *Browse* to browse the location where the keytab file is stored.

4 Do either of the following:

- Specify the name of the keytab file that contains the principal key or click *Browse* to select the location where the keytab file is stored.

  For more information on creating service principals and extracting the keys, refer to "Creating a Service Principal for an LDAP Server" on page 570 and "Extracting the Key of the Service Principal for eDirectory" on page 571.

- Specify the password and confirm the password, then select the encryption type and salt type combination.

5 Click *OK* to set the password.

6 (Optional) To set the password for another principal, click *Repeat Task*.

E.3.4 Editing Foreign Principals

You can add Kerberos principal names to the eDirectory users using iManager.

1 In iManager, click *Kerberos Management* > *Edit Foreign Principals* to open the Edit Foreign Principals page.

2 Specify the FDN of a valid User object or use the *Object Selector* icon to select the User object reference.

3 Click *OK*.

4 Specify the foreign principal names, then click *Add* .

The principal name must be in the format `principalname@REALMNAME`.

To delete the foreign principal name, select the name and then click *Delete* .

5 Click *OK*.

NOTE: Kerberos principal names should be unique in the tree. If eDirectory is configured as a LDAP back end to a KDC realm, foreign principal names should not be configured in eDirectory for that realm. Instead, you can associate an existing Kerberos principal name with an eDirectory user DN using the following command:

```
kadmin.local -q 'modprinc -x linkdn=<eDir DN> <principal>@<realm>'
```

You can also associate a Kerberos principal name with an eDirectory user DN at the time of principal creation, using one of the following commands:

```
kadmin.local -q 'ank -x dn=<eDir DN> <principal>@<realm>'
```

```
kadmin.local -q 'ank -x linkdn=<eDir DN> <principal>@<realm>'
```

E.3.5 Configuring SASL GSSAPI Authentication if MIT Kerberos KDC Uses eDirectory as Back End

If MIT Kerberos KDC uses eDirectory as the back end, to enable the MIT KDC principals to authenticate to eDirectory using SASL GSSAPI, perform the following procedure after configuration of MIT KDC:

1 In iManager, edit the security container object (cn=security):

1a Add the objectclass krbContainerRefAux to the security container.

1b Set the attribute krbContainerReference to point to the Kerberos container.

For example:

```
cn=Kerberos,cn=Security
```

2 In iManager, select *Kerberos Management* > *Set MasterKey* to open the Set Master Key page.

3 Select the MIT KDC realm and specify the password. It should be the password that you used as the master password while creating the MIT KDC realm using kdb5_ldap_util.

NOTE: If the Kerberos realm is being created by a user who is not the tree administrator, the tree administrator should grant the Create entry rights to the user over the Kerberos container.

E.4 Creating a Login Sequence

For information on creating a login sequence, refer to the "Managing Login Sequences" section in the *Novell Modular Authentication Services 3.3 Administration Guide* (http://www.novell.com/documentation/nmas33/admin/data/a49tv39.html).

E.5 How Does LDAP Use SASL-GSSAPI?

Once you have configured SASL-GSSAPI, it is added along with the other SASL methods to the supportedSASLMechanisms attribute in rootDSE.

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.

Therefore, once you configure GSSAPI, it becomes the default mechanism.

However, to specifically do an LDAP operation over the SASL GSSAPI mechanism, you can mention GSSAPI at the command line.

For example, in OpenLDAP to do a search using the GSSAPI mechanism, enter the following:

```
ldapsearch -Y GSSAPI -h 164.99.146.48 -b "" -s base
```

E.6 Error Messages

The SASL-GSSAPI error messages are logged into the following locations:

- Linux: `ndsd.log`

For more information, refer to "Error Messages" in the *Novell eDirectory 8.8 SP7 Troubleshooting Guide* (http://www.novell.com/documentation/edir88/edir88tshoot/data/front.html).

F Security Considerations

This appendix contains the following topics:

- Section F.1, “LDAP Binds,” on page 577
- Section F.2, “Nessus Scan Results,” on page 577

F.1 LDAP Binds

The LDAP binds should take place over a secure connection. We recommend that you always use a SSL/TLS connection and keep in mind the following considerations:

- The key transmitted over the wire can be sniffed out. So you need to physically secure the corporate network against eaves-dropping or “packet sniffing”.
- You need to keep the servers in a physically secure location with access by authorized personnel only.
- When the product is used by users outside of the corporate firewall, a VPN should be employed.
- If a server is accessible from outside the corporate network, a firewall should be configured to prevent direct access to the server.
- Audit logs should be checked periodically.
- Different administrative duties should be given to separate people. Delegation of administration provides granular control over the directory objects.
- We recommend that you identify a particular LDAP server as the right server for Kerberos management. You can specify the server name in iManager.

IMPORTANT: The user needs to access the LDAP server using the DNS name instead of the IP address of the server. This is because the conversion of the IP address to the DNS name is not secure.

F.2 Nessus Scan Results

The following vulnerabilities were reported by Nessus port scan:

• LDAP servers that are not properly configured allow users to connect to the server and query for information

Explanation: Null Bind is enabled on eDirectory LDAP server by default but can be disabled on the server. To enhance the security of the server, disable the NULL bind on the LDAP server port 389. For more information, see Section 15.4, “Configuring LDAP Objects,” on page 356.

Solution: Disable Null Bind on the server.

- **LDAP servers that are not properly configured set the directory base as null**

  Explanation: Information can be picked even without prior knowledge of the directory structure. With the help of Null Bind, an anonymous user can query the LDAP server using tools like "LdapMiner."

  Solution: Although there is no way to disable it, security threat like this can be minimized by disabling Null Bind.

- **The remote service supports the use of weak SSL ciphers suites**

  Explanation: The remote host supports the use of SSL ciphers that offer either weak encryption or no encryption at all.

  Solution: Reconfigure the affected application, if possible, to avoid use of weak ciphers.

- **The remote directory server leaks information**

  Explanation: This host is a Novell eDirectory server, and has Browse rights on the PUBLIC object.

  Solution: If applications using eDirectory do not depend on having PUBLIC rights, then assign the rights given to PUBLIC to authenticated users (ROOT) only. If this is an external system, it is recommended to block the access to port 524 from the Internet.

- **SSL certificate is signed with an unknown certificate authority**

  Explanation: The X.509 certificate of the remote host is not signed by a known public certificate authority. If the remote host is a public host in production, this nullifies the use of SSL as anyone can establish a connection in the middle and attack against the remote host.

  Solution: This occurs when the client application does not have the certificate of the certificate authority that signed the server's certificate in its trusted certificate store. Purchase a certificate from a known certificate authority for the server and deploy it. Or, if the server's certificate has been issued either by the tree's organizational certificate authority or by an external or third-party certificate authority, then import or add the certificate authority's certificate in the applications trusted certificate store.

  For more information, see "Deciding Which Type of Certificate Authority to Use," in the *Novell Certificate Server 3.3 Administration Guide* (http://www.novell.com/documentation/crt33/crtadmin/data/fbgbacdg.html).