NetIQ® eDirectory™ 8.8 SP8

Troubleshooting Guide

September 2013

Legal Notice

THIS DOCUMENT AND THE SOFTWARE DESCRIBED IN THIS DOCUMENT ARE FURNISHED UNDER AND ARE SUBJECT TO THE TERMS OF A LICENSE AGREEMENT OR A NON-DISCLOSURE AGREEMENT. EXCEPT AS EXPRESSLY SET FORTH IN SUCH LICENSE AGREEMENT OR NON-DISCLOSURE AGREEMENT, NETIQ CORPORATION PROVIDES THIS DOCUMENT AND THE SOFTWARE DESCRIBED IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW DISCLAIMERS OF EXPRESS OR IMPLIED WARRANTIES IN CERTAIN TRANSACTIONS; THEREFORE, THIS STATEMENT MAY NOT APPLY TO YOU.

For purposes of clarity, any module, adapter or other similar material ("Module") is licensed under the terms and conditions of the End User License Agreement for the applicable version of the NetIQ product or software to which it relates or interoperates with, and by accessing, copying or using a Module you agree to be bound by such terms. If you do not agree to the terms of the End User License Agreement you are not authorized to use, access or copy a Module and you must destroy all copies of the Module and contact NetIQ for further instructions.

This document and the software described in this document may not be lent, sold, or given away without the prior written permission of NetIQ Corporation, except as otherwise permitted by law. Except as expressly set forth in such license agreement or non-disclosure agreement, no part of this document or the software described in this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, or otherwise, without the prior written consent of NetIQ Corporation. Some companies, names, and data in this document are used for illustration purposes and may not represent real companies, individuals, or data.

This document could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein. These changes may be incorporated in new editions of this document. NetIQ Corporation may make improvements in or changes to the software described in this document at any time.

U.S. Government Restricted Rights: If the software and documentation are being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), in accordance with 48 C.F.R. 227.7202-4 (for Department of Defense (DOD) acquisitions) and 48 C.F.R. 2.101 and 12.212 (for non-DOD acquisitions), the government's rights in the software and documentation, including its rights to use, modify, reproduce, release, perform, display or disclose the software or documentation, will be subject in all respects to the commercial license rights and restrictions provided in the license agreement.

© 2013 NetIQ Corporation and its affiliates. All Rights Reserved.

For information about NetIQ trademarks, see https://www.netiq.com/company/legal/.

Contents

About this Book and the Library 9
About NetIQ Corporation 11

1 Resolving Error Codes 13

2 Installation and Configuration 15

2.1 Installation . . . 15
2.1.1 Fatal Error Occurs in Schema Sync When Installing a Second eDirectory Server into the Tree on a SLES 11 Machine . . . 15
2.1.2 Installation Not Successful . . . 15
2.1.3 Installation Takes a Long Time . . . 16
2.1.4 eDirectory Install Fails for Container Administrators . . . 16
2.1.5 NICI Installation Failed - 1497 . . . 16
2.1.6 Naming Objects . . . 16
2.1.7 NICI Does Not Get Installed in the Server Mode on Windows . . . 17
2.1.8 Tarball Upgrade Fails With "Cannot open or remove a file containing a running program" Error Message . . . 17
2.1.9 Issue with eDirectory and YUM . . . 17
2.1.10 Performance Issues When Running eDirectory with BTRFS. . . . 17
2.2 Configuration . . . 18
2.2.1 Loopback Referrals Are Returned By a Directory Server . . . 18
2.2.2 Tree Name Lookup Failed: -632 Error While Configuring eDirectory 8.8 on Linux . . . 18
2.2.3 Adding New Servers . . . 18
2.2.4 Excluding the DIB directory from Backup or Antivirus Processes . . . 19
2.2.5 eDirectory ndsconfig Displays an Error on RHEL 32-Bit Platform . . . 19
2.2.6 IP AG Certificate Does Not Get Created on SLES 11 64-Bit Platform . . . 19
2.3 Upgrade . . . 19
2.3.1 Upgrade Fails if the Mount Point Is Set to /var/opt/novell/eDirectory/data . . . 19
2.3.2 Upgrading eDirectory After Applying a Patch Does Not Remove the Patch Version on a Windows System . . . 20
2.4 Multiple Instances . . . 20
2.4.1 If the First Instance is Down, HTTP Does Not Work . . . 21
2.4.2 eDirectory Does Not Listen on All the Configured Interfaces . . . 21
2.4.3 ndsd Falls Back to Default Port if the Interface Specified is Incorrect . . . 21
2.4.4 How to Rebuild .edir Directory . . . 21

3 Determining the eDirectory Version Number 23

3.1 Windows. . . . 23
3.2 Linux. . . . 23

4 Log Files 25

4.1 modschema.log . . . 25
4.2 dsinstall.log . . . 25
4.3 ndsd.log . . . 25

5 Troubleshooting LDIF Files 27

5.1 Understanding LDIF . . . 27

5.1.1 LDIF File Format . . . 27
5.1.2 LDIF Content Records . . . 28
5.1.3 LDIF Change Records . . . 29
5.1.4 Line Folding within LDIF Files . . . 33
5.1.5 Hashed Password Representation in LDIF Files . . . 34
5.2 Debugging LDIF Files . . . 34
5.2.1 Enabling Forward References . . . 35
5.2.2 Checking the Syntax of LDIF Files . . . 37
5.2.3 Resolving the LDIF Error File . . . 38
5.2.4 Using LDAP SDK Debugging Flags . . . 38
5.3 Using LDIF to Extend the Schema . . . 39
5.3.1 Adding a New Object Class . . . 39
5.3.2 Adding a New Attribute . . . 40
5.3.3 Adding or Removing Auxiliary Classes . . . 41
5.4 ldif2dib Limitations . . . 43
5.4.1 Simple Password LDIF . . . 43
5.4.2 Schema . . . 43
5.4.3 ACL Templates . . . 44
5.4.4 Signal Handler . . . 44

6 Troubleshooting SNMP 45

6.1 Traps Might Not Get Generated As Expected . . . 45
6.2 SNMP Group Object . . . 45
6.3 SNMP Initializing Errors . . . 46
6.4 SNMP Subagent Does Not Start . . . 46
6.5 LDAP SNMP Statistics Not Reported . . . 46
6.6 Segmentation Fault Error while Accessing the Subagent . . . 46
6.7 SNMP Issues . . . 46
6.7.1 Issues After Upgrading from eDirectory 8.7.3 to eDirectory 8.8 . . . 47
6.7.2 Errors While Starting the NDS Subagent . . . 47
6.7.3 Restarting ndssnmpsa . . . 47
6.7.4 Errors While Starting ndssnmpsa . . . 48
6.7.5 Errors While Stopping ndssnmpsa . . . 48
6.7.6 Compiling edir.mib . . . 48
6.7.7 Modifying the SNMP Configuration File . . . 48
6.7.8 Using SNMP After a New Tree Installation . . . 48
6.7.9 SNMP Object Creation Error on Windows Server . . . 49
6.7.10 Uninstalling SNMP with eDirectory Uninstallation . . . 49

7 iMonitor 51

7.1 Browsing for Objects Containing Double-Byte Characters in iMonitor . . . 51
7.2 Agent Health Check on a Single-Server Tree . . . 51
7.3 iMonitor Report Does Not Save the Records for Each Hour . . . 52
7.4 Creation and Modification Time Stamps . . . 52
7.5 iMonitor Issues in Older Versions of Mozilla . . . 52
7.6 Run Report Screen Layout Not Aligned on iMonitor . . . 52
7.7 iMonitor Displays Error -672 . . . 52
7.8 Time Stamps Displayed in Hexadecimal Format . . . 52
7.9 Issue with iMonitor Trace Configuration in Internet Explorer 10 . . . 53

8 iManager 55

8.1 LDAP Operations Fail After Creating a New LDAP Group Using Quick Create . . . 55
8.2 Issues While Backing Up on Red Hat EUC in the Japanese Locale . . . 55

9 Obituaries **57**

9.1 Examples . . . 58
9.1.1 Deleting an Object . . . 58
9.1.2 Moving an Object . . . 59
9.2 Prevention . . . 59
9.3 Troubleshooting Tips . . . 60
9.3.1 Solutions . . . 61
9.3.2 Previous Practices . . . 62

10 Migrating to NetIQ eDirectory **63**

10.1 Migrating the Sun ONE Schema to NetIQ eDirectory . . . 63
10.1.1 Step 1: Perform the Schema Cache Update Operation . . . 63
10.1.2 Step 2: Rectify the Error LDIF File to Eliminate the Errors . . . 63
10.1.3 Step 3: Import the LDIF File . . . 65
10.2 Migrating the Active Directory Schema to NetIQ eDirectory Using ICE . . . 66
10.2.1 Step 1: Perform the Schema Cache Update Operation . . . 66
10.2.2 Step 2: Rectify the Error LDIF File to Eliminate the Errors . . . 66
10.2.3 Step 3: Import the LDIF File . . . 67
10.3 Migrating from OpenLDAP to NetIQ eDirectory . . . 67
10.3.1 Prerequisites . . . 67
10.3.2 Migrating the OpenLDAP Schema to eDirectory . . . 67
10.3.3 Migrating the Open LDAP Data to NetIQ eDirectory . . . 68
10.3.4 Making PAM Work with NetIQ eDirectory After Migration . . . 68

11 Schema **71**

12 DSRepair **73**

12.1 Running DSRepair on an NFS Mounted DIB on Linux . . . 73
12.2 Running DSRepair with -R Option Hangs . . . 73
12.3 Running DSRepair after Upgrade or Migration . . . 73

13 Replication **75**

13.1 Encrypted Replication Issues . . . 75
13.1.1 Configuring Encrypted Replication Through iManager . . . 75
13.1.2 Merging Trees With Encrypted Replication Enabled Fails . . . 75
13.2 Recovering from eDirectory Replica Problems . . . 75

14 Clone DIB Issues **77**

14.1 Clone DIB Fails With -601 and -603 Errors . . . 77
14.2 Clone DIB Can Fail Immediately After Offline Bulkload . . . 77
14.3 Issue in Cloning with Enabled Encrypted Replication Feature . . . 77

15 NetIQ Public Key Infrastructure Services **79**

15.1 PKI Operations Not Working . . . 79
15.2 LDAP Search from Netscape Address Book Fails . . . 79
15.3 Removing the configuration of an eDirectory server that is acting as a treekey server in a multiserver tree after having moved the existing eDirectory objects to a different server fails with the error code for Crucial Replica . . . 80

15.4 While uninstalling the eDirectory Server holding the CA, the KMOs created on that server will be moved to another server in the tree and become invalid . . . 80
15.5 PKI Plugin Encounters Error When Installed in iManager 2.7.6 Patch1 and Lower Versions . . . 80

16 Troubleshooting Utilities on Linux 81

16.1 NetIQ Import Convert Export Utility . . . 81
16.2 ndsconfig Utility . . . 81
16.2.1 Configuring ndsconfig to Run From Non-Default Location . . . 81
16.2.2 ndsconfig Does Not Verify an Invalid Configuration File Path . . . 81
16.2.3 ndsconfig get Outputs Junk Characters for Non-English Characters . . . 82
16.3 ndsmerge Utility . . . 82
16.4 DSTrace Utility . . . 82
16.5 ndsbackup Utility . . . 82
16.6 Using DSRepair . . . 82
16.6.1 Syntax . . . 83
16.6.2 Troubleshooting DSRepair . . . 89
16.7 Using DSTrace . . . 89
16.7.1 Basic Functions . . . 90
16.7.2 Debugging Messages . . . 90
16.7.3 Background Processes . . . 93

17 NMAS on Linux 99

17.1 Unable to Log In Using Any Method . . . 99
17.2 The User Added Using the ICE Utility Is Unable to Log In Using Simple Password . . . 99

18 Troubleshooting on Windows 101

18.1 The eDirectory for Windows Server Won't Start. . . . 101
18.2 The Windows Server Can't Open the eDirectory Database Files . . . 101
18.3 SLP_NETWORK_ERROR(-23) Occurs in Windows Machines . . . 102
18.4 Incorrect Installation Path Appears in the Browse Page During eDirectory Installation . . . 103

19 Accessing HTTPSTK When DS Is Not Loaded 105

19.1 Setting the sadmin Password on Windows . . . 105
19.2 Setting the sadmin Password on Linux . . . 105

20 Encrypting Data in eDirectory 107

20.1 Error Messages . . . 107
20.1.1 -6090 0xFFFFE836 ERR_ER_DISABLED . . . 107
20.1.2 -6089 0xFFFFE837 ERR_REQUIRE_SECURE_ACCESS . . . 107
20.1.3 -666 FFFFFD66 INCOMPATIBLE NDS VERSION . . . 108
20.2 Problem With Duplicate Encryption Algorithms . . . 109
20.3 Encryption of Stream Attributes . . . 109
20.4 Configuring Encrypted Replication through iManager . . . 109
20.5 Viewing or Modifying Encrypted Attributes through iManager . . . 110
20.6 Merging Trees With Encrypted Replication Enabled Fails . . . 110
20.7 Limber Displays -603 Error . . . 110

21 The eDirectory Management Toolbox 111

21.1 Unable to Stop the eMTool Services . . . 111

21.2 Restore gives -6020 error . . . 111
21.3 eDirectory Service Manager Issues . . . 112
21.3.1 Deletion of a Moved Object . . . 112
21.3.2 Issue with Moving a Dynamic Group . . . 112
21.3.3 Issue with Repairing Network Addresses through eMBox . . . 112
21.3.4 Viewing French Man Pages . . . 112
21.3.5 Deleting a Moved Object . . . 112
21.3.6 eDirectory Does Not Generate a Logout Event due to eDirectory Client Limitation . . . 113
21.3.7 Issues Generated by TERM While Running DSTrace . . . 113
21.3.8 eMBox Does Not Handle Double-Byte Characters . . . 113

22 SASL-GSSAPI 115

22.1 SASL-GSSAPI Issues . . . 115
22.1.1 Issue with Multiple User Objects . . . 115
22.1.2 Authorization ID . . . 115
22.2 Log File . . . 115
22.3 Error Messages . . . 115

23 Miscellaneous 119

23.1 Backing Up a Container . . . 120
23.2 Repeated eDirectory Logins . . . 120
23.3 Enabling Event System Statistics . . . 120
23.4 Tracking Memory Corruption Issues on Linux . . . 120
23.5 TCP Connection not Terminating after Abnormal Logout . . . 120
23.6 NDS Error, System Failure (-632) Occurs When Doing ldapsearch for the User Objects . . . 122
23.7 Disabling SecretStore . . . 122
23.7.1 On Linux . . . 122
23.7.2 On Windows . . . 122
23.8 Viewing SLP Man Pages . . . 122
23.9 dsbk Configuration File Location . . . 123
23.10 SLP Interoperability Issues on OES Linux . . . 123
23.11 ldif2dib Fails to Open the Error Log File When the DIB Directory Exists In the Custom Path . . . 123
23.12 eDirectory Server Does Not Start Automatically in the Virtual SLES 10 . . . 123
23.13 ndsd Does Not Start After a System Crash . . . 123
23.14 Do not Execute DSTrace With All Tags Enabled on Linux Computers . . . 124
23.15 LDAP is Not RFC Compliant For Anonymous Search Requests . . . 124
23.16 Troubleshooting Ports with Custom eDirectory 8.8 Instances . . . 124
23.17 Rebooting the Host . . . 124
23.18 ndsd Not Listening at the Loopback Address on a Given NCP Port . . . 124
23.19 LDAP Transaction OIDs . . . 124
23.20 Errors -5871 and -5875 in LDAP Trace . . . 125
23.21 NDSCons Gives -625 Error if a Tree is Renamed . . . 125
23.22 Listening on Multiple NICs Slows Down eDirectory ldapsearch Performance . . . 125
23.23 Unable to Limit the Number of Concurrent Users on Linux Platforms . . . 125
23.24 ndsd Fails to Shut Down Due to SLP . . . 125
23.25 Restarting NLDAP on Windows . . . 126
23.26 SecretStore over LDAP . . . 126
23.27 Interoperability Issues . . . 126
23.27.1 Cannot Change the Passphrase after Unlocking SecretStore . . . 126
23.27.2 User Credentials Modified through SecretStore Are Reset to Null . . . 126
23.27.3 Creating a Different Credential Set with the Same User Overwrites the Previous Credential Set . . . 126

24 IPV6 **127**

24.1 The LDAP Secure Search Works Either With IPv4 or IPv6 But Not With Both 127
24.2 ICE Plug-In Does Not Work for the IPV6 Addresses 127
24.3 Listeners for Unspecified IPv6 Addresses in Linux and Windows 128

About this Book and the Library

The *Troubleshooting Guide* describes how to resolve issues with the NetIQ eDirectory (eDirectory) product.

For the most recent version of the *NetIQ eDirectory 8.8 SP8 Troubleshooting Guide*, see the NetIQ eDirectory 8.8 online documentation Web site.

Intended Audience

The guide is intended for network administrators.

Other Information in the Library

The library provides the following information resources:

XDASv2 Administration Guide

Describes how to configure and use XDASv2 to audit eDirectory and NetIQ Identity Manager.

Installation Guide

Describes how to install eDirectory. It is intended for network administrators.

Administration Guide

Describes how to manage and configure eDirectory.

What's New Guide

Describes the new features of eDirectory.

Tuning Guide for Linux Platforms

Describes how to analyze and tune eDirectory on Linux platforms to yield superior performance in all deployments.

These guides are available at NetIQ eDirectory 8.8 documentation Web site.

For information about the eDirectory management utility, see the *NetIQ iManager 2.7 Administration Guide*.

About NetIQ Corporation

We are a global, enterprise software company, with a focus on the three persistent challenges in your environment: Change, complexity and risk—and how we can help you control them.

Our Viewpoint

Adapting to change and managing complexity and risk are nothing new

In fact, of all the challenges you face, these are perhaps the most prominent variables that deny you the control you need to securely measure, monitor, and manage your physical, virtual, and cloud computing environments.

Enabling critical business services, better and faster

We believe that providing as much control as possible to IT organizations is the only way to enable timelier and cost effective delivery of services. Persistent pressures like change and complexity will only continue to increase as organizations continue to change and the technologies needed to manage them become inherently more complex.

Our Philosophy

Selling intelligent solutions, not just software

In order to provide reliable control, we first make sure we understand the real-world scenarios in which IT organizations like yours operate — day in and day out. That's the only way we can develop practical, intelligent IT solutions that successfully yield proven, measurable results. And that's so much more rewarding than simply selling software.

Driving your success is our passion

We place your success at the heart of how we do business. From product inception to deployment, we understand that you need IT solutions that work well and integrate seamlessly with your existing investments; you need ongoing support and training post-deployment; and you need someone that is truly easy to work with — for a change. Ultimately, when you succeed, we all succeed.

Our Solutions

- Identity & Access Governance
- Access Management
- Security Management
- Systems & Application Management
- Workload Management
- Service Management

Contacting Sales Support

For questions about products, pricing, and capabilities, contact your local partner. If you cannot contact your partner, contact our Sales Support team.

Worldwide:	www.netiq.com/about_netiq/officelocations.asp
United States and Canada:	1-888-323-6768
Email:	info@netiq.com
Web Site:	www.netiq.com

Contacting Technical Support

For specific product issues, contact our Technical Support team.

Worldwide:	www.netiq.com/support/contactinfo.asp
North and South America:	1-713-418-5555
Europe, Middle East, and Africa:	+353 (0) 91-782 677
Email:	support@netiq.com
Web Site:	www.netiq.com/support

Contacting Documentation Support

Our goal is to provide documentation that meets your needs. If you have suggestions for improvements, click **Add Comment** at the bottom of any page in the HTML versions of the documentation posted at www.netiq.com/documentation. You can also email Documentation-Feedback@netiq.com. We value your input and look forward to hearing from you.

Contacting the Online User Community

Qmunity, the NetIQ online community, is a collaborative network connecting you to your peers and NetIQ experts. By providing more immediate information, useful links to helpful resources, and access to NetIQ experts, Qmunity helps ensure you are mastering the knowledge you need to realize the full potential of IT investments upon which you rely. For more information, visit http://community.netiq.com.

1 Resolving Error Codes

For a complete list and explanation of eDirectory error codes, see the NetIQ Error Codes Web page (http://www.novell.com/documentation/nwec/).

2 Installation and Configuration

- Section 2.1, “Installation,” on page 15
- Section 2.2, “Configuration,” on page 18
- Section 2.3, “Upgrade,” on page 19
- Section 2.4, “Multiple Instances,” on page 20

2.1 Installation

This section discusses various problems you may encounter during the eDirectory 8.8 installation along with troubleshooting tips.

- Section 2.1.1, “Fatal Error Occurs in Schema Sync When Installing a Second eDirectory Server into the Tree on a SLES 11 Machine,” on page 15
- Section 2.1.2, “Installation Not Successful,” on page 15
- Section 2.1.3, “Installation Takes a Long Time,” on page 16
- Section 2.1.4, “eDirectory Install Fails for Container Administrators,” on page 16
- Section 2.1.5, “NICI Installation Failed - 1497,” on page 16
- Section 2.1.6, “Naming Objects,” on page 16
- Section 2.1.7, “NICI Does Not Get Installed in the Server Mode on Windows,” on page 17
- Section 2.1.8, “Tarball Upgrade Fails With "Cannot open or remove a file containing a running program" Error Message,” on page 17
- Section 2.1.9, “Issue with eDirectory and YUM,” on page 17
- Section 2.1.10, “Performance Issues When Running eDirectory with BTRFS,” on page 17

2.1.1 Fatal Error Occurs in Schema Sync When Installing a Second eDirectory Server into the Tree on a SLES 11 Machine

Configure an eDirectory tree and Install another server into the tree. In both the cases, select the option to use all the available interfaces. Use the same interfaces for both the servers. For example, 127.0.0.2. Start DSTrace on the first server with SCMA, SKLK and SYNC options.

2.1.2 Installation Not Successful

- Check for the following error message in the `/var/adm/messages` directory:

  ```
  Unable to bind to SLP Multicast Address. Multicast route not added?
  ```

  This message is displayed if the Linux or Solaris machine is not configured for a multicast route address.

Add the multicast route address and restart the slpuasa daemon.

- If the `-632: Error description System failure` error message appears during installation, exit from the installation process.

  Set the `n4u.base.slp.max-wait` parameter to a larger value, such as 50, in the `/etc/opt/novell/eDirectory/conf/nds.conf` file, then restart the installation process.

- During installation, if the `Tree Name Not Found` error message is displayed, do the following:

1. Check whether multicast routing is enabled on the Solaris host that you are installing the product on.
2. Specify the IP address of the master server of the Tree partition.

2.1.3 Installation Takes a Long Time

When you are installing eDirectory into an existing tree and the installation takes a long time to complete, look at the DSTrace screen on the server. If the `-625 Transport failure` message appears, you need to reset the address cache.

To reset the address cache, enter the following command at the system console:

```
set dstrace = *A
```

2.1.4 eDirectory Install Fails for Container Administrators

The eDirectory 8.8 installation program supports installations by administrators who have supervisor rights to the container that the server resides in. In order to handle this, the first server that eDirectory 8.8 is installed into must have supervisor rights to [Root] to extend the schema. From that point on, subsequent servers do not have to have rights to [Root]. However, with eDirectory 8.8, depending on the platform that eDirectory 8.8 is installed in to first, all schema might not be extended, requiring supervisor rights to [Root] for subsequent server installations on different platforms.

If eDirectory 8.8 will be installed on multiple platforms, make sure that you have supervisor rights to [Root] for the first server eDirectory will be installed on for EACH platform. For example, if the first server that eDirectory 8.8 is going to be installed on is running Linux, and eDirectory 8.8 will also be installed on Solaris, the first server for each platform must have supervisor rights to [Root]. Subsequent servers on each platform will only have to have container administrator rights to the container where the server is being installed.

For additional information, see solution NOVL83874 (http://support.novell.com/docs/Tids/Solutions/10073723.html) in the *eDirectory 8.7.x Readme Addendum*.

2.1.5 NICI Installation Failed - 1497

A message warning that the NetIQ International Cryptographic Infrastructure (NICI) initialization failed means that the `NFK` file is not right. Ensure that you have the right `NFK` file. This problem might not occur on Linux platforms, as by default the `NFK` file is part of the NICI package.

2.1.6 Naming Objects

When you use special characters while naming objects, the `-671 No Such Parent` error message appears. Avoid using any of the following special characters when naming objects:

\ /, * ? .

2.1.7 NICI Does Not Get Installed in the Server Mode on Windows

In the Properties dialog box of the `NICIFK` file there is a tab called Security. If there are no names in the Group or user names field, then this issue occurs.

To work around this problem, do the following:

1. Remove the `NICIFK` file.

   This is present in `C:/Windows/system32/novell/nici` if the system root is `C:/Windows/system32`. If the system root is `F:/Windows/system32` then this file is present in `F:/Windows/system32/novell/nici`.
2. Install eDirectory.

2.1.8 Tarball Upgrade Fails With "Cannot open or remove a file containing a running program" Error Message

While doing Tarball Upgrade in AIX, in the file copying stage if you get `Cannot open or remove a file containing a running program` error message, perform the following steps to resolve the issue:

1. Run `/usr/sbin/slibclean` as a root user.
2. Continue the upgrade from file copying stage.

2.1.9 Issue with eDirectory and YUM

If you install eDirectory 8.8 SP6 or later on a Red Hat Enterprise Linux server with the YUM package manager installed, you may encounter an issue when using YUM.

YUM and eDirectory 8.8 both use the `libexpat.so.0` library, and when you run YUM with one or more options, YUM returns an error in the console. To work around this error, use a text editor to comment out the following line in the `/etc/ld.so.conf.d/novell-NDSbase.conf` file and then run `ldconfig`:

```
/opt/novell/eDirectory/lib64
```

After commenting out the line and running `ldconfig`, ensure that you run the following command in a terminal window each time you start eDirectory:

```
source /opt/novell/eDirectory/bin/ndspath
```

Restart eDirectory using the same terminal. ndspath resolves the necessary path dependencies.

2.1.10 Performance Issues When Running eDirectory with BTRFS

If you install eDirectory on a SLES server within a BTRFS filesystem, you may experience performance issues when performing LDAP operations or using the NetIQ Import Conversion Export Utility (ICE). For performance reasons, it is recommended that you use the ext3 filesystem for your eDirectory server.

2.2 Configuration

The section contains problems you may encounter during the eDirectory 8.8 configuration.

- Section 2.2.1, “Loopback Referrals Are Returned By a Directory Server,” on page 18
- Section 2.2.2, “Tree Name Lookup Failed: -632 Error While Configuring eDirectory 8.8 on Linux,” on page 18
- Section 2.2.3, “Adding New Servers,” on page 18
- Section 2.2.4, “Excluding the DIB directory from Backup or Antivirus Processes,” on page 19
- Section 2.2.5, “eDirectory ndsconfig Displays an Error on RHEL 32-Bit Platform,” on page 19
- Section 2.2.6, “IP AG Certificate Does Not Get Created on SLES 11 64-Bit Platform,” on page 19

2.2.1 Loopback Referrals Are Returned By a Directory Server

When eDirectory is configured to listen on loopback addresses, the loopback addresses are stored and returned to the clients when they perform searches and other operations. The referrals are not applicable to the clients that attempts to connect from the machines other than the server. Therefore, the clients fail to connect by using those loopback referrals. However, the other referrals returned by the server still work for the clients.

Trying to connect to each loopback referrals and then choosing the correct referrals could affect the performance of the clients.

To workaround: select only one interface that eDirectory can communicates on; do not select the loopback interfaces during the install.

2.2.2 Tree Name Lookup Failed: -632 Error While Configuring eDirectory 8.8 on Linux

While configuring eDirectory 8.8 on Linux, you might get the Tree name lookup failed: -632 error. To resolve this, perform the following steps:

1. After installing the SLP package, ensure that you manually start SLP as follows:

   ```
   /etc/init.d/slpuasa start
   ```

2. After uninstalling the SLP package, ensure that you manually stop SLP as follows:

   ```
   /etc/init.d/slpuasa stop
   ```

2.2.3 Adding New Servers

You cannot add a new server into a context if its fully qualified DN length is more than 255 characters. The length restriction applies to a fully qualified DN and not to the context length. The fully qualified DN of any object can have a maximum of 255 characters.

2.2.4 Excluding the DIB directory from Backup or Antivirus Processes

After installing eDirectory, you should configure your environment to exclude the DIB directory on your eDirectory server from any antivirus or backup software processes. If you do not exclude the DIB directory from processes of this type, you may encounter corrupted DIB files or `-618 FFFFFD96 INCONSISTENT DATABASE` errors.

Use the eDirectory Backup Tool to back up your DIB directory. For more information about backing up eDirectory, see "Backing Up and Restoring NetIQ eDirectory" in the *NetIQ eDirectory 8.8 SP8 Administration Guide*.

2.2.5 eDirectory ndsconfig Displays an Error on RHEL 32-Bit Platform

eDirectory ndsconfig displays the following error on RHEL 32-bit system.

```
/opt/novell/eDirectory/lib/libsal.so.1.0.0
```

```
error while loading shared libraries: /opt/novell/lib/libccs2.so: cannot
restore segment prot after reloc: Permission denied
```

To workaround the issue: Run the following commands.

```
chcon -t textrel_shlib_t '/opt/novell/eDirectory/lib/libsal.so.1.0.0'
```

```
chcon -t textrel_shlib_t '/opt/novell/lib/libccs2.so.2.7.6'
```

2.2.6 IP AG Certificate Does Not Get Created on SLES 11 64-Bit Platform

Consider a scenario where eDirectory 8.8 SP8 has both IPv4 and IPv6 configured and only one of the them (for example, IPv4) has an entry in the `/etc/hosts` file, and the other interface is accessible from a remote machine. If you configure eDirectory to listen on both the IPs, the IP AG certificate is generated only for the IP that is listed in the `/etc/hosts` file. In this example, it is generated for IPv4.

2.3 Upgrade

- Section 2.3.1, "Upgrade Fails if the Mount Point Is Set to /var/opt/novell/eDirectory/data," on page 19
- Section 2.3.2, "Upgrading eDirectory After Applying a Patch Does Not Remove the Patch Version on a Windows System," on page 20

2.3.1 Upgrade Fails if the Mount Point Is Set to /var/opt/novell/eDirectory/data

Upgrading eDirectory by using the `ndsconfig upgrade` command fails, if mount point is set to `/var/opt/novell/eDirectory/data`.Upgrade halts and the following error message is displayed:

```
ERROR: Unable to check if the directory "/var/opt/novell/eDirectory/data_upg_bak"
already exists. If the directory exists, delete it and execute `ndsconfig upgrade -
-config-file /etc/nds.conf`to restart the upgrade operation.
```

The problem arises because during the upgrade, the `/var/opt/novell/eDirectory/data` directory is renamed to `/var/opt/novell/eDirectory/data_upg_bak` in order that no customer data is lost. In this case, `/var/opt/novell/eDirectory/data` directory is the mount point, which cannot be renamed.

To workaround this issue, do either of the following:

- Change the mount point to `/var/opt/novell/eDirectory`.
- Perform the following:
  1. Create `/var/opt/novell/eDirectory/data_upg_bak` directory.
  2. Move the files from `/var/opt/novell/eDirectory/data` to `/var/opt/novell/eDirectory/data_upg_bak`.

  IMPORTANT: Keep the `/var/opt/novell/eDirectory/data` directory empty to ensure smooth upgrade.

2.3.2 Upgrading eDirectory After Applying a Patch Does Not Remove the Patch Version on a Windows System

When you upgrade eDirectory after applying a patch, the patch version is not upgraded; but, the base version of the product is upgraded.

This issue is observed and reproduced for the following upgrade scenarios:

Table 2-1 *eDirectory Versions*

Base Product Version	Patch Version	Upgraded Version
eDirectory 873	87310	eDirectory 88 SP3
eDirectory 873		eDirectory 88 SP3
eDirectory 873		eDirectory 873 SP10
eDirectory 88 SP6	any patch	eDirectory 88 SP8

The issue occurs because eDirectory installers and patch installers in Windows are separate. Base product of eDirectory is installed via NIS framework and patches like eDirectory 8.8 SP5 Patch 2 are installed by using Nulsoft Installer Script (NSIS). Because the installers are different, only the base version of the product is upgraded; not the patch installed via NSIS.

To workaround this issue, remove the registry entry of the patch (for example: eDirectory 8.7.3 SP9/ eDirectory 8.7.3 SP10/eDirectory 8.8 SP5 patch 2 and eDirectory 8.8 SP5 patch 3) during the upgrade.

2.4 Multiple Instances

While handling multiple instances of eDirectory, you may encounter the following problems:

- Section 2.4.1, “If the First Instance is Down, HTTP Does Not Work,” on page 21
- Section 2.4.2, “eDirectory Does Not Listen on All the Configured Interfaces,” on page 21

2.4.1 If the First Instance is Down, HTTP Does Not Work

On Linux, if eDirectory is configured on a box with multiple NIC cards and if HTTP is bound to more than one interface; if the first interface goes down, HTTP would not be accessible from the remaining interfaces.

This is because the remaining interfaces will redirect the request to the first one, but the first interface is down.

To resolve this issue, if the first interface goes down, restart eDirectory.

2.4.2 eDirectory Does Not Listen on All the Configured Interfaces

Ensure that all the interfaces on which eDirectory is configured are up and connected.

2.4.3 ndsd Falls Back to Default Port if the Interface Specified is Incorrect

When using `ndsconfig new` or `ndsmanage` to create a second instance of the directory, if the interface specified is incorrect, nds tries to use the default interface. If you specify non default port (for example 1524), the interface specified is incorrect, it uses the default interface and the default port of 524.

For `n4u.server.interfaces`, if the interface specified is incorrect, then `ndsd` will try to listen on the first interface and the port number would be the one specified in `n4u.server.tcp-port`.

2.4.4 How to Rebuild .edir Directory

The `.edir` directory is used for tracking multiple instances of eDirectory. To recreate the lost or corrupted instances file (`instances.$uid` file, where `$uid` specifies the user ID of the user in the system), you must create its individual instances file.

These files must contain the absolute location of the `nds.conf` files of all the instances configured by the user. For example, a user with `uid` as 1000 must create an `/etc/opt/novell/eDirectory/conf/.edir/instances.1000` instances file with the following entries:

```
/home/user1/instance1/nds.conf
/home/user1/instance2/nds.conf
```

3 Determining the eDirectory Version Number

The following sections list ways you can determine the version of eDirectory installed on a server:

- Section 3.1, “Windows,” on page 23
- Section 3.2, “Linux,” on page 23

3.1 Windows

- Run iMonitor.

  On the Agent Summary page, click Known Servers. Then under Servers Known to Database, click Known Servers. The Agent Revision column displays the internal build number for each server. For example, an Agent Revision number for eDirectory 8.7.1 might be 10510.64.

  For information on running iMonitor, see “Accessing iMonitor” in the *NetIQ eDirectory 8.8 SP8 Administration Guide*.

- Run `NDSCons.exe`.

  In the Windows Control Panel, double-click NetIQ eDirectory Services. In the Services column, select ds.dlm, then click Configure. The Agent tabs displays both the marketing string (for example, NetIQ eDirectory 8.8.1) and the internal build number (for example, 10510.64).

- Run an eDirectory utility.

  Most eDirectory utilities have an About option on their Help menu that displays the version number of the utility (for example, Merge Graft Utility 10510.35). Some utilities include the internal build version in the main label of the utility (for example, DSRepair - Version 10510.37).

  To load an eDirectory utility (such as DSMerge or DSRepair), double-click NetIQ eDirectory Services in the Windows Control Panel. In the Services column, select the utility, then click Start.

- View the properties of an eDirectory `.dlm` file.

  Right-click the `.dlm` file in Windows Explorer, then click the Version tab in the Properties dialog box. This will display the version number of the utility. The default location for eDirectory `.dlm` files is `C:\novell\NDS`.

3.2 Linux

- Run ndsstat.

  The ndsstat utility displays information related to eDirectory servers, such as the eDirectory tree name, the fully distinguished server name, and the eDirectory version. In the following example, eDirectory 8.7.1 is the product version (marketing string), and 10510.65 is the binary version (internal build number).

```
osg-dt-srv17:/>ndsstat
Tree Name: SNMP-HPUX-RASH
Server Name: .CN=osg-dt-srv17.O=novell.T=SNMP-HPUX-RASH.
Binary Version: 10510.65
Root Most Entry Depth: 0
Product Version: NDS/Linux - NDS eDirectory v8.8.8 [DS]
```

For information on running ndsstat, see "NetIQ eDirectory Linux Commands and Usage" in the *NetIQ eDirectory 8.8 SP8 Administration Guide*, or the ndsstat man page (`ndsstat.1m`).

- Run `ndsd --version`.

  For information on running ndsd, see "NetIQ eDirectory Linux Commands and Usage" in the *NetIQ eDirectory 8.8 SP8 Administration Guide*, or the ndsd man page (`ndsd.1m`).

- Run iMonitor.

  On the Agent Summary page, click Known Servers. Then under Servers Known to Database, click Known Servers. The Agent Revision column displays the internal build number for each server. For example, an Agent Revision number for NetIQ eDirectory 8.8.1 might be 10510.64.

  For information on running iMonitor, see "Accessing iMonitor" in the *NetIQ eDirectory 8.8 SP8 Administration Guide*.

- Run `rpm -qi NDSserv`.

  Entering this command will display similar information to `ndsd --version`.

4 Log Files

This section contains information on the following log files:

- Section 4.1, “modschema.log,” on page 25
- Section 4.2, “dsinstall.log,” on page 25
- Section 4.3, “ndsd.log,” on page 25

4.1 modschema.log

The `modschema.log` file contains the results of all schema extensions that are applied when an eDirectory server is installed into an existing tree. Each line of the log states which class or attribute is being added or modified and gives the status of the modification attempt.

This log is created or overwritten each time the install process is run, so it only represents the results of the last attempt. In addition to the eDirectory schema extensions, this log contains the results of any other schema extensions, such as the Lightweight Directory Access Protocol (LDAP) or SAS, applied by the DSINSTALL front end prior to adding the new eDirectory server.

This log will not be generated when a standalone server is installed or if the version of the target server is eDirectory 7.0.1 or later.

4.2 dsinstall.log

The first part of the `dsinstall.log` file lists environment variables that are set. The second part contains status messages documenting the eDirectory installation process.

4.3 ndsd.log

The `ndsd.log` file contains information about eDirectory server-related messages, such as, server shutdown and start messages and PKI and LDAP services start and shutdown messages. By default, this file is located in the `/var/opt/novell/eDirectory/log` directory.

You can increase the debug level for the `ndsd.log` file by modifying the following variable in the `nds.conf` file from `/etc/opt/novell/eDirectory/conf/nds.conf` file.

```
n4u.server.log-levels=Logxxxx
```

For more information on ndsd log levels, see “Managing Error Logging in eDirectory 8.8” in the *NetIQ eDirectory 8.8 SP8 What's New Guide*.

5 Troubleshooting LDIF Files

The NetIQ Import Conversion Export utility lets you easily import LDIF files into and export LDIF files from eDirectory. For more information, see "NetIQ Import Conversion Export Utility" in the *NetIQ eDirectory 8.8 SP8 Administration Guide*.

In order for an LDIF import to work properly, you must start with an LDIF file that the NetIQ Import Conversion Export utility can read and process. This section describes the LDIF file format and syntax and provides examples of correct LDIF files.

- Section 5.1, "Understanding LDIF," on page 27
- Section 5.2, "Debugging LDIF Files," on page 34
- Section 5.3, "Using LDIF to Extend the Schema," on page 39
- Section 5.4, "ldif2dib Limitations," on page 43

5.1 Understanding LDIF

LDIF is a widely used file format that describes directory information or modification operations that can be performed on a directory. LDIF is completely independent of the storage format used within any specific directory implementation, and is typically used to export directory information from and import data to LDAP servers.

LDIF is usually easy to generate. This makes it possible to use tools like awk or perl to move data from a proprietary format into an LDAP directory. You can also write scripts to generate test data in LDIF format.

5.1.1 LDIF File Format

NetIQ Import Conversion Export imports require LDIF 1 formatted files. The following are the basic rules for an LDIF 1 file:

- The first non-comment line must be version: 1.
- A series of one or more records follows the version.
- Each record is composed of fields, one field per line.
- Lines are separated by either a new line or a carriage return/new line pair.
- Records are separated by one or more blank lines.
- There are two distinct types of LDIF records: content records and change records. An LDIF file can contain an unlimited number of records, but they all must be of the same type. You can't mix content records and change records in the same LDIF file.
- Any line beginning with the pound sign (#) is a comment and is ignored when processing the LDIF file.

5.1.2 LDIF Content Records

An LDIF content record represents the contents of an entire entry. The following is an example of an LDIF file with four content records:

```
 1  version: 1
 2  dn: c=US
 3  objectClass: top
 4  objectClass: country
 5
 6  dn: l=San Francisco, c=US
 7  objectClass: top
 8  objectClass: locality
 9  st: San Francisco
10
11 dn: ou=Artists, l=San Francisco, c=US
12 objectClass: top
13 objectClass: organizationalUnit
14 telephoneNumber: +1 415 555 0000
15
16 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US
17 sn: Michaels
18 givenname: Peter
19 objectClass: top
20 objectClass: person
21 objectClass: organizationalPerson
22 objectClass: iNetOrgPerson
23 telephonenumber: +1 415 555 0001
24 mail: Peter.Michaels@aaa.com
25 userpassword: Peter123
26
```

This LDIF file is composed of the following parts:

Component	Description
Version Specifier	The first line of an LDIF file contains the version. Zero or more spaces are allowed between the colon and the version number, which is currently defined to be 1. If the version line is missing, any application processing the LDIF file is allowed to assume that the file is version 0. It's also possible that the LDIF file could be rejected as syntactically incorrect. NetIQ utilities that process LDIF assume a file version of 0 when the version line is missing.
Distinguished Name Specifier	The first line of every content record (lines 2, 6, 11, and 16 in the example above) specifies the DN of the entry that it represents. The DN specifier must take one of the following two forms: ◆ dn: *safe_UTF-8_distinguished_name* ◆ dn:: *Base64_encoded_distinguished_name*
Line Delimiters	The line separator can be either a line feed or a carriage return/line feed pair. This resolves a common incompatibility between Linux and Solaris text files, which use a line feed as the line separator, and MS-DOS* and Windows text files, which use a carriage return/line feed pair as the line separator.

Component	Description
Record Delimiters	Blank lines (lines 5, 10, 15, and 26 in the example above) are used as record delimiters. Every record in an LDIF file including the last record must be terminated with a record delimiter (one or more blank lines). Although some implementations will silently accept an LDIF file without a terminating record delimiter, the LDIF specification requires it.
Attribute Value Specifier	All other lines in a content records are value specifiers. Value specifiers must take on one of the following three forms: ◆ Attribute description: *value* ◆ Attribute description:: *Base64_encoded_value* ◆ Attribute description: < *URL*

5.1.3 LDIF Change Records

LDIF change records contain modifications to be made to a directory. Any of the LDAP update operations (add, delete, modify, and modify DN) can be represented in an LDIF change record.

LDIF change records use the same format for the distinguished name specifier, attribute value specifier, and record delimiter as LDIF content records. (See "LDIF Content Records" on page 28 for more information.) The presence of a `changetype` field is what distinguishes an LDIF change record from an LDIF content record. A `changetype` field identifies the operation specified by the change record.

A `changetype` field can take one of the following five forms:

Form	Description
changetype: add	A keyword indicating that the change record specifies an LDAP add operation.
changetype: delete	A keyword indicating that the change record specifies an LDAP delete operation.
changetype: moddn	A keyword indicating that the change record specifies an LDAP modify DN operation if the LDIF processor is bound to the LDAP server as a version 3 client or a modify RDN operation if the LDIF processor is bound to the LDAP server as a version 2 client.
changetype: modrdn	A synonym for the moddn change type.
changetype: modify	A keyword indicating that the change record specifies an LDAP modify operation.

The Add Change Type

An add change record looks just like a content change record (see "LDIF Content Records" on page 28) with the addition of the changetype: add field immediately before any attribute value fields.

All records must be the same type. You can't mix content records and change records.

```
 1 version: 1
 2 dn: c=US
 3 changetype: add
 4 objectClass: top
 5 objectClass: country
 6
 7 dn: l=San Francisco, c=US
 8 changetype: add
 9 objectClass: top
10 objectClass: locality
11 st: San Francisco
12
14 dn: ou=Artists, l=San Francisco, c=US
15   changetype: add
16 objectClass: top
17 objectClass: organizationalUnit
18 telephoneNumber: +1 415 555 0000
19
20 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US
21 changetype: add
22 sn: Michaels
23 givenname: Peter
24 objectClass: top
25 objectClass: person
26 objectClass: organizationalPerson
27 objectClass: iNetOrgPerson
28 telephonenumber: +1 415 555 0001
29 mail: Peter.Michaels@aaa.com
30 userpassword: Peter123
31
```

The Delete Change Type

Because a delete change record specifies the deletion of an entry, the only fields required for a delete change record are the distinguished name specifier and a delete change type.

The following is an example of an LDIF file used to delete the four entries created by the LDIF file shown in "The Add Change Type" on page 29.

IMPORTANT: To delete entries you have previously added, reverse the order of the entries. If you do not do this, the delete operation fails because the container entries are not empty.

```
 1 version: 1
 2 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US
 3 changetype: delete
 4
 5 dn: ou=Artists, l=San Francisco, c=US
 8   changetype: delete
 9
10 dn: l=San Francisco, c=US
11 changetype: delete
12
13 dn: c=US
14 changetype: delete
15
```

The Modify Change Type

The modify change type lets you to specify the addition, deletion, and replacement of attribute values for an entry that already exists. Modifications take one of the following three forms:

Element	Description
add: attribute type	A keyword indicating that subsequent attribute value specifiers for the attribute type should be added to the entry.
delete: attribute type	A keyword indicating that values of the attribute type are to be deleted. If attribute value specifiers follow the delete field, the values given are deleted. If no attribute value specifiers follow the delete field, then all values are deleted. If the attribute has no values, this operation will fail, but the desired effect will still be achieved because the attribute had no values to be deleted.
replace: attribute type	A keyword indicating that the values of the attribute type are to be replaced. Any attribute value specifiers that follow the replace field become the new values for the attribute type. If no attribute value specifiers follow the replace field, the current set of values is replaced with an empty set of values (which causes the attribute to be removed). Unlike the delete modification specifier, if the attribute has no values, the replace will still succeed. The net effect in both cases is the same.

The following is an example of a modify change type that will add an additional telephone number to the `cn=Peter` Michaels entry.

```
 1 version: 1
 2 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US
 3 changetype: modify
 4 # add the telephone number to cn=Peter Michaels
 4 add: telephonenumber
 5 telephonenumber: +1 415 555 0002
 6
```

Just as you can combine a mixture of modifications in a single LDAP modify request, you can specify multiple modifications in a single LDIF record. A line containing only the hyphen (-) character is used to mark the end of the attribute value specifications for each modification specifier.

The following example LDIF file contains a mixture of modifications:

```
 1 version: 1
 2
 3 # An empty line to demonstrate that one or more
 4 # line separators between the version identifier
 5 # and the first record is legal.
 6
 7 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US
 8 changetype: modify
 9 # Add an additional telephone number value.
10 add: telephonenumber
11 telephonenumber: +1 415 555 0002
12 -
13 # Delete the entire fascimiletelephonenumber attribute.
14 delete: facsimileTelephoneNumber
15 -
16 # Replace the existing description (if any exists)
17 # with two new values.
18 replace: description
```

```
19 description: guitar player
20 description: solo performer
21 -
22 # Delete a specific value from the telephonenumber
23 # attribute.
24 delete: telephonenumber
25 telephonenumber: +1 415 555 0001
26 -
27 # Replace the existing title attribute with an empty
28 # set of values, thereby causing the title attribute to
29 # be removed.
30 replace: title
31 -
32
```

The Modify DN Change Type

The modify DN change type lets you rename an entry, move it, or both. This change type is composed of two required fields and one optional field.

Field	Description
newrdn (required)	Gives the new name for the entry that will be assigned while processing this record. The new RDN specifier must take of the following two forms: ◆ newrdn: *safe_UTF-8_relative_distinguished_name* ◆ newrdn:: *Base64_encoded_relative_ distinguished_name* The new RDN specifier is required in all LDIF records with a modify DN change type.
deleteoldrdn (required)	The delete old RDN specifier is a flag that indicates whether the old RDN should be replaced by the newrdn or if it should be kept. It takes one of the two following forms: ◆ deleteoldrdn: 0 Indicates that the old RDN value should be kept in the entry after it is renamed. ◆ deleteoldrdn: 1 Indicates that the old RDN value should be deleted when the entry is renamed.
newsuperior (optional)	The new superior specifier gives the name of the new parent that will be assigned to the entry while processing the modify DN record. The new superior specifier must take of the following two forms: ◆ newsuperior: *safe_UTF-8_distinguished_name* ◆ newsuperior:: *Base64_encoded_distinguished_ name* The new superior specifier is optional in LDIF records with a modify DN change type. It is only given in cases where you want to re-parent the entry.

The following is an example of a modify DN change type that shows how to rename an entry:

```
 1 version: 1
 2
 3 # Rename ou=Artists to ou=West Coast Artists, and leave
 4 # its old RDN value.
 5 dn: ou=Artists,l=San Francisco,c=US
 6 changetype: moddn
 7 newrdn: ou=West Coast Artists
 8 deleteoldrdn: 1
 9
```

The following is an example of a modify DN change type that shows how to move an entry:

```
 1 version: 1
 2
 3 # Move cn=Peter Michaels from
 4 # ou=Artists,l=San Francisco,c=US to
 5 # ou=Promotion,l=New York,c=US and delete the old RDN.
 5 dn: cn=Peter Michaels,ou=Artists,l=San Francisco,c=US
 6 changetype: moddn
 7 newrdn: cn=Peter Michaels
 8 deleteoldrdn: 1
 9 newsuperior: ou=Promotion,l=New York,c=US
10
```

The following is an example of a modify DN change type that shows how to move an entry and rename it at the same time:

```
 1 version: 1
 2
 3 # Move ou=Promotion from l=New York,c=US to
 4 # l=San Francisco,c=US and rename it to
 5 # ou=National Promotion.
 5 dn: ou=Promotion,l=New York,c=US
 6 changetype: moddn
 7 newrdn: ou=National Promotion
 8 deleteoldrdn: 1
 9 newsuperior: l=San Francisco,c=US
10
```

IMPORTANT: The LDAP 2 modify RDN operation doesn't support moving entries. If you try to move an entry using the LDIF `newsuperior` syntax with an LDAP 2 client, the request will fail.

5.1.4 Line Folding within LDIF Files

To fold a line in an LDIF file, simply insert a line separator (a new line or a carriage return/new line pair) followed by a space at the place where you want the line folded. When the LDIF parser encounters a space at a beginning of the line, it knows to concatenate the rest of the data on the line with the data on the previous line. The leading space is then discarded.

You should not fold lines in the middle of a multibyte UTF-8 character.

The following is an example of an LDIF file with a folded line (see lines 13 and 14):

```
 1 version: 1
 2 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US
 3 sn: Michaels
 4 givenname: Peter
 5 objectClass: top
 6 objectClass: person
 7 objectClass: organizationalPerson
 8 objectClass: iNetOrgPerson
 9 telephonenumber: +1 415 555 0001
10 mail: Peter.Michaels@aaa.com
11 userpassword: Peter123
12 description: Peter is one of the most popular music
13  ians recording on our label. He's a big concert dr
14  aw, and his fans adore him.
15
```

5.1.5 Hashed Password Representation in LDIF Files

The hashed password is represented as base64 data in the LDIF file. The attribute name `userpassword` should be followed with the name of the encryption used for hashing the password. This name should be given within a pair of flower brackets "{ }" as shown below:

Example 1

For SHA hashed passwords:

```
1 version: 1 2 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US 3 sn:
Michaels 4 userpassword: {SHA}xcbdh46ngh37jsd0naSFDedjAS30dm5 objectclass:
inetOrgPerson
```

Example 2

For SSHA hashed passwords:

```
1 version: 1 2 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US 3 sn:
Michaels 4 userpassword: {SSHA}sGs948DFGkakdfkasdDF34DF4dS3skl5DFS5 objectclass:
inetOrgPerson
```

Example 3

For Digest MD5 hashed passwords:

```
1 version: 1 2 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US 3 sn:
Michaels 4 userpassword: {MD5}a45lkSDF234SDFG62dsfsf2DG2QEvgdmnk4305 objectclass:
inetOrgPerson
```

5.2 Debugging LDIF Files

- "Enabling Forward References" on page 35
- "Checking the Syntax of LDIF Files" on page 37
- "Resolving the LDIF Error File" on page 38
- "Using LDAP SDK Debugging Flags" on page 38

If you have problems with an LDIF file, consider the following:

5.2.1 Enabling Forward References

You might occasionally encounter LDIF files in which a record to add one entry comes before a record to add its parents. When this happens, an error is generated because the new entry's parent does not exist when the LDAP server attempts to add the entry.

To solve this problem, simply enable the use of forward references. When you enable the creation of forward references and an entry is going to be created before its parent exists, a placeholder called a forward reference is created for the entry's parent to allow the entry to be successfully created. If a later operation creates the parent, the forward reference is changed into a normal entry.

It is possible that one or more forward references will remain after your LDIF import is complete (if, for example, the LDIF file never created the parent for an entry). In this case, the forward reference will appear as an Unknown object in iManager. Although you can search on a forward reference entry, you cannot read attributes (except objectClass) from the forward reference entry because it does not have any attributes or attribute values. However, all LDAP operations will work normally on the real object entries located below the forward reference.

Identifying Forward Reference Entries

Forward reference entries have an object class of Unknown and also have their internal NDS EF_REFERENCE entry flag set. In iManager, entries with an object class of Unknown are represented by a round yellow icon with a question mark in the center. You can use LDAP to search for objects with an Unknown object class, although there is currently no way to access the entry flag settings through LDAP to be sure that they are forward reference entries.

Changing Forward Reference Entries into Normal Objects

You can change a forward reference entry into an normal object by simply creating it (using, for example, an LDIF file or an LDAP client request). When you ask eDirectory to create an entry that already exists as a forward reference, eDirectory transforms the existing forward reference entry into the object you asked it to create.

Using the NetIQ eDirectory Import Convert Export Wizard

To enable forward references during an LDIF import:

1. In NetIQ iManager, click the *Roles and Tasks* button  .
2. Click *eDirectory Maintenance* > *Import Convert Export Wizard*.
3. Click *Import Data* from *File on Disk*, then click *Next*.
4. Select *LDIF* as the type of file you want to import.
5. Specify the name of the file containing the data you want to import, specify the appropriate options, then click *Next*.
6. Specify the LDAP server where the data will be imported.
7. Add the appropriate options, as described in the following table:

Option	Description
Server DNS name/IP address	DNS name or IP address of the destination LDAP server
Port	Integer port number of the destination LDAP server
DER File	Name of the DER file containing a server key used for SSL authentication
Login method	Authenticated Login or Anonymous Login (for the entry specified in the User DN field)
User DN	Distinguished name of the entry that should be used when binding to the server-specified bind operation
Password	Password attribute of the entry specified in the User DN field

8 Under *Advanced Settings*, click *Allow Forward References*.

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

To enable forward references during a data-to-data server migration:

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

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

3 Click *Migrate Data Between Servers*, then click *Next*.

4 Specify the LDAP server holding the entries you want to migrate.

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

Option	Description
Server DNS name/IP address	DNS name or IP address of the source LDAP server
Port	Integer port number of the source LDAP server
DER file	Name of the DER file containing a server key used for SSL authentication
Login method	Authenticated Login or Anonymous Login (for the entry specified in the User DN field)
User DN	Distinguished name of the entry that should be used when binding to the server-specified bind operation
Password	Password attribute of the entry specified in the User DN field

6 Under *Advanced Settings*, click *Allow Forward References*.

7 Click *Next*.

8 Specify the search criteria (described below) for the entries you want to migrate:

Option	Description
Base DN	Base distinguished name for the search request If this field is left empty, the base DN defaults to " " (empty string).
Scope	Scope of the search request
Filter	RFC 2254-compliant search filter The default is `objectclass=*`.
Attributes	Attributes you want returned for each search entry

9 Click *Next*.

10 Specify the LDAP server where the data will be migrated.

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

NOTE: Ensure that the schema is consistent across LDAP Services.

Using the NetIQ Import Conversion Export Utility Command Line Interface

To enable forward references in the command line interface, use the -F LDAP destination handler option.

For more information, see "LDIF Destination Handler Options" in the *NetIQ eDirectory 8.8 SP8 Administration Guide*.

5.2.2 Checking the Syntax of LDIF Files

You can check the syntax of an LDIF file before you process the records in the file by using the Display Operations But Do Not Perform LDIF source handler option.

The LDIF source handler always checks the syntax of the records in an LDIF file as it processes them. Using this option disables the processing of the records and lets you verify the syntax.

Using the NetIQ eDirectory Import Convert Export Wizard

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

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

3 Click *Import Data* from *File on Disk*, then click *Next*.

4 Select *LDIF* as the type of file you want to import.

5 Specify the name of the file containing the data you want to import, specify the appropriate options.

6 Under *Advanced Settings*, click *Display Operations But Do Not Perform*, then click *Next*.

7 Specify the LDAP server where the data will be imported.

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

Option	Description
Server DNS name/IP address	DNS name or IP address of the destination LDAP server
Port	Integer port number of the destination LDAP server
DER File	Name of the DER file containing a server key used for SSL authentication
Login method	Authenticated Login or Anonymous Login (for the entry specified in the User DN field)
User DN	Distinguished name of the entry that should be used when binding to the server-specified bind operation
Password	Password attribute of the entry specified in the User DN field

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

Using the NetIQ Import Conversion Export Utility Command Line Interface

To check the syntax of an LDIF file in the command line interface, use the -n LDIF source handler option.

For more information, see "LDIF Source Handler Options" in the *NetIQ eDirectory 8.8 SP8 Administration Guide*.

5.2.3 Resolving the LDIF Error File

The NetIQ Import Conversion Export utility automatically creates an LDIF file listing any records that failed processing by the destination handler. You can edit the LDIF error file generated by the utility, fix the errors in the command line utility, then reapply it to the server to finish an import or data migration that contained failed records.

To configure error log options in the command line utility, use the `-l` general option.

For more information, see "General Options" in the *NetIQ eDirectory 8.8 SP8 Administration Guide*.

5.2.4 Using LDAP SDK Debugging Flags

To understand some LDIF problems, you might need to see how the LDAP client SDK is functioning. You can set the following debugging flags for the LDAP source handler, the LDAP destination handler, or both.

Value	Description
0x0001	Trace LDAP function calls.
0x0002	Print information about packets.
0x0004	Print information about arguments.
0x0008	Print connections information.
0x0010	Print BER encoding and decoding information.
0x0020	Print search filter information.

Value	Description
0x0040	Print configuration information.
0x0080	Print ACL information.
0x0100	Print statistical information.
0x0200	Print additional statistical information.
0x0400	Print shell information.
0x0800	Print parsing information.
0xFFFF (-1 Decimal)	Enable all debugging options.

To enable this functionality, use the `-e` option for the LDAP source and LDAP destination handlers. The integer value you give for the `-e` option is a bitmask that enables various types of debugging information in the LDAP SDK.

For more information, see"LDAP Source Handler Options" and "LDAP Destination Handler Options" in the *NetIQ eDirectory 8.8 SP8 Administration Guide*.

5.3 Using LDIF to Extend the Schema

Because LDIF can represent LDAP update operations, you can use LDIF to modify the schema.

5.3.1 Adding a New Object Class

To add a class, simply add an attribute value that conforms to the specification for `NDSObjectClassDescription` to the `objectClasses` attribute of the `subschemaSubentry`.

```
NDSObjectClassDescription = "(" whsp
   numericoid whsp
   [ "NAME" qdescrs ]
   [ "DESC" qdstring ]
   [ "OBSOLETE" whsp ]
   [ "SUP" oids ]
   [ ( "ABSTRACT" / "STRUCTURAL" / "AUXILIARY" ) whsp ]
   [ "MUST" oids ]
   [ "MAY" oids ]
   [ "X-NDS_NOT_CONTAINER" qdstrings ]
   [ "X-NDS_NONREMOVABLE" qdstrings ]
   [ "X-NDS_CONTAINMENT" qdstrings ]
   [ "X-NDS_NAMING" qdstrings ]
   [ "X-NDS_NAME" qdstrings ]
   whsp ")"
```

The following example LDIF file adds the `person` objectClass to the schema:

```
 1 version: 1
 2 dn: cn=schema
 3 changetype: add
 4 objectClasses: ( 2.5.6.6 NAME 'person' DESC 'Standard
 5   ObjectClass' SUP ndsLoginProperties STRUCTURAL MUST
 6   (cn $ sn) MAY (description $ seeAlso $ telephoneNum
 7  ber $ fullName $ givenName $ initials $ uid $ userPa
 8  ssword) X-NDS_NAMING ('cn' 'uid') X-NDS_CONTAINMENT
 9  ('organization' 'organizationalUnit' 'domain') X-NDS
10  _NAME 'Person' X-NDS_NOT_CONTAINER '1' X-NDS_NONREMO
11  VABLE '1')
12
```

Mandatory Attributes

Mandatory attributes are listed in the MUST section of the object class description. For the person object class, the mandatory attributes are cn and sn.

Optional Attributes

Optional attributes are listed in the MAY section of the object class description. The optional attributes in the person object class are description, seeAlso, telephoneNumber, fullName, givenName, initials, uid, and userPassword.

NOTE: The userPassword attribute cannot be used as an optional (MAY) attribute. The operation will fail if you try to use it as a mandatory (MUST) attribute in the new objectClass using this LDIF format to extend the schema.

Containment Rules

The object classes that can contain the object class being defined are given in the X-NDS_CONTAINMENT section of the object class description. The person object class can be contained by the organization, organizationalUnit, and domain object classes.

5.3.2 Adding a New Attribute

To add an attribute, simply add an attribute value that conforms to the specification for NDSAttributeTypeDescription to the attributes attribute of the subschemaSubentry.

```
NDSAttributeTypeDescription = "(" whsp
  numericoid whsp  ; AttributeType identifier
  [ "NAME" qdescrs ]  ; name used in AttributeType
  [ "DESC" qdstring ]  ; description
  [ "OBSOLETE" whsp ]
  [ "SUP" woid ]  ; derived from this other AttributeType
  [ "EQUALITY" woid]  ; Matching Rule name
  [ "ORDERING" woid]  ; Matching Rule name
  [ "SUBSTR" woid ]  ; Matching Rule name
  [ "SYNTAX" whsp noidlen whsp ] ;  Syntax OID
  [ "SINGLE-VALUE" whsp ]  ; default multi-valued
  [ "COLLECTIVE" whsp ]  ; default not collective
  [ "NO-USER-MODIFICATION" whsp ] ;  default user modifiable
  [ "USAGE" whsp AttributeUsage ] ;  default userApplications
  [ "X-NDS_PUBLIC_READ" qdstrings ]
                            ; default not public read ('0')
  [ "X-NDS_SERVER_READ" qdstrings ]
                             ; default not server read ('0')
  [ "X-NDS_NEVER_SYNC" qdstrings ]
                              ; default not never sync ('0')
  [ "X-NDS_NOT_SCHED_SYNC_IMMEDIATE" qdstrings ]
                            ;  default sched sync immediate ('0')
  [ "X-NDS_SCHED_SYNC_NEVER" qdstrings ]
                                 ;  default schedule sync ('0')
  [ "X-NDS_LOWER_BOUND" qdstrings ]
                             ;  default no lower bound('0')
                                  ;(upper is specified in SYNTAX)
  [ "X-NDS_NAME_VALUE_ACCESS" qdstrings ]
                          ; default not name value access ('0')
  [ "X-NDS_NAME" qdstrings ]  ; legacy NDS name
  whsp ")"
```

The following example LDIF file adds the title attribute type to the schema:

```
1 version: 1
2 dn: cn=schema
3 changetype: add
4 attributeTypes: ( 2.5.4.12 NAME 'title' DESC 'Standa
5  rd Attribute' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{
6  64} X-NDS_NAME 'Title' X-NDS_NOT_SCHED_SYNC_IMMEDIA
7  TE '1' X-NDS_LOWER_BOUND '1')
8
```

Single-Valued versus Multivalued

An attribute defaults to multivalued unless it is explicitly made single-valued. The following example LDIF file makes title single-valued by adding the `SINGLE-VALUE` keyword after the `SYNTAX` section:

```
1 version: 1
2 dn: cn=schema
3 changetype: add
4 attributeTypes: ( 2.5.4.12 NAME 'title' DESC 'Standa
5  rd Attribute' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{
6  64} SINGLE-VALUE X-NDS_NAME 'Title' X-NDS_NOT_SCHED
7  _SYNC_IMMEDIATE '1' X-NDS_LOWER_BOUND '1')
8
```

Adding an Optional Attribute to an Existing Object Class

Although adding new schema elements is an acceptable practice, modifying or extending existing schema elements is usually dangerous. Because every schema element is uniquely identified by an OID, when you extend a standard schema element, you effectively create a second definition for the element even though it still uses the original OID. This can cause incompatibility problems.

There are times when it is appropriate to change schema elements. For example, you might need to extend or modify new schema elements as you refine them during development. Instead of adding new attributes directly to a class, you should generally use auxiliary classes only to

- Add new attributes to an existing object class.
- Subclass an existing object class.

5.3.3 Adding or Removing Auxiliary Classes

The following sample LDIF file creates two new attributes, creates an auxiliary class with these new attributes, then adds an `inetOrgPerson` entry with the `auxiliary` class as an object class of the entry and with values for the `auxiliary` class attributes.

```
version: 1
# Add an attribute to track a bear's hair. The attribute is
# multi-valued, uses a case ignore string syntax,
# and has public read rights
# Values may include: long hair, short, curly, straight,
# none, black, and brown
# X-NDS_PUBLIC_READ '1' The 1 allows public read,
# 0 denies public read
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: ( 2.16.840.1.113719.1.186.4.10 NAME
'bearHair' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
X-NDS_PUBLIC_READ '1' )

# add an attribute to store a bear's picture
dn: cn=schema
```

```
changetype: modify
add: attributeTypes
attributeTypes: ( 2.16.840.1.113719.1.186.4.11 NAME
'bearPicture' SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
SINGLE-VALUE )

# create an Auxiliary class for the bearfeatures
dn: cn=schema
changetype: modify
add: objectclasses
objectclasses: (2.16.840.1.113719.1.186.6.101 NAME
'bearFeatures' MAY (bearHair $ bearPicture) AUXILIARY)

# now create a user named bobby
dn: cn=bobby,o=bearcave
changetype: add
cn: bobby
sn: bear
givenName: bobby
bearHair: Short
bearHair: Brown
bearHair: Curly
bearPicture:< file:///c:/tmp/alien.jpg
objectClass: top
objectClass: person
objectClass: inetOrgPerson
objectClass: bearFeatures

# now create a person named john that will later be changed
# into a bear when bearFeatures is added to its objectClass
# list
dn: cn=john,o=bearcave
changetype: add
cn: John
sn: bear
givenName: john
objectClass: top
objectClass: person
objectClass: inetOrgPerson

# now morph john into a bear by adding bearFeatures
dn: cn=john,o=bearcave
changetype: modify
add: objectClass
objectClass: bearFeatures
-
add: bearHair
bearHair: long
bearHair: black
#bearPicture:< file:///c:/tmp/john.jpg>
-

# to morph john back to a person, simply delete the
# objectClass bearFeatures
dn: cn=john,o=bearcave
changetype: modify
delete: objectClass
objectClass: bearFeatures
```

When removing auxiliary classes, you don't have to delete all of the values associated with the `auxiliary` class when you remove the `auxiliary` class from the objectClass list. eDirectory does this automatically.

If the `auxiliary` class had `MUST` attributes, they must all be specified in the same modify operation that adds the `auxiliary` class to the objectClass list, or the modification will fail.

Known Problems with XML Parsing

XML processing of any LDIF Record (LDIF format or records generated from LDAP server) will not succeed if the individual records will not satisfy all the XML rules specified in the XML file.

5.4 ldif2dib Limitations

- Section 5.4.1, “Simple Password LDIF,” on page 43
- Section 5.4.2, “Schema,” on page 43
- Section 5.4.3, “ACL Templates,” on page 44
- Section 5.4.4, “Signal Handler,” on page 44

5.4.1 Simple Password LDIF

On Windows, while uploading LDIF with a simple password, ldif2dib might fail if the NICI keys in the `system` and ***`Administrator`*** folders are not in sync.

To work around this issue, use the following procedure to access the keys in the `nici/system` folder:

1. Go to the `C:\Windows\system32\novell\nici\` folder (for 32-bit NICI).

   or

   Go to the `C:\Windows\SysWOW64\novell\nici\` folder (for 64-bit NICI).
2. Back up the files in the ***`Administrator`*** folder.
3. Go to the *Security* tab in the Properties window of the `system` folder.
4. Select *Advanced Options* and go to the *Owner* tab.
5. Select *Administrator*.
6. Go back to the *Security* tab and add Administrator to the list.
7. Repeat Step 3 through Step 6 to get read access to all the files inside the system folder.
8. Overwrite the files in the ***`Administrator`*** folder with the ones in the `system` folder.
9. After the upload is done, copy the backed-up files to the ***`Administrator`*** folder.
10. Change the Administrator’s access to the `system` folder and also the files within the folder.

5.4.2 Schema

The LDIF file should mention all the object classes that an entry belongs to. You should also include the classes that an entry belongs to because of inheritance of classes. For example, an entry of type `inetOrgPerson` has following syntax in the LDIF file:

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

5.4.3 ACL Templates

Objects bulkloaded using the `ldif2dib` utility are not added with ACLs that are specified in the ACL templates for the object class of the object.

5.4.4 Signal Handler

You can temporarily suspend the offline bulkload operation by pressing the s or S key. You can use the Escape key (Esc) to stop the bulkload operation.

6 Troubleshooting SNMP

This section includes information for troubleshooting SNMP on all platforms.

- Section 6.1, “Traps Might Not Get Generated As Expected,” on page 45
- Section 6.2, “SNMP Group Object,” on page 45
- Section 6.3, “SNMP Initializing Errors,” on page 46
- Section 6.4, “SNMP Subagent Does Not Start,” on page 46
- Section 6.5, “LDAP SNMP Statistics Not Reported,” on page 46
- Section 6.6, “Segmentation Fault Error while Accessing the Subagent,” on page 46
- Section 6.7, “SNMP Issues,” on page 46

6.1 Traps Might Not Get Generated As Expected

Traps are sent only if the corresponding verb request is received by the server. They are not sent in any other cases. For example, `ndsDeleteAttribute` is sent only when the `ndsRemoveEntry` (trap number 108) request is sent. But an application can always read the ACLs and decide to check whether the user has sufficient rights to perform the delete operation. In this case, the `ndsDeleteAttribute` trap is not generated. However, you can use iMonitor to view the verb statistics on a particular server.

To get the traps for all occurrences, set the time interval to zero.

You can enable traps to send only on failure conditions. You can enable traps to get them under all conditions.

ndssnmpsa must be restarted when the master agent is restarted

To restart ndssnmpsa, stop ndssnmpsa and then start it again.

To stop ndssnmpsa, enter the following:

Linux: `/etc/init.d/ndssnmpsa stop`

To start ndssnmpsa, enter the following:

Linux: `/etc/init.d/ndssnmpsa start`

6.2 SNMP Group Object

If the installation of the SNMP Group object fails, you can rectify this problem by executing the following command on the server console:

```
ndsconfig add -m snmp
```

6.3 SNMP Initializing Errors

eDirectory SNMP initialization component. Error code: -255

or

Initialization failure. Error code: -255

The possible cause could be that you have not specified `hostname:port` or `IP_address:port` as a paramater to the `SERVER` command in eDirectory SNMP configuration file.

The eDirectory SNMP configuration file is `ndssnmp.cfg`. It is located in the following directories:

- Linux: `/etc/opt/novell/eDirectory/conf/ndssnmp/`
- Windows: *`install_directory`*`\SNMP\`

6.4 SNMP Subagent Does Not Start

While starting the SNMP subagent you might get a segmentation error. This might be because of extra spaces in the `ndssnmp.cfg` file. Remove the spaces and start ndssnmpsa.

6.5 LDAP SNMP Statistics Not Reported

When anonymous bind is disabled, LDAP SNMP statistics are not reported.

To resolve this issue:

1. Allow anonymous bind.
2. Start the subagent.
3. Disable/disallow anonymous bind.

6.6 Segmentation Fault Error while Accessing the Subagent

When a user tries to start the subagent (`ndssnmpsa`) by using an incorrect eDirectory password, a segmentation fault error occurs.

To avoid getting this error, ensure that you use the correct eDirectory password while starting the subagent.

6.7 SNMP Issues

- Section 6.7.1, “Issues After Upgrading from eDirectory 8.7.3 to eDirectory 8.8,” on page 47
- Section 6.7.2, “Errors While Starting the NDS Subagent,” on page 47
- Section 6.7.3, “Restarting ndssnmpsa,” on page 47
- Section 6.7.4, “Errors While Starting ndssnmpsa,” on page 48
- Section 6.7.5, “Errors While Stopping ndssnmpsa,” on page 48
- Section 6.7.6, “Compiling edir.mib,” on page 48

- Section 6.7.7, “Modifying the SNMP Configuration File,” on page 48
- Section 6.7.8, “Using SNMP After a New Tree Installation,” on page 48
- Section 6.7.9, “SNMP Object Creation Error on Windows Server,” on page 49
- Section 6.7.10, “Uninstalling SNMP with eDirectory Uninstallation,” on page 49

6.7.1 Issues After Upgrading from eDirectory 8.7.3 to eDirectory 8.8

After upgrading from eDirectory 8.7.3 to eDirectory 8.8, you might get the following error:

```
%%% Attempting to restart the NetIQ eDirectory SNMP subagent (ndssnmpsa)...
Starting NDS SNMP Subagent ...
Initialization failure. Error code : -255
Please Wait...
Done

%%% Unable to start ndssnmpsa... Please try starting it manually...
```

This error occurs because with eDirectory 8.8, eDirectory does not listen on the localhost. Earlier the `ndssnmp.cfg` file had `SERVER localhost` set by default.

To resolve this error, you need to manually edit the `ndssnmp.cfg` file and include the host name of the eDirectory server, which needs to be monitored.

For example, type the following in the `ndssnmp.cfg` file:

```
SERVER test-server
```

test-server is the hostname on which eDirectory is running on the default NCP port (that is 524). If eDirectory is running on a different port (for ex: 1524), the entry should be as follows:

```
SERVER test-server:1524
```

6.7.2 Errors While Starting the NDS Subagent

The subagent can fail with the following message:

```
Unable to load library: libnetsnmp.so
```

To resolve this, export the environment variable SNMP_MAJOR_VERSION with the net-snmp library’s (`libnetsnmp.so`) major version number. For example, you might use the following command:

```
export SNMP_MAJOR_VERSION=10
```

6.7.3 Restarting ndssnmpsa

When the master agent is restarted on Linux, ndssnmpsa needs to be restarted.

To restart ndssnmpsa, stop ndssnmpsa and then start it again.

To stop ndssnmpsa, enter the following command:

```
/etc/init.d/ndssnmpsa stop
```

To start ndssnmpsa, enter the following:

```
/etc/init.d/ndssnmpsa start
```

6.7.4 Errors While Starting ndssnmpsa

When you start ndssnmpsa on Linux, you might get the following errors:

```
Error: eDirectory SNMP Initialization component. Error code: -168
```

```
Error: eDirectory SNMP Initialization component. Error code: 9
```

To resolve this, unload and load ndssnmp using the following commands:

```
/opt/novell/eDirectory/bin/ndssnmp -u
```

```
/opt/novell/eDirectory/bin/ndssnmp -l
```

6.7.5 Errors While Stopping ndssnmpsa

When ndssnmpsa is stopped on SLES 9, an error message similar to `"*** glibc detected *** double free or corruption (!prev): 0x0819cdd0 *** "` is displayed on the screen.

You can ignore these messages.

6.7.6 Compiling edir.mib

The eDirectory MIB file (`<eDirectoryInstallRootDir>\snmp\edir.mib`) on Windows compiles with some errors and warnings on HP OpenView. You can ignore these errors.

6.7.7 Modifying the SNMP Configuration File

If LDAP is not configured to run in clear text mode, the name of the trusted root certificate file must be given in the SNMP configuration file (for example, SSLKEY `C:\Novell\nds\trust.der`) before bringing up the eDirectory SNMP subagent.

`ndssnmp.cfg` is found in `C:\novell\nds\snmp` on Windows.

6.7.8 Using SNMP After a New Tree Installation

When you install eDirectory 8.8 SP8 for the first time (creating a new tree), if the Windows SNMP Service is installed on the server, and the SNMP Service has one or more dependent services, eDirectory cannot shut down the SNMP Service. If this happens, SNMP is not ready to use after the eDirectory installation.

Follow these steps to restart the SNMP service:

1. Click *Start* > *Settings* > *Control Panel* > *Administrative Tools* > *Services*.
2. Right-click *SNMP Service* in the *Name* list, then click *Stop*.
3. Click *Yes to All*.
4. Right-click *SNMP Service* in the *Name* list, then click *Start*.

6.7.9 SNMP Object Creation Error on Windows Server

While installing eDirectory on any supported Windows platform server, if you get an SNMP group object creation error, you need to manually create the SNMP group object. For information on the steps to manually create an SNMP object, refer to the "eDirectory and SNMP" (http://www.netiq.com/documentation/edir88/edir88/data/ag7hr1h.html) section of the *NetIQ eDirectory 8.8 SP8 Administration Guide*.

6.7.10 Uninstalling SNMP with eDirectory Uninstallation

If the Windows SNMP Service is installed on a server, and the SNMP Service has one or more dependent services, the eDirectory uninstall does not delete all the SNMP files in the `C:\novell\nds` folder. However, the other uninstallation processes complete successfully, including the deletion of the SNMP registry entries, and the deconfiguration process that the NetIQ SNMP agent does with DS and the SNMP Service.

To complete the uninstallation:

1. Click *Start* > *Settings* > *Control Panel* > *Administrative Tools* > *Services*.
2. Right-click *SNMP Service* in the *Name* list, then click *Stop*.
3. Click *Yes to All*.
4. Right-click *SNMP Service* in the *Name* list, then click *Start*.
5. Manually delete the remaining SNMP files in the `C:\novell\nds` folder.

7 iMonitor

- Section 7.1, “Browsing for Objects Containing Double-Byte Characters in iMonitor,” on page 51
- Section 7.2, “Agent Health Check on a Single-Server Tree,” on page 51
- Section 7.3, “iMonitor Report Does Not Save the Records for Each Hour,” on page 52
- Section 7.4, “Creation and Modification Time Stamps,” on page 52
- Section 7.5, “iMonitor Issues in Older Versions of Mozilla,” on page 52
- Section 7.6, “Run Report Screen Layout Not Aligned on iMonitor,” on page 52
- Section 7.7, “iMonitor Displays Error -672,” on page 52
- Section 7.8, “Time Stamps Displayed in Hexadecimal Format,” on page 52
- Section 7.9, “Issue with iMonitor Trace Configuration in Internet Explorer 10,” on page 53

7.1 Browsing for Objects Containing Double-Byte Characters in iMonitor

When using iMonitor to browse an eDirectory tree for objects, an object with double-byte characters in the name might not correctly hyperlink to the object properties.

7.2 Agent Health Check on a Single-Server Tree

The Agent Health check feature in iMonitor shows a Warning icon in the Results column when run on a single server tree because of the Perishable Data status. This does not mean that the tree is not healthy or that the Agent Health check is not working as designed. Perishable Data indicates the amount of data that has not yet been synchronized to at least one replica. A single server tree, by its nature, means that the data is always at risk for catastrophic failure because there is no other place that the data is replicated. If you lose the hard disk, you lose the data.

If you don't want to view health check warnings about Perishable Data or Readable Replica Counts on your single server tree, you can turn off these health checks by editing the `ndsimonhealth.ini` file to change the following entries:

```
perishable_data-active: OFF
```

and

`ring_readable-Min_Marginal: 1` or `ring_readable-active: OFF`

This turns off the warnings for Readable Replica Count and Perishable Data.

7.3 iMonitor Report Does Not Save the Records for Each Hour

The custom reports feature in iMonitor is designed to place the URL specified by the user into the saved report (the saved HTML file) when the custom report is created. That means that when you open a saved custom report that has been run, you see the live (current) data instead of the data captured by the URL at the time the custom report is run. This issue will be resolved in a future release of iMonitor.

7.4 Creation and Modification Time Stamps

Because Linux platforms do not maintain the creation time of a file, iMonitor shows both the creation and modification times to be the same.

7.5 iMonitor Issues in Older Versions of Mozilla

If you access iMonitor using Mozilla versions earlier than 1.5, iMonitor might have problems during DSTrace Flag selection. Mozilla might not support all the operations.

7.6 Run Report Screen Layout Not Aligned on iMonitor

The navigation and assistant frames appear twice on Linux.

To work around this problem, refresh the page.

7.7 iMonitor Displays Error -672

A few operations fail and display error -672, when some debug tool is running in parallel with iMonitor.

On Linux

iMonitor displays error -672 if dsdump tool is running in parallel with iMonitor.

To resolve this issue, exit the dsdump tool, before starting iMonitor.

On Windows

iMonitor displays error -672 if the dsbrowse or dsedit tool is running in parallel with iMonitor.

To resolve this issue, exit the dsbrowse and dsedit tool before starting iMonitor.

7.8 Time Stamps Displayed in Hexadecimal Format

If you set a Time syntax attribute with a value prior to January 1, 1970, iMonitor displays the time stamp for the attribute in hexadecimal format instead of in standard date/time format. iMonitor displays all attributes with values after January 1, 1970, in date/time format.

7.9 Issue with iMonitor Trace Configuration in Internet Explorer 10

Trace configuration in iMonitor does not work in Internet Explorer 10.

To work around this issue, launch Internet Explorer 10 in compatibility mode and iMonitor address to the list of Trusted Sites, then restart the browser.

8 iManager

- Section 8.1, “LDAP Operations Fail After Creating a New LDAP Group Using Quick Create,” on page 55
- Section 8.2, “Issues While Backing Up on Red Hat EUC in the Japanese Locale,” on page 55

8.1 LDAP Operations Fail After Creating a New LDAP Group Using Quick Create

Quick Create only creates an LDAP group object with dummy attributes that you can later modify. It creates the LDAP Group object with version one instead of nine. Therefore, all the LDAP operations fail as it is not possible to associate any LDAP server due to version incompatibility.

To work around this issue, after creating the LDAP group using Quick Create, change the LDAP Group object version number to nine.

8.2 Issues While Backing Up on Red Hat EUC in the Japanese Locale

You might have problems while backing up by using iManager on Red Hat EUC in the Japanese locale. The fix for this issue will be available with iManager 2.6.

9 Obituaries

Obituaries serve as operational attributes that eDirectory places on objects to ensure referential integrity during operations such as delete, move, rename, and restore. For example, if Group A has a member, User B, and User B is deleted, the directory automatically removes the reference to User B from Group A. In eDirectory 8.8 SP 8, the obituaries generated by the Delete, Move, and Rename operations are optimized by default.

NOTE: Objects with obituaries are considered every time an agent outbound synchronizes, and by the obituary process, which is scheduled to run at the end of an inbound synchronization cycle.

There are three general classifications for obituaries:

- Primary obituaries include the types Dead (0001), Restored (0000), Moved (0002), New RDN (0005), and Tree New RDN (0008).
- Secondary obituaries are generally associated with a Primary obituary and represent the agents and partitions that need to be notified of the operation specified in the Primary obituary. They include the types Back Link (0006), Used By (000C), and Move Tree (000a).
- Tracking obituaries include the types Inhibit Move (0003), Old RDN (0004), and Tree Old RDN (0007).

Obituaries, with the exception of Tracking obituaries, must move through a set of synchronizing states:

- Initial State or Issued (0)
- Notified (1)
- OK to Purge (2)
- Purgeable (4)

The states are recorded in the Flags field in the obituary attribute. Before an obituary can move to the next state, the current state must have been synchronized to all replicas of the real object. In order to determine whether all replicas in the ring have seen a given obituary state, a vector is computed from the transitive vector. In eDirectory 8.6 and later, a non-stored Obituary Vector is used. In previous versions of eDirectory, the Purge Vector is used. If the Modification Timestamp (MTS) on the obituary is older than the computed vector, the server responsible for that obituary can advance it to the next state.

For a Secondary obituary of type Back Link, the agent that holds the master replica of the object with the obituary is responsible for advancing the states. For a Secondary obituary of type Used By, the replica agent that created it is responsible for advancing the obituary states as long as that replica still exists. If it does not still exist, the agent holding the master of that partition takes over advancing the obituary states for the Used By obituary. For a Move Tree obituary, the master of the root partition is responsible for advancing the states.

Primary obituaries can be advanced in their states only after all Secondary obituaries have advanced through all of their states. After the Primary obituary reaches its last state, and that state synchronizes to all servers in the ring, all that remains is the object husk, which is an object without attributes—one

which can subsequently be purged from the system by the Purge Process. Tracking obituaries are removed after the Primary obituary is ready to be removed or, in the case of `Inhibit_move`, the Tracking obituary is removed after the Primary obituary has moved to the `OBF_NOTIFIED` state on the master replica.

The replica responsible for processing obituaries does so on a background process (the Obituary Process), which is scheduled on a per-partition basis after a given partition finishes an inbound synchronization cycle. If there are no other replicas of the partition, the Outbound Replication Process is still scheduled on the heartbeat interval. The Outbound Replication Process then starts the Obituary Process. The Obituary Process cannot be manually scheduled, nor does it need to be. As synchronization occurs, the transitive vectors are updated, thus advancing the Purge Vector and Obit Vector. As these vectors move forward, the obituary states are allowed to move forward. This, together with the automatic scheduling done upon inbound synchronization, completes the obituary processing cycle. Therefore, the lifeblood of obituary processing is object synchronization.

For an object that is being removed, after all obituaries whose associated Primary obituary is of type Dead have been advanced to the last state (Purgeable), and that state has been synchronized to all replicas, a new process is responsible for removing the remaining entry husk from the database. The Purge Process runs automatically to remove these husks. You can manually schedule the Purge Process and modify its automatic schedule interval by using the Agent Configuration page in iMonitor.

9.1 Examples

This section contains the following examples:

- "Deleting an Object" on page 58
- "Moving an Object" on page 59

9.1.1 Deleting an Object

1. Add the Primary obituary OBT_DEAD.

   The Back Link attribute contains a list of servers that have an interest in this object and need to be notified of changes to this entry. For every DN listed in the Back Link attribute and all servers listed in the entry's partition replica attribute, eDirectory adds a Back Link obituary. The creation time of the Primary obituary, OBT_DEAD, is stored in the Secondary obituary.

   The Used By attribute contains a list of partitions that have an interest in this object and need to be notified of changes to this entry. For every DN listed in the Used By attribute, eDirectory adds a Used By obituary. The creation time of the Primary obituary, OBT_DEAD, is stored in the Secondary obituary.

2. Remove all attributes but the obituaries.

   The Outbound Replication Process then synchronizes this change to all other servers in the replica ring.

   On the next inbound synchronization of this partition, the Obituary Process is started, which does the following:

   - Computes a time vector which is a minimum transitive vector, referred to as the purge vector. Later versions of eDirectory compute a second minimum vector, called the obituary vector, which does not consider replicas which are subordinate references.
   - Each Obituary in this partition is now examined.

If the obituary is a Primary obituary, there are no Secondary obituaries, and the attribute's modification time (MTS) on the obituary is older than the Purge Vector, then all servers have seen the change and this obituary will be removed.

If the obituary is a Back Link obituary and this server is the master, then this server is responsible for processing this obituary.

IMPORTANT: Perform the required operation for this state if it has not been done. Most often, this is done by notifying an external reference.

If the obituary is a Used By obituary and this server is the server where the delete occurred (determined by comparing the replica number in the obituary's MTS to our replica number), this server is responsible for processing this obituary.

- If this server is responsible for processing a particular Secondary obituary type (Back Link or Used By), all Secondary obituaries of that type on an entry are in the same state, the required operation for that state has been completed on all obituaries (for example, servers have been notified), and the obituary's MTSs for that obituary type are older than the Obituary Vector, then all Secondary obituaries of that type can be advanced to the next state.

9.1.2 Moving an Object

Move acts much like Delete, but with the following changes:

- Before the Primary obituary is placed on the move source, a partial entry is created in the destination container and a Tracking obituary (OBT_INHIBIT_MOVE) is placed on that partial entry. This Tracking obituary is placed to prevent the entry from being moved or taking part in a partition operation before the full entry is transferred from the source.
- On the source entry, the Primary obituary is OBT_MOVED.
- After the Primary obituary (OBT_MOVED) is moved to the Notified state (meaning that all replicas of the source know the entry is being moved) and all external references have been notified, the Tracking obituary (OBT_INHIBIT_MOVE) is removed from the destination entry.

9.2 Prevention

On a regular basis, run the iMonitor Server Information report. This report walks the entire tree, communicates with every NCP server it can find, and reports any errors it finds. You can use this report to diagnose time synchronization and limber problems, or to find out if the current server is able to communicate with all other servers from this server's perspective. If selected in the configuration page, the server can also generate NDS Agent Health information for every server in the tree. See "Configuring and Viewing Reports" in the *NetIQ eDirectory 8.8 SP8 Administration Guide* for more information on running the Server Information report.

If you are using iMonitor 2.0 or later, make sure that the Errors and Health Sub-report report options are enabled. The following items will be verified. You should browse the report and make sure that there are no errors.

- Based on the information in the `ndsimonhealth` configuration file stored with iMonitor (see "Configuration Files" in the *NetIQ eDirectory 8.8 SP8 Administration Guide*), this report will check the eDirectory agent version to ensure you are running the correct directory patches tree-wide.
- All servers are within Timesync tolerances.
- This server can communicate with all other servers.

- There have not been any servers improperly or incompletely removed from the tree.
- The Health subreport will indicate if any partitions are not within tolerance for the replication sync times.

If you are using iMonitor 1.5, select the Errors report option. The following items will be verified. You should browse the report and make sure that there are no errors.

- The agent version is displayed. Make sure all servers tree-wide are running the most current eDirectory Support Pack available from the NetIQ Support Web site (http://support.novell.com).
- All servers are within Timesync tolerances.
- This server can communicate with all other servers.
- There have not been any servers improperly or incompletely removed from the tree.

Using the iMonitor Obituary Listing report or the iMonitor Object Statistics report, you can find any obituaries on your system. If you find any obituaries that you don't believe are being processed, see Section 9.3, "Troubleshooting Tips," on page 60.

9.3 Troubleshooting Tips

There are two general reasons that obituaries don't process: either the obituary has been orphaned (that is, the obituary exists on some servers but not all servers) or the obituary is stuck (that is, it exists on all servers but its states are not advancing for some reason).

Do the following to troubleshoot orphaned or stuck obituaries:

- ❒ Don't panic!
- ❒ If the obituary is for an object not stored on this server (that is, the object is an External Reference):
  - Check to see if the real object has a matching obituary. If not, this obituary has been orphaned. See "Resolving Orphaned Obituaries on Extrefs" on page 61 for more information.
  - If the real object has a matching obituary, troubleshoot and resolve obituary problems on the real object before attempting to address any issues with the obit on the `ExtRef` partition.
- ❒ Make sure that the obituaries are correctly synchronized.
  - Use the iMonitor Agent Synchronization page to check for and resolve any synchronization errors.
  - Obituaries can change states only after all agents holding a copy of the replica ring have seen the state change. There are several ways to ensure that every replica has seen the data:

    While browsing the entry with obituaries, click the Entry Synchronization link. The page displayed will show all attributes that have not been synchronized to all replicas.

    Find the oldest time stamp on any of the obituary attribute values. The difference between that time and the current time should be greater than the interval shown in the Max Ring Delta field on the Partition Synchronization page.

    Evaluate the transitive vector.
- ❒ Run the iMonitor Server Information Report to ensure that all server communication is functioning.
- ❒ Examine the Agent Process Status: Obituaries to look for any errors.
  - Common problems in Agent Process Status: Obituaries include

-625, -622, -634, and -635 communication problems. See Server Information Report for more details.

-601, and -603, indicating servers that have been improperly removed, or that the Server object might have a base class of Unknown.

- Errors shown on this page are not fatal. The next time the obituary process runs for that partition, it will retry the operation. Resolve any issues shown in this page, then wait for the retry.

❒ While looking at obituary objects, walk around the replica ring, comparing the obituary around the ring.

- If not all replicas have a copy of the obituary and all attribute values are not purgeable, this object is inconsistent around the replica ring—and this is a case of an orphaned obituary. See "Resolving Orphaned Obituaries" on page 61 for more information.
- If the object exists on all replicas and is consistent, then it might not be advancing because of synchronization errors, or the obituary process might be getting errors.

❒ As needed, use Trace with the Obituary option enabled to examine the obituary process in detail.

❒ To prevent obituary problems in the future, upgrade to the latest Support Pack (for eDirectory 8.6 servers). There have been fixes for all known obituary issues.

9.3.1 Solutions

Use the proper solution referred to in Section 9.3, "Troubleshooting Tips," on page 60.

Before using any of these solutions, you must make sure that your data is safe. You might need to back up the directory database files, server configuration, and trustees. To increase the probability of success and to minimize future problems, upgrade to the latest eDirectory Support Packs.

Resolving Orphaned Obituaries

- **Preferred method:** If eDirectory 8.6 or later is on any of the servers in the replica ring, browse to the object in iMonitor, then select Send Single Entry. This will perform a nonauthoritative send to all other replicas.
- **Far less desirable method:** If all servers in the replica ring that have a copy of the orphaned obituary are older than eDirectory 8.6, load DSBrowse with the -a option, browse to the object, then time-stamp the entry. This will make the object as it exists on this server the authoritative copy. We do not recommend making objects authoritative as a matter of practice.

Resolving Orphaned Obituaries on Extrefs

- **Less desirable method:** Run DSRepair with the time stamp option selected.
- **Less desirable method:** Move a real replica to the server, wait for it to turn on, then wait for the obituary to be processed. If the obituary is not processed, use the information in Section 9.3, "Troubleshooting Tips," on page 60 to resolve the issue now that the object is on a real replica. After the obituary has processed, the replica can be removed if desired.

9.3.2 Previous Practices

In the past, several different strategies have been employed to resolve stuck obituaries. Some of these strategies involve expensive partitioning operations, or the use of undocumented features that might cause problems in the future.

The first strategy was to switch which replica held the master. This would work in some cases because the master is the agent responsible for moving the Back Link obituaries through their various states. In the case where the replica was inconsistent and the master didn't hold the deleted object, switching masters to an agent that held the deleted entry with its obituaries would give the new agent the license to push the obituaries through their states and eventually purge it out. Send Single Entry is a much cleaner and less dangerous way to resolve obituaries that are stuck because the replica is inconsistent.

The second strategy used was to run DSRepair with certain switches to delete all obituaries. (There is a third-party application which resolves stuck obituaries by launching DSRepair.) We do not recommend this strategy. Using those switches will delete all obituaries on this agent, which means that obituaries that are not stuck might also be removed, creating further replica inconsistencies and more stuck obituaries. Because this is not a distributed operation, you must run DSRepair on all of the servers with stuck obituaries, which increases the odds that one of those servers has obituaries for another partition which will be prematurely deleted. The premature deletion of obituaries can cause additional orphaned obituaries and, in turn, cause problems which can be found years later when you change replicas types, add new replicas, or perform other partitioning operations.

The third strategy used was to make objects authoritative, either using DSBrowse with the advanced mode operation and time stamping the entry, or running DSRepair with the -0T switch. This forces the entry to become authoritative and synchronize out to all other replicas. This should be done with great care because you might lose data changed on other servers. We recommend that this be a rarely employed method of obituary cleanup.

10 Migrating to NetIQ eDirectory

This chapter explains the process to migrate to NetIQ eDirectory from:

- Section 10.1, "Migrating the Sun ONE Schema to NetIQ eDirectory," on page 63
- Section 10.2, "Migrating the Active Directory Schema to NetIQ eDirectory Using ICE," on page 66

10.1 Migrating the Sun ONE Schema to NetIQ eDirectory

To migrate the Sun ONE schema to NetIQ eDirectory, complete the following steps:

"Step 1: Perform the Schema Cache Update Operation" on page 63
"Step 2: Rectify the Error LDIF File to Eliminate the Errors" on page 63
"Step 3: Import the LDIF File" on page 65

10.1.1 Step 1: Perform the Schema Cache Update Operation

You can write the errors encountered while comparing the schema to an error file using the following command:

```
ice -e LDIF error file name -C -a -SLDAP -s Sun ONE server -p Sun ONE port -DLDAP -
s eDirectory server -p eDirectory port
```

For example:

```
ice -e err.ldf -C -a -SLDAP -s sun_srv1 -p sun_port1 -DLDAP -s edir_srv2 -p
edir_port2
```

Any errors encountered while comparing the schema is written to the error file (`err.ldf` in the example). You do not need to login to perform this operation unless one of the servers require authentication in order to read the Root DSE. Microsoft Active Directory requires authentication to read the Root DSE.

10.1.2 Step 2: Rectify the Error LDIF File to Eliminate the Errors

- Sun ONE defines some schema definitions publicly that eDirectory does not. This includes attributes like `objectClasses`, `attributeTypes`, `ldapSyntaxes`, and `subschemSubentry`. These definitions exist internally and are very important to the schema, and therefore, they cannot be modified. Operations that try to modify these definitions results in the following error:

  ```
  LDAP error : 53 (DSA is unwilling to perform)
  ```

  Any records that contain references to these definitions cause the following error:

  ```
  LDAP error : 16 : ( No such attribute )
  ```

Thus, records that contain any reference to these objects or that try to modify these definitions need to be commented in the LDIF error file (`err.ldf` in the example).

- Some objectClasses definitions in Sun ONE do not have naming attributes. Adding these objectClasses would result in the following error in eDirectory:

  ```
  LDAP error : 80 (NDS error: ambiguous naming (-651)
  ```

  This error occurs because Sun ONE does not use the same method for determining naming rules as eDirectory.

  To solve this, you can use any *one* of the three following options:

  Option 1:

  Go through each of the offending objectClasses and add a valid naming attribute to each of them.

  For example:

  To add the naming attribute `[ cn ]` to the objectClass `netscapeMachineData` modify the entry (that is *emphasized* in the example below) in the `err.ldf` file to include the `X-NDS_NAMING` flag as shown below:

  ```
  dn: cn=schemachangetype: modifyadd: objectClassesobjectClasses: (
  2.16.840.1.113730.3.2.32 NAME 'netscapeMachineData'
      DESC 'iPlanet defined objectclass' SUP top STRUCTURAL MAY 'cn'     X-
  NDS_NAMING 'cn' )-
  ```

  Option 2:

  Go through each of the offending objectClasses and make them AUXILIARY or ABSTRACT.

  For example:

  To modify the definition of objectClass `netscapeMachineData` from STRUCTURAL to AUXILIARY, modify the `err.ldf` file entry (that is *emphasized* in the example below) as shown below:

  ```
  dn: cn=schemachangetype: modifyadd: objectClassesobjectClasses: (
  2.16.840.1.113730.3.2.32 NAME 'netscapeMachineData'
     DESC 'iPlanet defined objectclass' SUP top AUXILIARY )-
  ```

  To modify the definition of objectClass `netscapeMachineData` from STRUCTURAL to ABSTRACT, modify the `err.ldf` file entry (that is *emphasized* in the example below) as shown below:

  ```
  dn: cn=schemachangetype: modifyadd: objectClassesobjectClasses: (
  2.16.840.1.113730.3.2.32 NAME 'netscapeMachineData'
     DESC 'iPlanet defined objectclass' SUP top ABSTRACT )-
  ```

  Option 3:

  Add `cn` to the definition of `Top` in eDirectory, which causes a potential naming attribute for all objectClasses.

  There are two ways of adding `cn` to `Top`:

  - **Method 1:**

    Create a file as shown below and name it `topsch.ldf`.

    ```
    version : 1

    dn:cn=schema

    changetype :modify

    delete : objectclasses

    objectclasses : ( 2.5.6.0 NAME 'top' STRUCTURAL )
    ```

```
-

add:objectclasses

objectclasses : (2.5.6.0 NAME 'top' STRUCTURAL MAY cn)
```

Use the following NetIQ Import Conversion Export command line:

```
ice -SLDIF -f LDIF_file_name -DLDAP -s eDirectory_server -p eDirectory_port
-d eDirectory_Admin_DN -w eDirectory_password
```

For example:

```
ice -SLDIF -f topsch.ldf -DLDAP -s edir_srv2 -p edir_port2 -d
cn=admin,o=org -w pwd1
```

- **Method 2:**

  1. In NetIQ iManager, click the *Roles and Tasks* button  .
  2. Click *Schema* > *Add Attribute*.
  3. In the *Available Classes* list, select *Top*, then click *OK*.
  4. Double-click *CN* in the *Available Optional Attributes* list.
  5. Click *OK*.

- Some objectClass definitions contain `userPassword` as part of their mandatory attributes list. Adding such objectClasses to eDirectory cause the following error:

  ```
  LDAP error : 16 (No such attribute)
  ```

  To resolve this error, modify the objectClass definition to inherit the new objectClass from `ndsLoginProperties` and remove the `userPassword` attribute from the mandatory attribute list.

  For example:

  An objectClass containing `userPassword` in the mandatory attributes list:

  ```
  version : 1
  dn: cn=schemaz
  changetype: modify
  add: objectClasses
  objectClasses: ( 0.9.2342.19200300.100.4.19 NAME 'simpleSecurityObject' DESC '
   Standard LDAP objectClass' SUP top STRUCTURAL MUST userPassword )
  ```

  Needs to be modified as following (notice the change to the last line):

  ```
  version : 1
  dn: cn=schema
  changetype: modify
  add: objectClasses
  objectClasses: ( 0.9.2342.19200300.100.4.19 NAME 'simpleSecurityObject' DESC '
   Standard LDAP objectClass' SUP (ndsLoginProperties $ top) STRUCTURAL )
  ```

10.1.3 Step 3: Import the LDIF File

Use the following NetIQ Import Conversion Export command to import the modified schema compare LDIF file (`err.ldf` in our example):

```
ice -e error_file -SLDIF -f modified_LDIF_file -DLDAP -s eDirectory_server -p
eDirectory_port -d eDirectory_Admin_DN -w eDirectory_password
```

For example:

```
ice -e errors.ldf -SLDIF -f err.ldf -DLDAP -s edir_srv2 -p edir_port2 -d
cn=admin,o=org -w pwd1
```

10.2 Migrating the Active Directory Schema to NetIQ eDirectory Using ICE

While migrating schema from Active Directory to NetIQ eDirectory using ICE, schema migration for the `Computer` objectClass fails with an `ambiguous naming error (-651)` error.

To resolve this, complete the following steps:

"Step 1: Perform the Schema Cache Update Operation" on page 63
"Step 2: Rectify the Error LDIF File to Eliminate the Errors" on page 63
"Step 3: Import the LDIF File" on page 65

10.2.1 Step 1: Perform the Schema Cache Update Operation

While migrating schema from Active Directory to NetIQ eDirectory using ICE, ensure that you have provided the error log option (`-e`) of ICE as follows:

```
ice -e error_file -S ldap -s Active_Directory_server -p Active_Directory_port -d
Active_Directory_full_admin_context -w Active_Directory_password -D ldap -s
eDirectory_server -p eDirectory_port -d eDirectory_full_admin_context -w
eDirectory_password
```

For example:

```
ice -e err.ldf -S ldap -s activesrv1 -p activeport1 -d cn=admin,o=company -w
activepwd -D ldap -s edirsrv2 -p edirport2 -d cn=admin,o=company -w edirpwd
```

10.2.2 Step 2: Rectify the Error LDIF File to Eliminate the Errors

The failed entry would be present in the `err.ldf` file as shown below:

```
dn: cn=schema

changetype: modify

delete: objectclasses

objectclasses: ( 2.16.840.1.113719.1.1.6.1.4 NAME 'computer' )

-

add: objectclasses

objectclasses: ( 2.16.840.1.113719.1.1.6.1.4 NAME 'computer' SUP (device $
 user ) STRUCTURAL MAY (operator $ server $ status $ cn $ networkAddress $
 local PolicyFlags $ defaultLocalPolicyObject $ machineRole $ location $
 netbootInitialization $ netbootGUID $ netbootMachineFilePath $ siteGUID $
 operatingSystem $ operatingSystemVersion $ operatingSystemServicePack $
 operatingSystemHotfix $ volumeCount $ physicalLocationObject $ dNSHostName
 $ policyReplicationFlags $ managedBy $ rIDSetReferences $ catalogs $
 netbootSIFFile $ netboot MirrorDataFile ) X-NDS_NOT_CONTAINER '1' X
 -NDS_NONREMOVABLE '1' X-NDS_NAME 'Computer' )

-
```

Modify this entry in the error file (`err.ldf` in the example) to remove the `user` objectClass from the list of superior objectClasses in the definition of the `Computer` objectClass, as shown below:

```
dn: cn=schema

changetype: modify
```

```
delete: objectclasses

objectclasses: ( 2.16.840.1.113719.1.1.6.1.4 NAME 'computer' )

-

add: objectclasses

objectclasses: ( 2.16.840.1.113719.1.1.6.1.4 NAME 'computer' SUP device
 STRUCTURAL MAY (operator $ server $ status $ cn $ networkAddress $ local
 PolicyFlags $ defaultLocalPolicyObject $ machineRole $ location $
 netbootInitialization $ netbootGUID $ netbootMachineFilePath $ siteGUID $
 operatingSystem $ operatingSystemVersion $ operatingSystemServicePack $
 operatingSystemHotfix $ volumeCount $ physicalLocationObject $ dNSHostName
 $ policyReplicationFlags $ managedBy $ rIDSetReferences $ catalogs $
 netbootSIFFile $ netbootMirrorDataFile ) X-NDS_NOT_CONTAINER '1' X
 -NDS_NONREMOVABLE '1' X-NDS_NAME 'Computer' )

-
```

10.2.3 Step 3: Import the LDIF File

Now, import the modified entry using the following ICE command:

```
ice -S ldif -f LDIF_file -D ldap -s Novell_eDirectory_server -p port_number -d
full_admin_context -w password
```

For example:

```
ice -S ldif -f err.ldf -D ldap -s edirsrv1 -p edirport1 -d cn=admin,o=company -w
pwd1
```

10.3 Migrating from OpenLDAP to NetIQ eDirectory

- Section 10.3.1, "Prerequisites," on page 67
- Section 10.3.2, "Migrating the OpenLDAP Schema to eDirectory," on page 67
- Section 10.3.3, "Migrating the Open LDAP Data to NetIQ eDirectory," on page 68
- Section 10.3.4, "Making PAM Work with NetIQ eDirectory After Migration," on page 68

10.3.1 Prerequisites

The data that is migrated from an OpenLDAP server can have MD5 passwords, which may cause the applications to break if the appropriate NetIQ Modular Authentication Service (NMAS) methods are not installed. The NMAS method, SimplePassword, needs to be installed for the NetIQ eDirectory using the command as below:

```
nmasinst -addmethod admin_context treename configfile -h Hostname:port-w password
```

For example: `nmasinst -addmethod admin.novell eDir-Tree /Linux/eDirectory/nmas/NmasMethods/Novell/SimplePassword/config.txt -h eDir_srv:524 -w secret`

10.3.2 Migrating the OpenLDAP Schema to eDirectory

To migrate the OpenLDAP schema to eDirectory, complete the following steps:

- "Step 1: Perform the Schema Cache Update Operation" on page 68
- "Step 2: Rectify the Error LDIF File to Eliminate the Errors" on page 68

Step 1: Perform the Schema Cache Update Operation

You can write the errors encountered while comparing the schema to an error file using the following command:

```
ice -e error_file -C -a -S ldap -s OpenLDAP_server -p Open_LDAP_port - D ldap -s
eDirectory_server -p eDirectory_port -d eDirectory_full_admin_context -w
eDirectory_password
```

For example:

```
ice -e err.ldf -C -a -SLDAP -s open_srv1 -p open_port1 -DLDAP -s edir_srv2 -p
edir_port2 -d cn=admin,o=novell -w secret
```

Any errors encountered while comparing the schema is written to the error file (`err.ldf` in the example).

Step 2: Rectify the Error LDIF File to Eliminate the Errors

Open LDAP defines some schema definitions publicly, which include attributes like `objectClasses`, `attributeTypes`, `ldapSyntaxes`, and `subschemSubentry`. These definitions exist internally and are very important to the schema, and therefore, they cannot be modified. Operations that try to modify these definitions results in the following error:

```
LDAP error : 53 (DSA is unwilling to perform)
```

Any records that contain references to these definitions cause the following error:

```
LDAP error : 16 ( No such attribute )
```

Thus, records that contain any reference to these objects or that try to modify these definitions need to be commented in the LDIF error file (`err.ldf` in the example).

10.3.3 Migrating the Open LDAP Data to NetIQ eDirectory

Execute the following command to migrate the data:

```
ice -e error_data.ldif -SLDAP -s OpenLDAP_server -p OpenLDAP_port  -d admin_context
-w password -t -b dc=blr,dc=novell,dc=com -F objectclass=* -DLDAP -d  admin_context
-w password  -l  -F
```

For example:

```
ice -e err_data.ldif -SLDAP -s open_srv1 -p open_port1  -d
cn=administrator,dc=blr,dc=novell,dc=com -w secret1 -t -b dc=blr,dc=novell,dc=com
-F objectclass=* -DLDAP -d cn=admin,o=novell -w secret2 -l -F
```

Some objects also may fail due to forward referencing and internal dependencies on the objects, which may not break any applications.

10.3.4 Making PAM Work with NetIQ eDirectory After Migration

After migrating from OpenLDAP to eDirectory, you need to make some changes for PAM to work with eDirectory.

Changes in /etc/ldap.conf File

```
# The distinguished name to bind to the server with.
# Optional: default is to bind anonymously.
  binddn cn=admin,o=acme
  ...
# The credentials to bind with.
# Optional: default is no credential.
  bindpw secret
  ...
# The search scope.
  scope sub
  ...
# Filter to AND with uid=%s
  pam_filter objectclass=inetorgperson
  ...
# Remove old password first, then update in
# cleartext. Necessary for use with Novell
# Directory Services (NDS)
  pam_password nds
  ...
  ssl off
  ...
```

Changes to the Data in the Directory

This change is only specific to the scenario where the users objects in OpenLDAP have CRYPT as the password hash algorithm.

Using iManager, add the following attribute with the specified value to the container having all the user objects:

Attribute: `sasDefaultLoginSequence`

Value: Simple Password

11 Schema

This section includes information for troubleshooting schema.

Troubleshooting Schema

When an auxiliary class is disassociated with an object, the value is not immediately deleted, but it is marked as not present. The auxiliary class is associated with the entry until the DRL process cleans up these values during the actual object validation.

Because the DRL is a resource-consuming background process, other operations are slow during this cleanup. The duration of the cleanup process depends on the number of actual objects and external references in the system. Because this process is CPU and memory intensive, you should not run it frequently. By default, the Backlinker background process runs 50 minutes after ndsd is started and then subsequently runs every 13 hours.

Clearing an auxiliary class from an entry can take between 0 and 13 hours, plus the time taken to process this entry in the system.

To work around this issue, delete the entry of the auxiliary class by triggering the Backlinker through DSTrace or iMonitor.

NOTE: When the object is deleted, the values are immediately purged because this deletion is handled by other background processes.

---

12 DSRepair

- Section 12.1, “Running DSRepair on an NFS Mounted DIB on Linux,” on page 73
- Section 12.2, “Running DSRepair with -R Option Hangs,” on page 73
- Section 12.3, “Running DSRepair after Upgrade or Migration,” on page 73

12.1 Running DSRepair on an NFS Mounted DIB on Linux

You might get -732 or -6009 errors while trying to run the ndsrepair (DSRepair) operations on an NFS-mounted DIB on Linux systems.

12.2 Running DSRepair with -R Option Hangs

After enabling encrypted attributes on indexed attributes, if you run ndsrepair (DSRepair) with the `-R` option, it hangs.

12.3 Running DSRepair after Upgrade or Migration

If you run DSRepair unattended after an upgrade or migration from 8.7.3.x server, an `Invalid Ancestor ID list for the entry` error message appears.

This can be ignored, because an Ancestor ID upgrade is done as part of the background process, after the DIB upgrade or migration process finishes.

13 Replication

eDirectory offers the NetIQ robust directory service and the fault tolerance inherent in replication. Replication allows you to keep copies of the eDirectory database, or portions of it, on multiple servers at once.

- Section 13.1, "Encrypted Replication Issues," on page 75
- Section 13.2, "Recovering from eDirectory Replica Problems," on page 75

13.1 Encrypted Replication Issues

- Section 13.1.1, "Configuring Encrypted Replication Through iManager," on page 75
- Section 13.1.2, "Merging Trees With Encrypted Replication Enabled Fails," on page 75

13.1.1 Configuring Encrypted Replication Through iManager

You cannot configure encrypted replication through iManager if any server in the replica ring is down.

13.1.2 Merging Trees With Encrypted Replication Enabled Fails

When encrypted replication is enabled, merging trees fails. Disable secure replication on each tree before doing a merge.

13.2 Recovering from eDirectory Replica Problems

You should always keep multiple replicas of eDirectory partitions. If you do so and one replica becomes corrupted or is lost because of a failed hard disk, you can delete that replica using NetIQ iManager and replace it with a new one from the intact replica.

For more information on deleting replicas, see "Administering Replicas" (http://www.novell.com/documentation/edir88/edir88/data/fbgciaad.html) in the *NetIQ eDirectory 8.8 SP8 Administration Guide*.

14 Clone DIB Issues

- Section 14.1, “Clone DIB Fails With -601 and -603 Errors,” on page 77
- Section 14.2, “Clone DIB Can Fail Immediately After Offline Bulkload,” on page 77
- Section 14.3, “Issue in Cloning with Enabled Encrypted Replication Feature,” on page 77

14.1 Clone DIB Fails With -601 and -603 Errors

When encrypted attributes and encrypted replication is enabled at the tree level, clone DIB fails with the following errors:

- Clone DIB on target server fails with the -601 error while configuring SAS
- After Clone DIB, the newly created clone object fails with the -603 error

To work around these issues, disable encrypted attributes and encrypted replication.

14.2 Clone DIB Can Fail Immediately After Offline Bulkload

If you try taking the clone of a server immediately after an offline bulkload, it might result in a failure, if the bulkload has been done with the disable indices option.

However, this is not an issue if the dibclone is initiated a few hours after the bulkload completion.

14.3 Issue in Cloning with Enabled Encrypted Replication Feature

While cloning with the Encrypted Replication feature enabled on the source server, modify the ER policy to temporarily exclude the cloned server. This can be changed after the configuration of the cloned server is complete.

15 NetIQ Public Key Infrastructure Services

- Section 15.1, “PKI Operations Not Working,” on page 79
- Section 15.2, “LDAP Search from Netscape Address Book Fails,” on page 79
- Section 15.3, “Removing the configuration of an eDirectory server that is acting as a treekey server in a multiserver tree after having moved the existing eDirectory objects to a different server fails with the error code for Crucial Replica.,” on page 80
- Section 15.4, “While uninstalling the eDirectory Server holding the CA, the KMOs created on that server will be moved to another server in the tree and become invalid,” on page 80
- Section 15.5, “PKI Plugin Encounters Error When Installed in iManager 2.7.6 Patch1 and Lower Versions,” on page 80

15.1 PKI Operations Not Working

If PKI operations in iManager are not working, it could be because NetIQ PKI Services are not running on the Linux. Start the PKI Services by entering `npki -1`.

If you cannot create certificates, you need to ensure that the NICI module has been properly installed. See “Initializing the NICI Module on the Server” in the *NetIQ eDirectory 8.8 SP8 Administration Guide*. To verify if NICI is initialized, see “Verifying Whether NICI Is Installed and Initialized on the Server” in the *NetIQ eDirectory 8.8 SP8 Administration Guide*.

15.2 LDAP Search from Netscape Address Book Fails

If you are using an export version of the Netscape browser and a Key Material Object (KMO) key size larger than 512 bits associated with the LDAP Server object, the LDAP search from the Netscape Address Book might fail.

Use a domestic version of the Netscape browser in such cases.

15.3 Removing the configuration of an eDirectory server that is acting as a treekey server in a multiserver tree after having moved the existing eDirectory objects to a different server fails with the error code for Crucial Replica.

To complete the operation, change the Key Server DN attribute in the W0 object under Security Container > KAP to another server in the tree that has downloaded the treekey from this server.

1. In NetIQ iManager, click the *Roles and Tasks* button  .
2. Click *eDirectory Administration* > *Modify Object*.
3. Specify the name and context of the W0 object (usually W0.KAP.Security), then click *OK*.
4. In the *Valued Attributes* column, select *NDSPKI:SD Key Server DN*, then click *Edit*.
5. Specify the name and context of a different server in the *Security Domain Key Server's DN* field, then click *OK*.
6. Click *Apply*, then click *OK*.

15.4 While uninstalling the eDirectory Server holding the CA, the KMOs created on that server will be moved to another server in the tree and become invalid

You should re-create the CA and KMOs for the tree. See "Creating an Organizational Certificate Authority Object" and "Creating a Server Certificate Object" in the *NetIQ eDirectory 8.8 SP8 Administration Guide* for more information.

We recommend that you do not uninstall the eDirectory server where the CA for the tree has been created.

15.5 PKI Plugin Encounters Error When Installed in iManager 2.7.6 Patch1 and Lower Versions

To work around this issue, create a `libntls.so.8` symbolic link pointing to `libntls.so`, as follows:

```
ln -sf /var/opt/novell/iManager/nps/WEB-INF/bin/linux/libntls.so
/var/opt/novell/iManager/nps/WEB-INF/bin/linux/libntls.so.8
```

16 Troubleshooting Utilities on Linux

- Section 16.1, “NetIQ Import Convert Export Utility,” on page 81
- Section 16.2, “ndsconfig Utility,” on page 81
- Section 16.3, “ndsmerge Utility,” on page 82
- Section 16.4, “DSTrace Utility,” on page 82
- Section 16.5, “ndsbackup Utility,” on page 82
- Section 16.6, “Using DSRepair,” on page 82
- Section 16.7, “Using DSTrace,” on page 89

16.1 NetIQ Import Convert Export Utility

If an LDAP server is refreshed or unloaded, while a NetIQ Import Conversion Export operation is running, the `LBURP operation is timed out` message is displayed on the screen. The server recovers later, when the LBURP operation times out.

16.2 ndsconfig Utility

This section consists of the following:

- Section 16.2.1, “Configuring ndsconfig to Run From Non-Default Location,” on page 81
- Section 16.2.2, “ndsconfig Does Not Verify an Invalid Configuration File Path,” on page 81
- Section 16.2.3, “ndsconfig get Outputs Junk Characters for Non-English Characters,” on page 82

16.2.1 Configuring ndsconfig to Run From Non-Default Location

If you receive an error when running the ndsconfig utility from a location other than the default `/opt/novell/eDirectory/bin` directory, ensure that you export the ndspath before running ndsconfig. Use the following command:

```
source /opt/novell/eDirectory/bin/ndspath
```

After you export the command, enter `ndsconfig` to run the ndsconfig utility, instead of `./ndsconfig`.

16.2.2 ndsconfig Does Not Verify an Invalid Configuration File Path

To create the necessary configuration file, ndsconfig requires the full path and the configuration filename. When the same path name is passed for both the configuration file and the instance directory, ndsconfig cannot create the configuration file and aborts the operation.

16.2.3 ndsconfig get Outputs Junk Characters for Non-English Characters

The `ndsconfig get` command outputs junk characters on Linux for some parameters that contain non-English characters.

To work around this, enter the specify parameter name you want to get, as follows:

```
ndsconfig get <parameter_to_be_displayed>
```

For a list of parameters, refer to the `nds.conf` man page.

16.3 ndsmerge Utility

The PKI servers are not active after a merge operation. They must be restarted using the `npki -l` command.

Merge operations might not be successful on different versions of the product. If your server is running an older version of NDS or eDirectory, update to the latest version of eDirectory, then continue the merge operations.

The merging of two trees will not succeed if containers with the same name subordinate to a tree are present in both the source and target trees. Rename one of the containers, then continue with the merge operation.

During the graft operation, error message `-611 Illegal Containment` might appear. Modify the schema by running `ndsrepair`. Then run `ndsrepair -S` and select *Optional Schema Enhancements*.

16.4 DSTrace Utility

When you turn on the DSTrace screen, an error message might display indicating that a primary object is invalid for the reference link. You can ignore this message if eDirectory is functioning correctly.

16.5 ndsbackup Utility

While backing up eDirectory, `NDS Error: Connect to NDS server failed` might display. This might be caused by eDirectory listening on a port other than the default port 524. At the command line, enter the port number that eDirectory was configured on. For example, if eDirectory is configured on port number 1524, enter the following:

```
ndsbackup sR 164.99.148.82:1524
```

In eDirectory 8.8 and later, while you back up the data, NDS Error: Requires a Password might be displayed. This is because the server might have attributes marked for encryption and you might not have used the option -E to encrypt or decrypt the backup data.

16.6 Using DSRepair

This section consists of the following:

- "Syntax" on page 83
- Section 16.6.2, "Troubleshooting DSRepair," on page 89

Use the DSRepair utility at the server console to do the following:

- Correct eDirectory problems such as bad records, schema mismatches, bad server addresses, and external references.
- Make advanced changes to the eDirectory schema.
- Perform the following operations on the eDirectory database:
  - Check the structure of the database automatically without closing the database and without database intervention.
  - Check the database index.
  - Repair the database without closing the database and locking out users.
  - Reclaim free space by discarding empty records.

16.6.1 Syntax

To run DSRepair, use the following syntax:

```
ndsrepair {-U| -P| -S| -C| -E| -N| -T| -J entry_id}
[-A yes|no] [-O yes|no] [-F filename] [-Ad]
```

or

```
ndsrepair -R [-l yes|no [-u yes|no] [-m yes|no] [-i yes|no] [-f yes|no] [-d yes|no]
[-t yes|no] [-o yes|no] [-r yes|no] [-v yes|no] [-c yes|no] [-A yes|no] [-O yes|no]
[-F filename]
```

IMPORTANT: The `-Ad` option should not be used without prior direction from NetIQ Support personnel.

DSRepair Options

Option	Description
-R	Repairs the local eDirectory database. Use this repair operation to resolve inconsistencies in the local database so that it can be opened and accessed by eDirectory. This option has suboptions that facilitate repair operations on the database. It has function modifiers that are explained in "Function Modifiers Used with the -R Option" on page 85. This option, with no suboptions, is the suggested means of repair unless you are told by NetIQ Support to perform certain operations manually.
-P	Replica and Partition Operations option. Lists the partitions that have replicas stored in the current server's eDirectory database files. The Replica options menu provides options to repair replicas, cancel a partition operation, schedule synchronization, and designate the local replica as the master replica. For more information, see "Replica and Partition Operations Option" on page 86.
-S	Global Schema Operations option. This option contains several schema operations that might be necessary to bring the server's schema into compliance with the master of the Tree object. However, these operations should be used only when necessary. The local and unattended repair operations already verify the schema.

Option	Description
-C	Check External Reference Object option. Checks each external reference object to determine if a replica containing the object can be located. If all servers that contain a replica of the partition with the object are inaccessible, the object will not be 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 12 hours.
-N	Servers Known to This Database option. Lists all servers known to the local eDirectory database. If your current server contains a replica of the Tree partition, this server displays a list of all servers in the eDirectory tree. Select one server to cause the server options to be executed.
-J	Repairs a single object on the local server. You will need to provide the Entry ID (in hexadecimal format) of the object you want to repair. You can use this option instead of using the Unattended Repair (-U) option to repair one particular object that is corrupted. The Unattended Repair option can take many hours depending on the size of database. This option will help you save time.
-T	Time Synchronization option. Contacts every server known to the local eDirectory database and requests information about each server’s time synchronization status. If this server contains a replica of the Tree partition, then every server in the eDirectory tree will be polled. The version of eDirectory that is running on each server is also reported.
-A	Append to the existing log file. The information is added to the existing log file. By default, this option is enabled.
-O	Logs the output in a file. By default, this option is enabled.
-F *filename*	Logs the output in the specified file.
-U	Unattended Full Repair option. Instructs DSRepair to run and exit without further user intervention. This option locks the database and updates server referrals. You can view the log file after the repair has completed to determine what changes DSRepair has made.

Function Modifiers Used with the -R Option

Modifier	Description
-l	Locks the eDirectory database during the repair operation.
-u	Uses a temporary eDirectory database during the repair operation.
-m	Maintains the original unrepaired database.
-i	Checks the eDirectory database structure and the index.
-f	Reclaims the free space in the database.
-d	Rebuilds the entire database.
-t	Performs a tree structure check. Set it to Yes to check all the tree structure links for correct connectivity in the database. Set it to No to skip the check. Default=Yes
-o	Rebuilds the operational schema.
-r	Repairs all the local replicas.
-v	Validates the stream files.
-c	Checks local references.

Global Schema Operations

You can use the `ndsrepair -S ([-Ad] advanced switch)` option to display a list showing all the schema operations that you can perform. The following table shows the available options.

Option	Description
Request Schema From Master Server	Requests the master replica of the root of the tree to synchronize its schema to this server. Any changes to the schema will be propagated to this server from the master replica of the Tree object for the next 24 hours. If all servers request the schema from the master replica, network traffic can increase.
Reset Local Schema	Invokes a schema reset that clears the time stamps on the local schema and requests an inbound schema synchronization. This option is unavailable if executed from the master replica of the Tree partition. This is to ensure that all servers in the tree are not reset at the same time.
Optional Schema Enhancements	Extends and modifies the schema for containment and other schema enhancements. This option requires this server to contain a replica of the Tree partition, and the replica state must be On.

Option	Description
Import Remote Schema (Advanced Switch Option)	Select an eDirectory tree that contains the schema you want to add to the schema of the current tree. After you select a tree, the server that holds the master replica of the Tree partition is contacted. The schema from that server will be used to extend the schema on the current tree.
Declare a New Epoch (Advanced Switch Option)	When you declare a new schema epoch, the master replica of the Tree partition is contacted and illegal time stamps are repaired on the schema declared on that server. All other servers receive a new copy of the schema including the repaired time stamps. If the receiving server contains a schema that was not in the new epoch, objects and attributes that use the old schema are changed to the Unknown object class or attribute.

Replica and Partition Operations Option

Enter the following command to display information about each replica stored on the server:

```
ndsrepair -P
```

Select the required replica. The following options are displayed:

- Repair All Replicas

  Repairs all replicas displayed in the replica table.

- Repair Selected Replica

  Repairs only the selected replica listed in the replica table.

  IMPORTANT: Repairing a replica consists of checking each object in the replica for consistency with the schema and data according to the syntax of the attribute. Other internal data structures associated with the replica are also checked. If you have not repaired the local eDirectory database in the last 30 minutes, you should do so before repairing any replicas.

- Schedule Immediate Synchronization

  Schedules the immediate synchronization of all the replicas. This is useful if you are viewing the DSTrace screen and want to view eDirectory information for the synchronization process without having to wait for it to run as normally scheduled.

- Cancel Partition Operation

  Cancels a partition operation on the selected partition. This option might be necessary if an operation appears to be incomplete or is not completing due to problems in the eDirectory tree, such as a missing server or bad communication links. Some operations might not be cancelled if they have progressed too far.

- Designate This Server as the New Master Replica

  Designates the local replica of the selected partition as the new master replica. Use this option to designate a new master replica if the original master replica is lost.

- Report Synchronization Status of All Servers

  Reports replica synchronization status of all partitions on the current server. It displays the time of the last successful synchronization to all servers and any errors that have occurred since the last synchronization.

- Synchronize the Replica on All Servers

  Determines the complete synchronization status on every server that has a replica of the selected partition. This helps you determine the health of a partition. If all of the servers with a replica of the partition are synchronizing properly, then the partition is considered healthy. Each server performs an immediate synchronization to every other server in the replica ring. Servers do not synchronize to themselves. Therefore, the status for the current server's own replicas is displayed as Host.

- Repair Ring, All Replicas

  Repairs the replica ring of all the replicas displayed in the replica table.

- Repair Ring, Selected Replica

  Repairs the replica ring of selected replica listed in the replica table.

IMPORTANT: Repairing a replica ring consists of checking the replica ring information on each server that contains a replica of a given partition and validating remote ID information. If you have not repaired the local eDirectory database in the last 30 minutes, you should do so before repairing all or selected rings. You can repair the local database using the `-R` option. For more information, see "-R" on page 83.

- View Replica Ring

  Displays a list of all servers that contain a replica of the selected partition. This set of servers is called the replica ring. The replica ring list shows information about the type of replica and current status for each server in the ring. Select a server after viewing the replica ring to view server options.

  Server Options

  - Report Synchronization Status on the Selected Server

    Reports replica synchronization status for a selected partition that has a replica on a selected server. This operation reads the synchronization status attribute from the replica root object on each server that holds replicas of the partitions. It displays the time of the last successful synchronization to all servers and any errors that have occurred since the last synchronization. This option displays a warning message if synchronization has not completed within 12 hours.

  - Synchronize the Replica on the Selected Server

    Determines the complete synchronization status on the selected server that has a replica of the selected partition. This helps you determine the health of a partition. If the server with a replica on the partition is synchronizing properly, the partition is considered healthy. The server is immediately synchronized to every other server in the replica ring. The server does not synchronize with itself. Therefore, the status for the current server's own replica is displayed as Host.

  - Send All Objects to Every Replica in the Ring

    Sends all objects from the selected server in the replica ring to all other servers that contain a replica of the partition. This operation can generate a lot of network traffic. Use this option to ensure that the selected partition's replica on the selected server in the replica ring is synchronized with all other servers in the replica ring. This operation cannot be performed on a server that contains only a subordinate reference replica of the partition.

  - Receive All Objects from the Master to This Replica

    Receives all objects from the master replica to the replica on the selected servers. This operation can generate a lot of network traffic. Use this option to ensure that the selected partition's replica on the selected server in the replica ring is synchronized with the master replica. This operation cannot be performed on a server that contains only a master replica.

- View Entire Server's Name

  Used to view the complete server name when the width of the server name is too long to view from within the server table.

- Remove This Server from Replica Ring

  (Advanced switch option.) Removes a selected server from the selected replica stored on the current server. If a server appears in the replica ring but it is no longer part of the eDirectory tree or no longer contains a replica of the partition, delete the Server object using iManager. When the Server object has been deleted, the object should eventually be excluded from the replica ring.

  WARNING: Misuse of this operation can cause irrevocable damage to the eDirectory database. You should not use this option unless directed by NetIQ Support personnel.

- View Entire Partition Name

  Determines the complete distinguished partition name when the width of the partition is too great to view from within the replica table.

- Repair Time Stamps and Declare a New Epoch

  (Advanced switch option.) Provides a new point of reference to the master replica so that all updates to replicas of the selected partition are current. This operation is always performed on the master replica of a partition. The master replica does not need to be in the local replica on this server. Time stamps are placed on objects when they are created or modified and they must be unique. All time stamps in a master replica are examined. If any time stamps are post-dated to the current network time, they are replaced with a new time stamp.

- Destroy the Selected Replica on This Server

  (Advanced switch option.) Removes the selected replica on this server. Using this option is not recommended. Use this option only when all other utilities are unable to delete the replica.

- Delete Unknown Leaf Objects

  (Advanced switch option.) Deletes all objects in the local eDirectory database that have the unknown object class and maintain no subordinate objects. This option marks Unknown objects for deletion. The deletion will later be synchronized to other replicas in the eDirectory tree.

  WARNING: Use this option only when the objects cannot be modified or deleted using iManager.

Options on Servers Known to This Database

The following repair options are available for servers:

- Repair All Network Address

  Checks the network address for every server in the local eDirectory database. This option searches the SLP directory agent, depending on the transport protocol available, for each server's name. Each address is then compared to the Server object's network address property and the address record of each replica property of every partition Tree object. If the addresses are different, they are updated to be the same.

- Repair Selected Server's Network Address

  Checks the network address for a specific server in the local eDirectory database files. This option searches the SLP directory agent, depending on the transport protocols currently bound for the server's name.

- View Entire Server's Name

  Displays the complete name of the server when the width of the server name is too great to view from within the server's table. This option is the same as the -P option. For more information, see "-P" on page 83.

Examples

To perform an unattended repair and log events in the `/root/ndsrepair.log` file, or to append events to the log file if it already exists, enter the following command:

```
ndsrepair -U  -A no -F /root/ndsrepair.log
```

To display a list of all global schema operations along with the advanced options, enter the following command:

```
ndsrepair -S -Ad
```

To repair the local database by forcing a database lock, enter the following command:

```
ndsrepair -R -l yes
```

NOTE: The input for the `ndsrepair` command can be redirected from an option file. The option file is a text file that can contain replica and partition operation-related options and suboptions that do not require authentication to the server. Each option or suboption is separated by a new line. Make sure that the contents of the file are in the proper sequence. If the contents are not in the proper sequence, the results will be unpredictable.

16.6.2 Troubleshooting DSRepair

Error -786 While Running DSRepair

When using DSRepair, you need to have three times the size of DIB free in the specific partition of your machine in which DSRepair is running.

16.7 Using DSTrace

To use the DSTrace utility in a Linux environment, run the following command from the server prompt:

```
/opt/novell/eDirectory/bin/ndstrace
```

The full syntax for the ndstrace command is as follows:

```
ndstrace [-l|-u|-c "command1;....."|--version] [-h <local_interface:port>] [--
config-file <configuration_file_path>] [thrd <thread ID>] [svty <severity_level>]
[conn <connection_ID>]
```

The DSTrace utility has three main parts:

- "Basic Functions" on page 90
- "Debugging Messages" on page 90
- "Background Processes" on page 93

16.7.1 Basic Functions

The basic functions of DSTrace are to:

- View internal eDirectory activity and debugging messages in Linux.
- Initiate limited synchronization processes.

You can use the DSTrace utility in either UI mode or command line mode. By default, DSTrace runs in UI mode. To start DSTrace in UI mode, enter the following command at the server prompt:

```
/opt/novell/eDirectory/bin/ndstrace
```

To start DSTrace in command line mode, enter the following command at the prompt:

```
/opt/novell/eDirectory/bin/ndstrace -l
```

To initiate basic DSTrace functions, enter commands at the server prompt using the following syntax:

`ndstrace` *`command_option`*

The following table lists the command options that you can enter.

Option	Description
ON	Starts the eDirectory trace screen with basic trace messages.
OFF	Disables the trace screen.
ALL	Starts the eDirectory trace screen and displays all the trace messages.
AGENT	Starts the eDirectory trace screen with the trace messages that are equivalent to the ON, BACKLINK, DSAGENT, JANITOR, RESNAME, and VCLIENT flags.
DEBUG	Turns on a predefined set of trace messages typically used for debugging. The flags set are ON, BACKLINK, ERRORS, EMU, FRAGGER, INIT, INSPECTOR, JANITOR, LIMBER, MISC, PART, RECMAN, REPAIR, SCHEMA, SKULKER, STREAMS, and VCLIENT.
NODEBUG	Leaves the trace screen enabled, but turns off all debugging messages previously set. This option also leaves the messages set to the ON command option.

16.7.2 Debugging Messages

When the DSTrace screen is enabled, the information displayed is based on a default set of filters. If you want to view more or less than the default, you can manipulate the filters using the debugging message flags. The debugging messages help you determine the status of eDirectory and verify that everything is working well.

Each eDirectory process has a set of debugging messages. To view the debugging messages on a particular process, use a plus sign (+) and the process name or option. To disable the display of a process, use a minus sign (-) and the process name or option. The following are some examples:

Message	Description
`set ndstrace = +SYNC`	Enables the synchronization messages.
`set ndstrace = -SYNC`	Disables the synchronization messages.
`set ndstrace = +SCHEMA`	Enables the schema messages.

You can also combine the debugging message flags by using the Boolean operators & (which means AND) and | (which means OR). The syntax for controlling the debugging messages at the server console is as follows:

`set ndstrace = <trace_flag> [parameter]`

The following table describes the trace flags for the debugging messages. You can enter abbreviations for each of the trace flags.

Trace Flag	Description
ABUF	Messages and information related to inbound and outbound packet buffers that contain data being received in conjunction with, or in response to, an eDirectory request.
ALOC	Messages to show the details of memory allocation.
AREQ	Messages related to inbound requests from other servers or clients.
AUTH	Messages and error reports relating to authentication.
BASE	Debug error messages at the minimum debugging level.
BLNK	Backlink and inbound obituary messages and error reports.
CBUF	Messages related to outbound DS Client requests.
CHNG	Change cache messages.
COLL	Status and error reports concerning an object's update information when the update has been previously received.
CONN	Messages that show information about the servers your server is trying to connect to, and about errors and timeouts that might be causing your server not to connect.
DNS	Messages about the eDirectory-integrated DNS server processes.
DRLK	Distributed reference link messages.
DVRS	Messages to show DirXML® driver-specific areas that eDirectory might be working on.
DXML	Messages to show details of DirXML events.
FRAG	Messages from the NCP™ fragger which breaks eDirectory messages into NCP-sized messages.
IN	Messages related to inbound requests and processes.
INIT	Messages related to the initialization of eDirectory.

Trace Flag	Description
INSP	Messages related to the integrity of objects in the source server's local database. Using this flag increases the demands on the source server's disk storage system, memory, and processor. Do not leave this flag enabled unless objects are being corrupted.
JNTR	Messages related to the following background processes: janitor, replica synchronization, and flat cleaner.
LDAP	Messages related to the LDAP server.
LMBR	Messages related to the limber process.
LOCK	Messages related to the use and manipulation of the source server's local database locks.
LOST	Messages related to lost entries.
MISC	Messages from different sources in eDirectory.
MOVE	Messages from the move partition or move subtree operations.
NCPE	Messages to show the server receiving NCP-level requests.
NMON	Messages related to iMonitor.
OBIT	Messages from the obituary process.
PART	Messages related to partition operations from background processes and from request processing.
PURG	Messages about the purge process.
RECM	Messages related to the manipulation of the source server's database.
RSLV	Reports related to the processing of resolve name requests.
SADV	Messages related to the registration of tree names and partitions with Service Location Protocol (SLP).
SCMA	Messages related to the schema synchronization process.
SCMD	Messages showing the details of schema-related operations. They give details of both inbound and outbound synchronization.
SKLK	Messages related to the replica synchronization process.
SPKT	Messages related to eDirectory NCP server-level information.
STRM	Messages related to the processing of attributes with a Stream syntax.
SYDL	Messages showing more details during the replication process.
SYNC	Messages about inbound synchronization traffic (what is being received by the server).
TAGS	Displays the tag string that identifies the trace option that generated the event on each line displayed by the trace process.
THRD	Messages to show when any background processes (threads) begin and end.
TIME	Messages about the transitive vectors that are used during the synchronization process.

Trace Flag	Description
TVEC	Messages related to the following attributes: Synchronize Up To, Replica Up To, and Transitive Vector.
VCLN	Messages related to the establishment or deletion of connections with other servers.

As you use the debugging messages in DSTrace, you will find that some of the trace flags are more useful than others. One of the favorite DSTrace settings of NetIQ Support is actually a shortcut:

```
set ndstrace = A81164B91
```

This setting enables a group of debugging messages.

16.7.3 Background Processes

In addition to the debugging messages, which help you check the status of eDirectory, there is a set of commands that force the eDirectory background processes to run. To force the background process to run, place an asterisk (*) before the command. For example:

```
set ndstrace = *H
```

You can also change the status, timing, and control for a few of the background processes. To change these values, place an exclamation point (!) before the command and enter a new parameter or value. For example:

```
set ndstrace = !H 15 (parameter_value_in_minutes)
```

The following is the syntax for each statement controlling the background processes of eDirectory:

```
set ndstrace = <trace_flag> [parameter]
```

The following table lists the trace flags for the background processes, any required parameters, and the process the trace flags are displayed.

Trace Flag	Parameters	Description
*A	None	Resets the address cache on the source server.
*AD	None	Disables the address cache on the source server.
*AE	None	Enables the address cache on the source server.
*B	None	Schedules the backlink process to begin execution on the source server in one second.
!B	Time	Sets the interval (in minutes) for the backlink process. Default=1500 minutes (25 hours) Range=2 to 10080 minutes (168 hours)
*CT	None	Displays the source server's outbound connection table and the current statistical information for the table. These statistics do not give any information about the inbound connections from other servers or clients to the source server.

Trace Flag	Parameters	Description
*CTD	None	Displays, in comma-delimited format, the source server's outbound connection table and the current statistical information for the table. These statistics do not give any information about the inbound connections from other servers or clients to the source server.
*D	Replica rootEntry ID	Removes the specified local entry ID from the source server's Send All Object list. The entry ID must specify a partition root object that is specific to the server's local database. This command is usually used only when a Send All Updates process is endlessly trying to show updates and failing because a server is inaccessible.
!D	Time	Sets the inbound and outbound synchronization interval to the specified number of minutes. Default=24 minutes. Range=2 to 10080 minutes (168 hours)
!DI	Time	Sets the inbound synchronization interval to the specified number of minutes. Default=24 minutes Range=2 to 10080 minutes (168 hours)
!DO	Time	Sets the outbound synchronization interval to the specified number of minutes. Default=24 minutes Range=2 to 10080 minutes (168 hours)
*E	None	Reinitializes the source server's entry cache.
!E	None	Schedules the inbound and outbound synchronization processes to begin execution.
!EI	None	Schedules the inbound synchronization process to begin execution.
!EO	None	Schedules the outbound synchronization process to begin execution.
*F	None	Schedules the flat cleaner process, which is part of the janitor process, to begin execution on the source server in five seconds.
!F	Time	Sets the interval (in minutes) for the flat cleaner process. Default=240 minutes (4 hours) Range=2 to 10080 minutes (168 hours)

Trace Flag	Parameters	Description
*FL	1-10	Sets the number of rolling log files used by DSTrace. If you set this parameter to any value greater than 1, once the source server's `ndstrace.log` file reaches the configured maximum file size, DSTrace renames the file `ndstrace1.log` and creates a new `ndstrace.log` file. When that file reaches its maximum file size, the previous `ndstrace1.log` file is renamed `ndstrace2.log`, and the more recent `ndstrace.log` file is renamed `ndstrace1.log`. This process continues until DSTrace reaches the maximum number of rolling log files set by this option. Once the specified limit is reached, the oldest log files will be deleted and only the specified maximum number of rolling files will be maintained. You can configure a maximum of 10 rolling log files. By default, DSTrace must use at least 1 rolling log file. If you set this parameter to 0, DSTrace uses 1 as the parameter value.
*G	Replica rootEntry ID	Rebuilds the change cache of the specified root partition ID.
*H	None	Schedules the replica synchronization process to begin execution immediately on the source server.
!H	Time	Sets the interval (in minutes) for the heartbeat synchronization process. Default=30 minutes Range=2 to 1440 minutes (24 hours)
*HR	None	Clears the in-memory last-sent vector.
*I	Replica rootEntry ID	Adds the specified local entry ID to the source server's Send All Object list. The entry ID must specify a partition root object that is specific to the server's local database. The replica synchronization process checks the Send All Object list. If the entry ID of a partition's root object is in the list, eDirectory synchronizes all objects and attributes in the partition, regardless of the value of the Synchronized Up To attribute.
!I	Time	Sets the interval (in minutes) for the heartbeat synchronization process. Default=30 minutes Range=2 to 1440 minutes (24 hours)
*J	None	Schedules the purge process, which is part of the replica synchronization process, to begin running on the source server.
!J	Time	Sets the interval (in minutes) for the janitor process. Default=2 minutes Range=1 to 10080 minutes (168 hours)
*L	None	Schedules the limber process to begin running on the source server in five seconds.

Trace Flag	Parameters	Description
*M	Bytes	Changes the maximum file size used by the source server's `ndstrace.log` file. The command can be used regardless of the state of the debug file. The bytes specified must be a decimal value between 10000 bytes and 100 MB. If the value specified is higher or lower than the specified range, no change occurs.
!M	None	Reports the maximum memory used by eDirectory.
!N	0|1	Sets the name form. 0=hex only 1=full dot form
*P	None	Displays the tunable parameters and their default settings.
*R	None	Resets the size of the `ndstrace.log` file to zero bytes. This command is the same as the SET parameter NDS Trace File Length Set to Zero.
*S	None	Schedules the Skulker process, which checks whether any of the replicas on the server need to be synchronized.
!SI	Time	Sets the interval (in minutes) for the inbound schema synchronization process. Default=24 minutes Range=2 to 10080 minutes (168 hours)
!SO	Time	Sets the interval (in minutes) for the outbound schema synchronization process. Default=24 minutes Range=2 to 10080 minutes (168 hours)
!SIO	Time	Disables the inbound schema synchronization process for the specified number of minutes. Default=24 minutes Range=2 to 10080 minutes (168 hours)
!SO0	Time	Disables the inbound schema synchronization process for the specified number of minutes. Default=24 minutes Range=2 to 10080 minutes (168 hours)
*SS	None	Forces immediate schema synchronization.
*SSA	None	Schedules the schema synchronization process to begin immediately and forces schema synchronization with all target servers, even if they have been synchronized in the last 24 hours.
*SSD	None	Resets the source server's Target Schema Sync list. This list identifies which servers the source server should synchronize with during the schema synchronization process. A server that does not hold any replicas sends a request to be included in the target list of a server that contains a replica with its Server object.
*SSL	None	Prints the schema synchronization list of target servers.
*ST	None	Displays the status information for the background processes on the source server.
*STX	None	Displays the status information for the backlink process (external references) on the source server.

Trace Flag	Parameters	Description
*STS	None	Displays the status information for the schema synchronization process on the source server.
*STO	None	Displays the status information for the backlink process (obituaries) on the source server.
*STL	None	Displays the status information for the limber process on the source server.
!T	Time	Sets the interval (in minutes) for checking the server's UP state. Default=30 minutes Range=1 to 720 minutes (12 hours)
*U	Optional ID of server	If the command does not include an entry ID, this changed the status of any server that has been previously labeled **down** to **up**. If the command includes a local entry ID, it changes the status of the specified server from **down** to **up**. Entry IDs are specific to the source server's database and must refer to an object that represents a server.
!V	A list	Lists the restricted eDirectory versions. If no versions are listed, there are no restrictions. Each version is separated by a comma.
*Z	None	Displays the currently scheduled tasks.

17 NMAS on Linux

- Section 17.1, “Unable to Log In Using Any Method,” on page 99
- Section 17.2, “The User Added Using the ICE Utility Is Unable to Log In Using Simple Password,” on page 99

17.1 Unable to Log In Using Any Method

After installing and configuring NMAS, restart the eDirectory server.

After reinstalling a method after you have uninstalled a previous instance of that method, restart the eDirectory server.

17.2 The User Added Using the ICE Utility Is Unable to Log In Using Simple Password

While adding users with simple passwords through the NetIQ Import Conversion Export utility, use the `-l` option.

18 Troubleshooting on Windows

- Section 18.1, “The eDirectory for Windows Server Won’t Start,” on page 101
- Section 18.2, “The Windows Server Can’t Open the eDirectory Database Files,” on page 101
- Section 18.3, “SLP_NETWORK_ERROR(-23) Occurs in Windows Machines,” on page 102
- Section 18.4, “Incorrect Installation Path Appears in the Browse Page During eDirectory Installation,” on page 103

18.1 The eDirectory for Windows Server Won’t Start

If the eDirectory server fails to start when you boot the Windows server, a message will notify you that the service failed to start.

If there are no other eDirectory database replicas, users can't log in.

If there are other replicas, logging in might be slow and you will see communication errors and synchronization errors on the servers holding those replicas.

- The eDirectory server entries in the Windows Registry might have been edited, or the Windows Registry might be corrupt.
- eDirectory database files might have been corrupted or deleted.
- If the eDirectory server cannot start because another service did not start, you can get more information from *Start* > *Programs* > *Administrative Tools* > *Event Viewer*.

  You will need to resolve the related-service problem before starting the eDirectory server.
- The Registry or eDirectory executable files are corrupted or lost. Run the SAMMIG utility in the system directory. Select *Uninstall NDS on Windows NT* and include new eDirectory information in the NT domain. Continue with the uninstallation process until completed. Then restart `sammig.exe` and proceed to install eDirectory.
- Database files have been corrupted or deleted. If the eDirectory server comes up on the NT server but the service can’t open the eDirectory database files, see Section 18.2, “The Windows Server Can’t Open the eDirectory Database Files,” on page 101.
- The eDirectory server is not connected to a hub or switch or directly to a workstation (using a crossover cable). Connect the server to a hub or switch.

18.2 The Windows Server Can’t Open the eDirectory Database Files

If the eDirectory server can’t open the database files, a message on the Windows server will notify you.

If there are no other database replicas, users can't log in.

If there are other replicas, logging in might be slow and you will see communication errors and synchronization errors on the servers holding those replicas.

- The database files might have been corrupted through disk errors on the NT/2000 server.
- Someone might have deleted one or more of the database files.

If other replicas of the eDirectory database exist, complete the following steps:

1. Start NetIQ iManager from an administrative workstation.
2. Remove the corrupted replica from the replica ring.

   See "Deleting a Replica" in the *NetIQ eDirectory 8.8 SP8 Administration Guide* for more information.
3. Run the `sammig.exe` utility in the system directory, located at `c:\winnt\system32` on the NT server, or from the *Start* menu.
4. Select the option to create a new replica on the eDirectory server.

If this eDirectory server holds the only replica of the partition, complete the following steps:

1. Run the `sammig.exe` utility in the system directory, located at `c:\winnt\system32` on the NT Server, or from the *Start* menu.
2. Select *Uninstall NDS* on Windows and revert to the previous Windows domain state.
3. Continue with the uninstallation process until it has completed.
4. Restart the Migration Tool and proceed to install eDirectory on Windows.
5. Move the `User` objects from the NT/2000 domain to the eDirectory tree.

18.3 SLP_NETWORK_ERROR(-23) Occurs in Windows Machines

The Service Location Protocol (SLP) query returns -23 SLP_NETWORK_ERROR on a virtual machine having a DHCP address or on a physical or a virtual machine in which SLP is not broadcasted.

You can avoid the SLP error by configuring the Directory Agent in your network in one of these ways:

1. Copy the `C:\Windows\System32\Novell\eDir\OpenSLP\slp.conf` file to the `c:\Windows\` directory.
2. Open the `slp.conf` file by using a text editor and change the following line:

   ```
   ;net.slp.DAAddresses = myDay1,myDa2,myDa3
   ```

   to

   ```
   net.slp.DAAddresses = <Give your DA Address>
   ```
3. Save the changes, then close the file.

OR

1. Copy the `C:\Windows\System32\Novell\eDir\OpenSLP\slp.conf` file to the `c:\Windows\` directory.
2. Open the `slp.conf` file by using a text editor and change the following line:

   ```
   ;net.slp.isDA = true
   ```

   to

```
net.slp.isDA = true
```

3 Save the changes, then close the file.

18.4 Incorrect Installation Path Appears in the Browse Page During eDirectory Installation

Manually change the path to the desired location.

19 Accessing HTTPSTK When DS Is Not Loaded

You can set up a preconfigured admin user that allows access to the HTTP Protocol Stack (HTTPSTK) when DS is not loaded. The preconfigured admin user, sadmin, has rights that are equivalent to the eDirectory Admin User object. If the server is in a state where eDirectory is not functioning correctly, you can log in to the server as this user and perform all the diagnostic and debugging tasks necessary that do not require eDirectory.

- Section 19.1, "Setting the sadmin Password on Windows," on page 105
- Section 19.2, "Setting the sadmin Password on Linux," on page 105

19.1 Setting the sadmin Password on Windows

Use the DHost remote manager page (accessible through the `/dhost` URL or from the root page) to set the sadmin password. `dhost.exe` must be running on the eDirectory server in order for you to set or change the sadmin password.

1. Open a Web browser.
2. In the address (URL) field, enter the following:

   `http://server.name:port/dhost`

   for example:

   `http://MyServer:80/dhost`

   You can also use the server IP address to access the DHost iConsole. For example:

   `http://137.65.135.150:80/dhost`
3. Specify a username, context, and password.
4. Click *HTTP Server*, then specify an sadmin password.
5. Verify the password you just specified, then click *Submit*.

19.2 Setting the sadmin Password on Linux

You can use either the DHost remote management page or the ndsconfig utility.

DHost remote management page

Use the DHost remote manager page (accessible through the `/dhost` URL or from the root page) to set the sadmin password. NetIQ eDirectory server must be running on the eDirectory server in order for you to set or change the sadmin password.

1. Open a Web browser.
2. In the address (URL) field, enter the following:

   `http://server.name:port/dhost`

   for example:

   `http://MyServer:80/dhost`

   You can also use the server IP address to access the DHost iConsole. For example:

   `http://137.65.135.150:80/dhost`
3. Specify a username, context, and password.
4. Click *HTTP Server*, then specify an sadmin password.
5. Verify the password you just specified, then click *Submit*.

ndsconfig

Use the ndsconfig utility to set the sadmin password. ndsd must be running on the eDirectory server in order for you to set or change the sadmin password.

Enter the following at the server console

`ndsconfig set http.server.sadmin-pwd=password`

where *password* is the new sadmin password.

For more information on using the ndsconfig utility, see "ndsconfig Utility Parameters" in the *NetIQ eDirectory 8.8 SP8 Installation Guide*.

20 Encrypting Data in eDirectory

In NetIQ eDirectory 8.8 and later, you can encrypt specific sensitive data while they are stored on the disk and while they are accessed by the client. This chapter provides you information on the errors you might encounter while using the encrypted attributes and replication features in eDirectory 8.8 and later. For more information on encrypted attributes and replication, refer to the *NetIQ eDirectory 8.8 SP8 Administration Guide* (http://www.netiq.com/documentation/edir88/edir88/data/a2iii88.html).

For information on other error messages in eDirectory, refer to the NetIQ Error Codes Web site (http://www.novell.com/documentation/nwec/)

20.1 Error Messages

This section contains information on the following error messages:

- Section 20.1.1, "-6090 0xFFFFE836 ERR_ER_DISABLED," on page 107
- Section 20.1.2, "-6089 0xFFFFE837 ERR_REQUIRE_SECURE_ACCESS," on page 107
- Section 20.1.3, "-666 FFFFFD66 INCOMPATIBLE NDS VERSION," on page 108

20.1.1 -6090 0xFFFFE836 ERR_ER_DISABLED

The eDirectory replica synchronization process tried to start encrypted replication with the target server. But the target eDirectory server has the encrypted replica synchronization process disabled

Possible Cause

Encrypted replication is disabled on the target eDirectory server.

Action

Enable encrypted replication on the target eDirectory server.

20.1.2 -6089 0xFFFFE837 ERR_REQUIRE_SECURE_ACCESS

An application (client access) tried to access an encrypted attribute over a clear text channel.

Source

eDirectory or NDS

Possible Cause

The encrypted attributes are configured to be accessed only over a secure channel. The application is trying to access the encrypted attributes over a clear text channel.

Action

The application should access the encrypted attributes through a secure channel, like LDAP secure channel or HTTP secure channel.

Possible Cause

If you get this error during replication, one or more servers in the replica ring have some attributes marked for encryption and are configured to be accessed only over secure channel.

Action

Change the configuration of the encrypted attribute policy, so that the encrypted attributes can be accessed over insecure channels. For more information, refer to the *NetIQ eDirectory 8.8 SP8 Administration Guide* (http://www.netiq.com/documentation/edir88/edir88/data/a2iii88.html).

Possible Cause

If you get this error when encrypted replication is configured at the partition level or between the replicas of the partition, then the replica ring has pre-eDirectory 8.8 servers in it.

Action

Upgrade all the servers in the replica ring to a version compatible with eDirectory 8.8.

20.1.3 -666 FFFFFD66 INCOMPATIBLE NDS VERSION

Text goes here

Possible Cause

If encrypted replication is enabled at a partition level and if you are trying to add a replica of this partition to an eDirectory server, then the eDirectory version on this server is incompatible with the version on the source server.

Action

Upgrade the server to a compatible version of eDirectory.

Possible Cause

If the parent partition has pre-eDirectory 8.8 servers (mixed version ring) and if the child partition has ER enabled, the merge and/or join partition operations would be disallowed and the ERR_INCOMPATIBLE_DS_VERSION error will be returned.

The reason for this is that the child partition contains sensitive data with ER enabled at the partition level and the parent partition having pre-eDirectory 8.8 server. With ER enabled only between eDirectory 8.8 servers, on merging, sensitive data is exposed when replicating to pre-eDirectory 8.8 servers.

Action

1. Upgrade the server to a compatible version of eDirectory.

   OR

2. Disable ER at the parent or child partition.

   NOTE: On disabling ER, replication will happen in the clear text form.

20.2 Problem With Duplicate Encryption Algorithms

If you add an attribute for encryption using LDIF, do not associate duplicate algorithms with one attribute.

For example, marking *title* as an encrypted attribute with AES and DES encryption algorithms makes it unclear as to which algorithm is ultimately considered. Each time when limber is run it appears the title attribute toggles between AES and DES. Therefore, it seems as though there were some configuration changes.

To prevent such scenarios, we recommend you to avoid duplicate algorithms been assigned to the same attribute.

This does not happen if you mark an attribute for encryption using iManager.

20.3 Encryption of Stream Attributes

Stream attributes might be present as clear text data. This is because eDirectory 8.8 does not encrypt stream attributes.

20.4 Configuring Encrypted Replication through iManager

You cannot configure encrypted replication through iManager if any server in the replica ring is down.

20.5 Viewing or Modifying Encrypted Attributes through iManager

If an attribute of an object is encrypted, you cannot view or modify the object by using iManager 2.5.

To work around this issue, you can view or modify the encrypted attribute over a secure channel, using any of the following methods:

- LDAP: The LDAP request must be send over a secure channel, which means that the trusted root certificate of the server must be used.
- ICE: LDIF scripts can be used to modify the object. If you do this, ICE must use a secure channel.
- Use iManager 2.5 FP2, iManager 2.6, or later.

NOTE: We recommend using iManager 2.6 or later for viewing or modifying encrypted attributes.

Alternatively, you can turn off the secure channel required option for viewing or modifying the encrypted attributes by disabling the `requireSecure` attribute in the EA policy. This makes the object and the encrypted attributes accessible by any client over clear text channel. After this, iManager will be able to access the object.

20.6 Merging Trees With Encrypted Replication Enabled Fails

When encrypted replication is enabled, merging trees fails. Disable secure replication on each tree before doing a merge.

20.7 Limber Displays -603 Error

Limber displays the -603 error if the server has only sub-ref replica of the encrypted attribute policy partition.

To work around this issue, do any one of the following:

- Give read access to the NCP server object. You can do this through iManager by adding a trustee at the tree root and giving read access to NCP server object. In the attributes, specify `attrEncryptionDefinition` and `attrEncryptionRequiresSecure`.
- Give Public Read access to the following attributes through LDAP or ndssch:
  - `attrEncryptionDefinition`
  - `attrEncryptionRequiresSecure`

21 The eDirectory Management Toolbox

The NetIQ eDirectory Management Toolbox (eMBox) lets you access all of the eDirectory backend utilities remotely as well as on the server.

eMBox works with NetIQ iManager to provide Web-based access to eDirectory utilities such as DSRepair, DSMerge, Backup and Restore, and Service Manager.

IMPORTANT: Role Based Services must be configured through iManager to the tree that is to be administered in order for eMBox tasks to be run.

All functions are accessible, either on the local server or remotely, through a command line client. You can perform tasks for multiple servers from one server or workstation using the eMBox Client. To run all eDirectory Management Tools (eMTools), including Backup, DSRepair, DSMerge, Schema Operations, and eDirectory Service Manager, eMBox must be loaded and running on the eDirectory server.

- Section 21.1, “Unable to Stop the eMTool Services,” on page 111
- Section 21.2, “Restore gives -6020 error,” on page 111
- Section 21.3, “eDirectory Service Manager Issues,” on page 112

21.1 Unable to Stop the eMTool Services

When running the command `serviceStop -n{service}`, where *{service}* is one of the services (`libsasl.so`, `libncpengine.so`, `libhttpstk.so`, or `libdsloader.so`), the following error occurs:

```
Service {service} could not be stopped, Error : -660
```

This is not an error. You cannot stop these processes (specifically `libsasl.so`, `libncpengine.so`, `libhttpstk.so`, and `libdsloader.so`), because there are other modules dependent on them.

21.2 Restore gives -6020 error

If you have roll forward logs in a default location, while performing Restore operation using DSBK or eMBox Client, you will get -6020 error. To avoid this error, you need to give `-s` switch in the `restore` command.

21.3 eDirectory Service Manager Issues

If you use the eDirectory Service Manager in iManager to stop eDirectory, restarting it through Service Manager is not possible. Use the eDirectory Services utility (`C:\novell\NDS\NDSCons.exe`) on the eDirectory server to restart eDirectory.

- Section 21.3.1, "Deletion of a Moved Object," on page 112
- Section 21.3.2, "Issue with Moving a Dynamic Group," on page 112
- Section 21.3.3, "Issue with Repairing Network Addresses through eMBox," on page 112
- Section 21.3.4, "Viewing French Man Pages," on page 112
- Section 21.3.5, "Deleting a Moved Object," on page 112
- Section 21.3.6, "eDirectory Does Not Generate a Logout Event due to eDirectory Client Limitation," on page 113
- Section 21.3.7, "Issues Generated by TERM While Running DSTrace," on page 113
- Section 21.3.8, "eMBox Does Not Handle Double-Byte Characters," on page 113

21.3.1 Deletion of a Moved Object

Deletion of a moved object might fail (error -637) in a tree with two or more servers.

21.3.2 Issue with Moving a Dynamic Group

Moving a Dynamic Group object with `dynamicgroup` in the `Object Class` attribute to another container breaks the dynamic group functionality. After the move, queries and searches on dynamic members do not work.

21.3.3 Issue with Repairing Network Addresses through eMBox

When you repair the network addresses through eMBox, it throws the following errors because eMBox is not updated with the recent fixes for repair:

```
ERROR: Could not find a net address for this server - Error : 11004
```

```
ERROR: Could not connect. Error : 11004
```

21.3.4 Viewing French Man Pages

To view the French man page on Red Hat Linux, export the following:

```
export MANPATH=/opt/novell/man/frutf8:/opt/novell/eDirectory/man/frutf8
```

21.3.5 Deleting a Moved Object

Deletion of a moved object might fail (error -637) in a tree with two or more servers.

21.3.6 eDirectory Does Not Generate a Logout Event due to eDirectory Client Limitation

eDirectory does not generate a Logout event when you log out of iManager. This is because of a technical limitation in the client part of eDirectory.

Auditing applications can use NWDS APIs to receive logout events. Applications that use LDAP can monitor logout with unbind events.

21.3.7 Issues Generated by TERM While Running DSTrace

TIME and TAGS tags are displayed as enabled (underlined), but not by default. When the TERM is set to VT100 or xterm from a Linux terminal, these tags are displayed as if they are enabled (underlined). This issue does not occur for any other term, such as dtterm.

21.3.8 eMBox Does Not Handle Double-Byte Characters

eMBox does not handle double-byte characters for setting a roll-forward directory through the eMBox client and iManager. This can still be done by using DSBK.

22 SASL-GSSAPI

This section discusses the error messages logged by the SASL-GSSAPI authentication mechanism.

- Section 22.1, “SASL-GSSAPI Issues,” on page 115
- Section 22.2, “Log File,” on page 115
- Section 22.3, “Error Messages,” on page 115

22.1 SASL-GSSAPI Issues

- Section 22.1.1, “Issue with Multiple User Objects,” on page 115
- Section 22.1.2, “Authorization ID,” on page 115

22.1.1 Issue with Multiple User Objects

LDAP bind with SASL GSSAPI fails if the same Kerberos principal is associated with multiple eDirectory user objects.

22.1.2 Authorization ID

RFC2222 specifies support for an authorization ID sent by the user and client. This is not supported by the SASL GSSAPI method.

22.2 Log File

Error messages are logged in the `ndsd.log` file in Linux installations.

22.3 Error Messages

SASL-GSSAPI: Reading Object *user_FDN* FAILED *eDirectory error code*

Cause: This error is generated in eDirectory. The `user_FDN` object does not exist.

SASL-GSSAPI: Reading principal names for *user_FDN* failed *eDirectory error code*

Cause: This error is generated in eDirectory. The Kerberos principal name is not attached to the user object (`userdn`).

SASL-GSSAPI: Reading Object *Realm_FDN* FAILED *eDirectory error code*

Cause: This error is generated in eDirectory. The `realm` object does not exist.

SASL-GSSAPI: Not enough memory

Cause: Not enough memory to perform the specific operation.

SASL-GSSAPI: Invalid Input Token

Cause: Token from client is defective or invalid

SASL-GSSAPI: NMAS error *NMAS error code*

Cause: This error is generated in NMAS and is an internal error.

SASL-GSS: Invalid LDAP service principal name *LDAP_service_principal_name*

Cause: The LDAP service principal name is invalid.

SASL-GSS: Reading LDAP service principal key from eDirectory failed

Cause: The LDAP service principal object is not created.

Cause: The realm object's master key is changed.

Cause: The LDAP service principal object was not found in the subtree of the realm to which it belongs.

SASL-GSS: Creating GSS context failed

Cause: The time is not in sync between the client, KDC and the eDirectory servers.

Cause: The key of the LDAP service principal was changed in the Kerberos database, but not updated in eDirectory.

Cause: The encryption type is not supported.

SASL GSSAPI: Invalid user FDN = *user_FDN*

Cause: The user FDN provided by the client is not valid.

SASL GSSAPI: No user DN is associated with principal *client_principal_name*

Cause: A `user` object under the subtree is not attached with the Kerberos principal name.

SASL GSSAPI: More than one user DN is associated with principal *client_principal_name*

Cause: More than one `user` object under the subtree is associated with the same principal.

ldap_simple_bind_s: Invalid credentials major = 1, minor =0

Cause: The cause might be the version mismatch between the LDAP service principal on the KDC server and the LDAP service principal on the eDirectory server. This is because every time you extract the LDAP service principal key to the keytab file, the key version number gets incremented.

Action:

Complete the following procedure:

1. Update the key in eDirectory server so that the version numbers are in sync.
2. Destroy the tickets at the client.
3. Get the TGT again for the principal.
4. Perform the LDAP `sasl` bind operation.

23 Miscellaneous

- Section 23.1, “Backing Up a Container,” on page 120
- Section 23.2, “Repeated eDirectory Logins,” on page 120
- Section 23.3, “Enabling Event System Statistics,” on page 120
- Section 23.4, “Tracking Memory Corruption Issues on Linux,” on page 120
- Section 23.5, “TCP Connection not Terminating after Abnormal Logout,” on page 120
- Section 23.6, “NDS Error, System Failure (-632) Occurs When Doing ldapsearch for the User Objects,” on page 122
- Section 23.7, “Disabling SecretStore,” on page 122
- Section 23.8, “Viewing SLP Man Pages,” on page 122
- Section 23.9, “dsbk Configuration File Location,” on page 123
- Section 23.10, “SLP Interoperability Issues on OES Linux,” on page 123
- Section 23.11, “ldif2dib Fails to Open the Error Log File When the DIB Directory Exists In the Custom Path,” on page 123
- Section 23.12, “eDirectory Server Does Not Start Automatically in the Virtual SLES 10,” on page 123
- Section 23.13, “ndsd Does Not Start After a System Crash,” on page 123
- Section 23.14, “Do not Execute DSTrace With All Tags Enabled on Linux Computers,” on page 124
- Section 23.15, “LDAP is Not RFC Compliant For Anonymous Search Requests,” on page 124
- Section 23.16, “Troubleshooting Ports with Custom eDirectory 8.8 Instances,” on page 124
- Section 23.17, “Rebooting the Host,” on page 124
- Section 23.18, “ndsd Not Listening at the Loopback Address on a Given NCP Port,” on page 124
- Section 23.19, “LDAP Transaction OIDs,” on page 124
- Section 23.20, “Errors -5871 and -5875 in LDAP Trace,” on page 125
- Section 23.21, “NDSCons Gives -625 Error if a Tree is Renamed,” on page 125
- Section 23.22, “Listening on Multiple NICs Slows Down eDirectory ldapsearch Performance,” on page 125
- Section 23.23, “Unable to Limit the Number of Concurrent Users on Linux Platforms,” on page 125
- Section 23.24, “ndsd Fails to Shut Down Due to SLP,” on page 125
- Section 23.25, “Restarting NLDAP on Windows,” on page 126
- Section 23.26, “SecretStore over LDAP,” on page 126
- Section 23.27, “Interoperability Issues,” on page 126

23.1 Backing Up a Container

While using ndsbackup to backup a container that has many objects (like a million), it might take some time to get the list of the objects in the container and start their individual backup.

23.2 Repeated eDirectory Logins

Repeated eDirectory logins can use up the available memory. Disable the Login Update attribute using iMonitor to overcome this problem.

23.3 Enabling Event System Statistics

Time related statistics are maintained for every event thrown and consumed in eDirectory. This information is useful for troubleshooting event consumer issues. These statistics are not required for normal functioning of directory; therefore, they are disabled for performance reasons. Event statistics can be enabled at runtime by using iMonitor advanced configuration parameters.

To view the event statistics, set the `ENABLE_EVENT_STATISTICS` parameter and restart the server. It is a permanent configuration parameter.

23.4 Tracking Memory Corruption Issues on Linux

On Linux platforms, eDirectory uses Google malloc (`libtcmalloc`) as the default memory allocator.

To track memory corruption issues, set the `MALLOC_CHECK_ environment variable` in the ndsd startup script. The startup script checks for this variable. If set, the default system malloc is used, else `libtcmalloc` is loaded.

MALLOC_CHECK Settings in ndsd

- When `MALLOC_CHECK_` is set to 0, any detected heap corruption is silently ignored.
- When `MALLOC_CHECK_` is set to 2, abort is called immediately.

  This helps to identify the real cause of the memory corruption at early stages, which might be difficult to track later.

23.5 TCP Connection not Terminating after Abnormal Logout

Sometimes the OES Linux server fails to detect a client host that has gone down abruptly due to a workstation crashing or a power outage. However, the connection is active for the default timeout (about 12 to 15 minutes) before the connection is cleared. If you have set the concurrent connections to 1, it is recommended that you either terminate the connection manually, or wait for the estimated timeout before logging in again. This situation occurs when the watchdog process fails to close the connection cleanly. So, if the concurrent connections are set to 1 and the connection is not cleared by

the watchdog, users cannot log in. Linux kernel provides three parameters to change the way `keepalive` probes work from the server side. Use these parameters to implement a workaround at the TCP level.

These parameters are available in `/proc/sys/net/ipv4/` directory.

- `tcp_keepalive_time`: Determines the frequency of sending the TCP `keepalive` packets to keep a connection alive if it is currently unused. This value is used only when `keepalive` is enabled.

  The `tcp_keepalive_time` takes an integer value in seconds. The default value is 7200 seconds or 2 hours. This holds good for most of the hosts and does not require many network resources. If you set this value to low, it engages your network resources with unnecessary traffic.
- `tcp_keepalive_probes`: Determines the frequency of sending TCP `keepalive` probes before deciding a broken connection.

  The `tcp_keepalive_probes` takes an integer value, recommended less than 50 depending on your `tcp_keepalive_time` and the `tcp_keepalive_interval` values. The default is to set to 9 probes before informing the application of the broken connection.
- `tcp_keepalive_intvl`: Determines the duration for a reply for each `keepalive` probe. This value is important to calculate the time before your connection has a `keepalive` death.

  The `tcp_keepalive_intvl` takes an integer value, the default is 75 seconds. So, 9 probes with 75 seconds each will take approximately 11 minutes. The default values of the `tcp_keepalive_probes` and `tcp_keepalive_intvl` variables can be used to evaluate the default time before the connection is timed out because of `keepalive`.

Modify these three parameters in a way that the change does not generate a lot of extra network traffic and still solves the problem. A sample modification could be as follows (a 3-minute detection time):

- `tcp_keepalive_time set -120`
- `tcp_keepalive_probes - 3`
- `tcp_keepalive_intvl - 20`

NOTE: Be careful with the parameter settings and avoid setting the already valid connections.

The settings take effect immediately after the files are modified. You need not restart any services. However, the settings are valid for the current session only. Once the server is re-booted, the settings revert to the default settings.

To make the setting permanent (even after a reboot), do the following:

Add the following entries in `/etc/sysctl.conf.`

- net.ipv4.tcp_keepalive_time=120
- net.ipv4.tcp_keepalive_probes=3
- net.ipv4.tcp_keepalive_intvl=20

We recommend these settings only if all the clients and servers are connected through LAN.

23.6 NDS Error, System Failure (-632) Occurs When Doing ldapsearch for the User Objects

Import the user objects with simple password and then enable universal password for the container where the user objects are imported. Stop the DS server and set the environment as `NDSD_TRY_NMASLOGIN_FIRST=true` and then start DS Server. When you do an ldapsearch for the user objects, which were imported with simple password, you get the following error:

```
ldap_bind: Unknown error,  additional info: NDS error: system failure (-632)
```

To resolve this issue, set the default login sequence as simple password for the container where user objects are imported before doing ldapsearch for those user objects.

When LDAP requests NMAS to log in a user, NMAS uses the default login sequence. If you do not specify a default login sequence for these users, then it will use the NDS sequence. If these users are not given an NDS password when you imported them, then the NDS sequence will not work. If you enable universal password, then the simple password will be synchronized with the NDS password and universal password when the user logs in with the simple password.

23.7 Disabling SecretStore

An eDirectory administrator can disable SecretStore using the following processes:

23.7.1 On Linux

1. Go to the `nds-modules` directory and rename or move the following SecretStore modules:

   ```
   libsss.so
   libssncp.so
   libssldp.so
   ```

2. Restart the server.

Alternatively, you can also comment out the line in the `/etc/opt/novell/eDirectory/conf/ndsmodules.conf` file that loads ssncp.

23.7.2 On Windows

1. Go to the `novell\nds` directory and rename or move the following SecretStore modules:

   ```
   lsss.dll
   sss.dlm
   ssncp.dlm
   ssldp.dlm
   ```

2. Restart the server.

23.8 Viewing SLP Man Pages

To view the man pages for SLP, you need to set the paths for the man pages. For example, on AIX, you must set the manpath to `/usr/share/man` instead of `/opt/novell/man`.

23.9 dsbk Configuration File Location

The `dsbk.conf` file is located in `/etc` instead of the location relative to the specific instance of eDirectory.

23.10 SLP Interoperability Issues on OES Linux

OpenSLP implements SLPv2, but NetIQ SLP (NDSslp) on Linux and Windows platforms implements SLPv1.

SLPv1 UAs do not receive replies from SLPv2 SAs, and SLPv2 UAs do not receive replies from SLPv1 SAs. That is, the clients with OpenSLP cannot see trees with NDSslp. Similarly, the clients with NDSslp cannot see trees with OpenSLP. For SLPv1 and SLPv2 to interact, you need to configure a DA that is running SLPv2.OES Linux ships with OpenSLP. However, eDirectory installed other Linux platforms, such as Red Hat Linux, might use NDSslp, which is shipped with eDirectory. Because of interoperability issues with the two versions of SLP, a tree advertised via OpenSLP multicast might not be visible to NDSslp and vice versa. To overcome this problem, you need to configure a DA that runs OpenSLP.

23.11 ldif2dib Fails to Open the Error Log File When the DIB Directory Exists In the Custom Path

ldif2dib fails to open the default log file, `ldif2dib.log` when the `dib` directory is relocated to a custom location.

To work around this issue, explicitly provide the log file location by using the `-b` switch.

23.12 eDirectory Server Does Not Start Automatically in the Virtual SLES 10

After adding packages, if you do not configure eDirectory by using YaST, you need to run the following command at the command line.

```
chkconfig -a ndsd
```

23.13 ndsd Does Not Start After a System Crash

In some situations, eDirectory services (`ndsd`) doesn't start after a system crash or a power failure. To start the eDirectory again, do the following:

1. Delete `/var/opt/novell/eDirectory/data/ndsd.pid` file.
2. Enter `/etc/init.d/ndsd start` command.

23.14 Do not Execute DSTrace With All Tags Enabled on Linux Computers

With all tags enabled, ensure you do not run DSTrace on the following:

- **A loaded system in Journal mode:** It tends to build up ndsd memory.
- **Servers in inline mode:** It crashes ndsd.

23.15 LDAP is Not RFC Compliant For Anonymous Search Requests

If a client performs an unauthenticated search operation when anonymous binds are disabled, the LDAP server responds with the bind result of inappropriate authentication instead of the search result, `operationsError`.

23.16 Troubleshooting Ports with Custom eDirectory 8.8 Instances

In eDirectory 8.8, if you configure a new instance in a custom location when the default instance server is down, it takes the default instance ports. The default instance does not come up, because the ports of the default instance are allotted to the custom location instance.

Follow the procedure in "Troubleshooting Ports with Custom eDirectory 8.8 Instances" (http://www.novell.com/coolsolutions/feature/17933.html) before rebooting the host.

23.17 Rebooting the Host

Only the default instance created through using the default instance binaries is started after reboot.

You can set the paths and use ndsmanage to start the other instances.

23.18 ndsd Not Listening at the Loopback Address on a Given NCP Port

When you have more than one eDirectory instance, the second instance and subsequent instances try to listen at the default 524 port instead of the NCP port on the loopback address.

To work around this issue, set the n4u.server.tcp-port parameter of the second instance to the port that it is supposed to listen on. The n4u.server.tcp-port parameter is located in the `nds.conf` file.

IMPORTANT: All eDirectory instances must be up before upgrading to eDirectory 8.8 SP8.

23.19 LDAP Transaction OIDs

In LDAP transaction support, the `supportedGroupingTypes` and `transactionGroupingType` OIDs are the same (`2.16.840.1.113719.1.27.103.7`).

23.20 Errors -5871 and -5875 in LDAP Trace

The -5871 and -5875 errors in LDAP trace are usually caused when LDAP client closes forcibly without doing an unbind. So, these errors need not be of concern and can be ignored. For more information on these errors, refer to the NetIQ Error Codes Web site (http://www.novell.com/documentation/nwec/).

23.21 NDSCons Gives -625 Error if a Tree is Renamed

If you rename the tree on the primary server and shutdown the DHost on the secondary server, the NDSCons utility gives transport failure error message -625 on the secondary server while DHost keeps running on both primary and secondary servers. The error occurs because NDSCons was running on secondary server when the tree was renamed on the primary server. NDSCons works fine if you close it and then restart it.

23.22 Listening on Multiple NICs Slows Down eDirectory ldapsearch Performance

To work around this issue,

Disable the NICs in the configuration file that slow down the ldapsearch performance.

or

Enable Advanced Referral Costing (ARC) by using the `set NDSTRACE =!ARC1` command in DSTrace.

23.23 Unable to Limit the Number of Concurrent Users on Linux Platforms

In eDirectory 8.8 SP8, you cannot limit the number of concurrent connections on Linux platforms. To resort to the old behavior (strict port-based checking), set following parameter in the `nds.conf` file.

```
n4u.server.mask-port-number=0
```

23.24 ndsd Fails to Shut Down Due to SLP

If you do not have an SLP Directory Agent (DA) configured on your network, finding services that use SLP may take a longer time. During eDirectory shutdown, ndsd tries to perform operations using SLP that may take a long time than the init script normally allows, thus causing a forced shutdown.

To workaround this issue:

1. Create an empty file with the name `hosts.nds` in the config directory. The config directory of a server can be obtained by running the following command `ndsconfig get n4u.server.confdir`
2. Set an environment variable NDS_USESLP to 0 by specifying export NDS_USESLP=0 in `/opt/novell/eDirectory/sbin/pre_ndsd_start`
3. Restart eDirectory.

23.25 Restarting NLDAP on Windows

After NLDAP is stopped, you need to restart the server to load NLDAP.

23.26 SecretStore over LDAP

The NetIQ SecretStore functionality does not work over LDAP. To resolve this, you need to refresh LDAP through iManager.

23.27 Interoperability Issues

- Section 23.27.1, “Cannot Change the Passphrase after Unlocking SecretStore,” on page 126
- Section 23.27.2, “User Credentials Modified through SecretStore Are Reset to Null,” on page 126
- Section 23.27.3, “Creating a Different Credential Set with the Same User Overwrites the Previous Credential Set,” on page 126

23.27.1 Cannot Change the Passphrase after Unlocking SecretStore

SecretStore locks if you try to retrieve a forgotten password by logging in with user credentials and a wrong passphrase. You can unlock SecretStore with administrator rights, and the NetIQ SecureLogin client allows you to log in without a passphrase. If you try changing the passphrase, the login fails and returns an error.

23.27.2 User Credentials Modified through SecretStore Are Reset to Null

When you try saving new credentials in SecretStore by using the iManager plug-in, a blank credential column displays because iManager fails to save the changes.

You can change the credentials from the SecretStore iManager plug-in only by logging in as a user instead of an administrator.

23.27.3 Creating a Different Credential Set with the Same User Overwrites the Previous Credential Set

When you save an alternate credential set, SecretStore fails to retain the first set and only the latest credential set is visible.

You can change the credentials from the SecretStore iManager plug-in only by logging in as a user instead of an administrator.

24 IPV6

This section includes information for troubleshooting IPv6 issues on all platforms.

- Section 24.1, “The LDAP Secure Search Works Either With IPv4 or IPv6 But Not With Both,” on page 127
- Section 24.2, “ICE Plug-In Does Not Work for the IPV6 Addresses,” on page 127
- Section 24.3, “Listeners for Unspecified IPv6 Addresses in Linux and Windows,” on page 128

24.1 The LDAP Secure Search Works Either With IPv4 or IPv6 But Not With Both

The LDAP secure search fails if the client address has both IPv4 and IPv6 addresses.

24.2 ICE Plug-In Does Not Work for the IPV6 Addresses

It fails to connect to the requested server if iManager is listening only on IPv4 address with the following error:

```
Unable to connect to the requested server. Verify the name/address and port.
```

To configure IPv6 for iManager to work with eDirectory, you must enable IPv6 using the following steps:

1. Set the following properties in the `catalina.properties` file and restart Tomcat.

   ```
   java.net.preferIPv4Stack=false
   ```

   ```
   java.net.preferIPv4Addresses=true
   ```

   Note that `java.net.preferIPv4Stack` applies for iManager to work with eDirectory and `java.net.preferIPv4Addresses` applies for browsers to work with iManager.

2. Go to *LDAP options* > *View LDAP Server* > *Connections* > *LDAP Server*, then add LDAP interfaces for the IPv6 addresses with port numbers.

   ```
   ldap://[xx::xx]:389
   ldaps://[xx::xx]:636
   ```

3. Configure the Role-Based Services, then log out from the session and log in again.

24.3 Listeners for Unspecified IPv6 Addresses in Linux and Windows

A listener for an unspecified IPv6 address accepts both IPv4 and IPv6 connections on Linux. Because of this behavior, Linux does not allow you to start both IPv4 and IPv6 unspecified listeners for the same port at the same time. Therefore; if a listener is already configured for an unspecified IPv6 address, the listener on the unspecified IPv4 address fails to start. Linux uses an unspecified address for LDAP listeners.

NOTE: On a SLES 10 computer, if an IPv4 unspecified listener is already there, then IPv6-specific IP listeners for the same port does not start. This is a known issue with SLES 10. However, SLES 11 does not have this issue.

On Windows, an unspecified IPv6 listener accepts only IPv6 connections. Therefore, you must configure a separate IPv4 listener to accept IPv4 connections along with IPv6 connections.

By default, both IPv4 and IPv6 listeners are configured for ldapInterfaces. Depending on the platform, ldapInterfaces starts the required listeners.