

Novell Identity Manager Driver for eDirectory™

3.5.1

www.novell.com

IMPLEMENTATION GUIDE

September 28, 2007



Novell®

Legal Notices

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

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

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

Copyright © 2007 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

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

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

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

Novell Trademarks

For a list of Novell trademarks, see [Trademarks \(http://www.novell.com/company/legal/trademarks/tmlist.html\)](http://www.novell.com/company/legal/trademarks/tmlist.html).

Third-Party Materials

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

Contents

About This Guide	9
1 Overview	11
1.1 What's New	11
1.2 Driver Concepts	11
1.2.1 Key Terms	11
1.2.2 How the eDirectory Driver Works	12
1.3 Driver Features	12
1.3.1 Local Platforms	12
1.3.2 Remote Platforms	12
1.3.3 Role-Based Entitlements	12
1.3.4 Password Synchronization	13
1.3.5 Synchronizing Data	13
2 Installing the eDirectory Driver	15
2.1 Driver Prerequisites	15
2.2 Where to Install the Driver	15
2.3 Installing the Driver	15
3 Upgrading the eDirectory Driver	17
3.1 Upgrading the Driver by Using Designer	17
3.2 Upgrading the Driver by Using iManager	20
3.3 Upgrading from DirXML 1.x	21
3.4 Upgrade Issues for the eDirectory Driver	21
4 Importing the Example Driver Configuration File	23
4.1 Using Designer to Import	23
4.2 Using iManager to Import	25
5 Configuring the Driver	27
5.1 Configuring Secure Identity Manager Data Transfers	27
5.1.1 Understanding eDirectory Driver Security	27
5.1.2 Setting Up a KMO	28
5.2 Configuring Driver Object Properties	29
5.2.1 Authentication Parameters	29
5.3 Configuring the Filter	31
5.4 Configuring Rules on the Publisher Channel	32
5.5 Using Driver Object Passwords	32
5.6 Migrating or Copying Objects	33

6	Activating the eDirectory Driver	35
7	Managing the eDirectory Driver	37
7.1	Starting, Stopping, or Restarting the eDirectory Driver	37
7.2	Migrating and Resynchronizing Data	37
7.3	Password Synchronization	38
7.4	Which Attributes Are Synchronized	39
7.5	Using the DirXML Command Line Utility	40
7.6	Viewing Driver Version Information	40
7.6.1	Viewing a Hierarchical Display of Version Information	40
7.6.2	Viewing the Version Information As a Text File	42
7.6.3	Saving Version Information	44
7.7	Reassociating a Driver Set Object with a Server Object	45
7.8	Changing the Driver Configuration	46
7.9	Storing Driver Passwords Securely with Named Passwords	46
7.9.1	Using Designer to Configure Named Passwords	47
7.9.2	Using iManager to Configure Named Passwords	47
7.9.3	Using Named Passwords in Driver Policies	49
7.9.4	Configuring Named Passwords by Using the DirXML Command Line Utility	50
7.10	Adding a Driver Heartbeat	53
8	Synchronizing Objects	55
8.1	What Is Synchronization?	55
8.2	When Does Synchronization Occur?	55
8.3	How Does the Metadirectory Engine Decide Which Object to Synchronize?	56
8.4	How Synchronization Works	57
8.4.1	Scenario One	57
8.4.2	Scenario Two	59
8.4.3	Scenario Three	60
9	Troubleshooting the eDirectory Driver	63
9.1	Troubleshooting Driver Processes	63
9.1.1	Viewing Driver Processes	63
9.2	Creating Functionality on the Driver Parameters Page	69
10	Backing Up the eDirectory Driver	71
10.1	Exporting the Driver in Designer	71
10.2	Exporting the Driver in iManager	71
11	Security: Best Practices	73
A	The DirXML Command Line Utility	75
A.1	Interactive Mode	75
A.2	Command Line Mode	84
B	Properties of the eDirectory Driver	89
B.1	Driver Configuration	89
B.1.1	Driver Module	90

B.1.2	Driver Object Password	91
B.1.3	Authentication	91
B.1.4	Startup Option	92
B.1.5	Driver Parameters	93
B.1.6	ECMAScript	95
B.2	Global Configuration Values	95
B.3	Named Passwords	96
B.4	Engine Control Values	97
B.5	Log Level	99
B.6	Driver Image	100
B.7	Security Equals	100
B.8	Filter	100
B.9	Edit Filter XML	101
B.10	Misc	101
B.11	Excluded Users	102
B.12	Driver Manifest	102
B.13	Driver Cache Inspector	103
B.14	Driver Inspector	103
B.15	Server Variables	104

About This Guide

This guide explains how to install and configure the Identity Manager Driver for eDirectory™.

- ◆ Chapter 1, “Overview,” on page 11
- ◆ Chapter 2, “Installing the eDirectory Driver,” on page 15
- ◆ Chapter 3, “Upgrading the eDirectory Driver,” on page 17
- ◆ Chapter 4, “Importing the Example Driver Configuration File,” on page 23
- ◆ Chapter 5, “Configuring the Driver,” on page 27
- ◆ Chapter 6, “Activating the eDirectory Driver,” on page 35
- ◆ Chapter 7, “Managing the eDirectory Driver,” on page 37
- ◆ Chapter 8, “Synchronizing Objects,” on page 55
- ◆ Chapter 9, “Troubleshooting the eDirectory Driver,” on page 63
- ◆ Chapter 10, “Backing Up the eDirectory Driver,” on page 71
- ◆ Chapter 11, “Security: Best Practices,” on page 73
- ◆ Appendix A, “The DirXML Command Line Utility,” on page 75
- ◆ Appendix B, “Properties of the eDirectory Driver,” on page 89

Audience

This guide is for Novell® eDirectory and Identity Manager administrators who are using the Identity Manager Driver for eDirectory.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Use the User Comment feature at the bottom of each page of the online documentation, or go to www.novell.com/documentation/feedback.html and enter your comments there.

Documentation Updates

For the most recent version of this document, see *Identity Manager Driver for eDirectory* in the Identity Manager Drivers section on the [Novell Documentation Web site \(http://www.novell.com/documentation/idm35drivers/index.html\)](http://www.novell.com/documentation/idm35drivers/index.html).

Additional Documentation

For information on Identity Manager and other Identity Manager drivers, see the [Novell Documentation Web site \(http://www.novell.com/documentation/idm35\)](http://www.novell.com/documentation/idm35).

Documentation Conventions

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

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

Overview

1

The Identity Manager Driver for eDirectory™ synchronizes objects and attributes between different eDirectory trees.

This driver is unique among all other Identity Manager drivers. Because you are synchronizing data between eDirectory trees, you will always have two drivers installed, each in its own tree. The driver in one tree communicates with the driver in the other tree.

- ◆ [Section 1.1, “What’s New,” on page 11](#)
- ◆ [Section 1.2, “Driver Concepts,” on page 11](#)
- ◆ [Section 1.3, “Driver Features,” on page 12](#)

1.1 What’s New

The eDirectory driver has no new features for Identity Manager 3.5.1.

For information on what’s new in Identity Manager 3.5.1, see [“What's New in Identity Manager 3.5.1?”](#) in the *Identity Manager 3.5.1 Installation Guide*.

1.2 Driver Concepts

- ◆ [Section 1.2.1, “Key Terms,” on page 11](#)
- ◆ [Section 1.2.2, “How the eDirectory Driver Works,” on page 12](#)

1.2.1 Key Terms

Driver shim: A Java file (`NdsToNds.jar`) loaded directly by Identity Manager. Communicates event changes to be sent from the Identity Manager Driver for eDirectory to an Identity Vault, communicates changes from the Identity Vault to the Identity Manager Driver for eDirectory, and operates as the link that connects the Identity Vault and the Identity Vault Driver object.

Driver: A set of policies, filters, and objects that act as the connector between an Identity Vault and the driver shim.

This software enables an application to publish events from an application to the directory, enables an application to subscribe to events from the directory, and synchronizes data between the directory and applications.

To establish a connection between the Metadirectory engine and an Identity Vault, you specify the driver’s configuration and connection parameters, policies, and filter values.

Driver object: A collection of channels, policies, rules, and filters that connect an application to an Identity Vault that is running Identity Manager.

Each driver performs different tasks. Policies, rules, and filters tell the driver how to manipulate the data to perform those tasks.

The Driver object displays information about the driver's configuration, policies, and filters. This object enables you to manage the driver and provide eDirectory management of the driver shim parameters.

Identity Vault. A hub, with applications and directories publishing their changes to it. The Identity Vault then sends changes to the applications and directories that have subscribed for them. This results in two main flows of data: the Publisher channel and the Subscriber channel.

1.2.2 How the eDirectory Driver Works

Channels, filters, and policies control data flow.

Publisher and Subscriber Channels

The eDirectory driver is installed and configured in two trees. The driver's Publisher channel in TreeA communicates with the driver's Subscriber channel in TreeB. Conversely, the driver's Publisher channel in TreeB communicates with the driver's Subscriber channel in TreeA.

Filters

Identity Manager uses filters to control which objects and attributes are shared. The default filter configurations for the eDirectory driver allow objects and attributes to be shared.

Policies

Identity Manager uses policies to control data synchronization between the eDirectory driver and the Identity Vaults. The eDirectory driver comes with an example configuration file, to set up policies.

1.3 Driver Features

- ◆ [Section 1.3.1, "Local Platforms," on page 12](#)
- ◆ [Section 1.3.2, "Remote Platforms," on page 12](#)
- ◆ [Section 1.3.3, "Role-Based Entitlements," on page 12](#)
- ◆ [Section 1.3.4, "Password Synchronization," on page 13](#)
- ◆ [Section 1.3.5, "Synchronizing Data," on page 13](#)

1.3.1 Local Platforms

The eDirectory driver runs in any Identity Manager 3.5.1 installation. See "[Identity Manager System Components and Requirements](#)" in the *Identity Manager 3.5.1 Installation Guide*.

1.3.2 Remote Platforms

The eDirectory driver supports remote connections without the Remote Loader. The driver does not work in the Remote Loader.

1.3.3 Role-Based Entitlements

The eDirectory driver example configuration does not implement any entitlements.

1.3.4 Password Synchronization

The eDirectory driver supports password synchronization via either synchronization of the Public/Private key pair or synchronization of the distribution password. The method is selected during the configuration import process and is labeled *Password Sync Version*.

1.3.5 Synchronizing Data

The eDirectory driver synchronizes data between two Identity Vaults or trees. The driver can run anywhere that a Metadirectory server is running.

Installing the eDirectory Driver

2

- ♦ Section 2.1, “Driver Prerequisites,” on page 15
- ♦ Section 2.2, “Where to Install the Driver,” on page 15
- ♦ Section 2.3, “Installing the Driver,” on page 15

2.1 Driver Prerequisites

- ❑ Requirements for Identity Manager. See “[Identity Manager Components and System Requirements](#)” in the *Identity Manager 3.5.1 Installation Guide*.
- ❑ The Novell® Certificate Server running on each server that hosts the eDirectory™ driver.
- ❑ A Certificate Authority (CA) so that SSL encryption can work.

2.2 Where to Install the Driver

You install Identity Manager and the eDirectory driver on both of the Novell eDirectory servers and in the trees that you want to synchronize. Therefore, the installation and configuration of the driver must be completed twice—once for the eDirectory driver in TreeA and once for the driver in TreeB.

The eDirectory driver does not use the Remote Loader technology because the driver in one tree communicates directly with the driver in the other tree.

The driver uses Novell Certificate Server™ and a Certificate Authority (CA) to ensure data security. All transactions between trees will be secured through SSL technology. For information on data security, see [Section 5.1, “Configuring Secure Identity Manager Data Transfers,” on page 27](#).

2.3 Installing the Driver

This section assumes that you have already installed the Metadirectory engine (and, most likely, other drivers) on the server and need to install only the eDirectory driver. For information on installing the Metadirectory engine, see “[Installing Identity Manager](#)” in the *Identity Manager 3.5.1 Installation Guide*.

Typically, an Identity Manager installation installs all drivers, including the eDirectory driver, at the same time that the Metadirectory engine is installed. If the eDirectory driver wasn’t installed at that time, you can install the driver separately. The schema won’t be extended during this driver install because the Identity Manager installation already extended it when the Metadirectory engine was installed.

If you don’t have a CD, download the file that you need for your platform (for example, `Identity_Manager_3_5_1_NW_Win.iso`) and create one. Downloads are available from [Novell Downloads \(http://download.novell.com/index.jsp\)](http://download.novell.com/index.jsp).

IMPORTANT: Because you are installing the driver on two separate Identity Vault (eDirectory) servers, you must complete procedures for each server.

During the installation, `NdsToNds.jar` is copied to the appropriate directory. The following table shows these locations per platform:

Operating System	Directory
Linux*, Solaris*, or AIX*	<code>/usr/lib/dirxml/classes</code> (For eDirectory 8.8: <code>opt/novell/eDirectory/lib/dirxml/classes</code>)
NetWare®	<code>sys:system\lib</code>
Windows* NT*/2000	The default is <code>novell\nds</code> , but you can specify any directory.

After the installation program ends, configure security as explained in [Section 5.1, “Configuring Secure Identity Manager Data Transfers,”](#) on page 27.

Table 2-1 *Where to Find Install Instructions*

Platform	Where to Find Install Instructions
Linux	See “Installing Identity Manager through the GUI Interface on UNIX/Linux Platforms” in the <i>Identity Manager 3.5.1 Installation Guide</i>
NetWare	See “Installing Identity Manager on NetWare” in the <i>Identity Manager 3.5.1 Installation Guide</i>
Windows	See “Installing Identity Manager on Windows” in the <i>Identity Manager 3.5.1 Installation Guide</i> .

Upgrading the eDirectory Driver

3

If you are upgrading from Identity Manager 3.5.0 to Identity Manager 3.5.1, skip this section (Upgrading the eDirectory Driver).

If you have been using a previous version of the driver, follow the instructions in this section instead of the instructions in [Chapter 2, “Installing the eDirectory Driver,”](#) on page 15.

Identity Manager 3.5.1 contains a new architecture for how policies reference one another. To take advantage of this new architecture, the driver configuration file provided for LDAP must be upgraded. For more information on the new architecture, see “[Upgrading Identity Manager Policies](#)” in *Understanding Policies for Identity Manager 3.5.1*.

You can upgrade by using either Designer for Identity Manager or iManager.

- [Section 3.1, “Upgrading the Driver by Using Designer,”](#) on page 17
- [Section 3.2, “Upgrading the Driver by Using iManager,”](#) on page 20
- [Section 3.3, “Upgrading from DirXML 1.x,”](#) on page 21
- [Section 3.4, “Upgrade Issues for the eDirectory Driver,”](#) on page 21

3.1 Upgrading the Driver by Using Designer

- 1 Make sure that you have updated your driver with all the patches for the version you are currently running.

We recommend this step for all drivers, to help minimize upgrade issues.

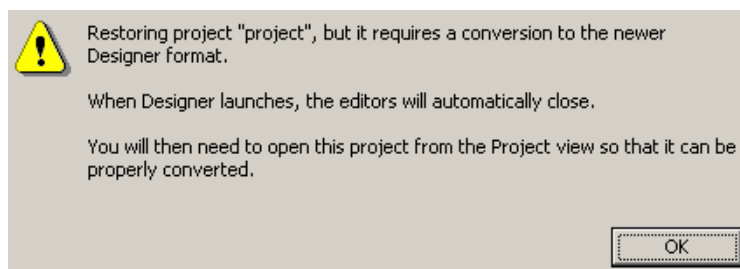
- 2 Back up the driver.

See [Chapter 10, “Backing Up the eDirectory Driver,”](#) on page 71 for instructions on how to back up the driver.

- 3 Install Designer version 2.1 or later, then launch Designer.

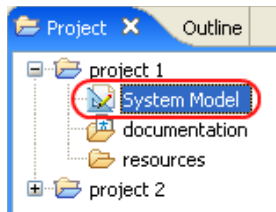
If you had a project open in Designer when you upgraded Designer, proceed to [Step 4](#). If you didn’t have a project open in Designer when you upgraded Designer, skip to [Step 5](#).

- 4 If you had a project open when upgrading Designer, read the warning message, then click *OK*.

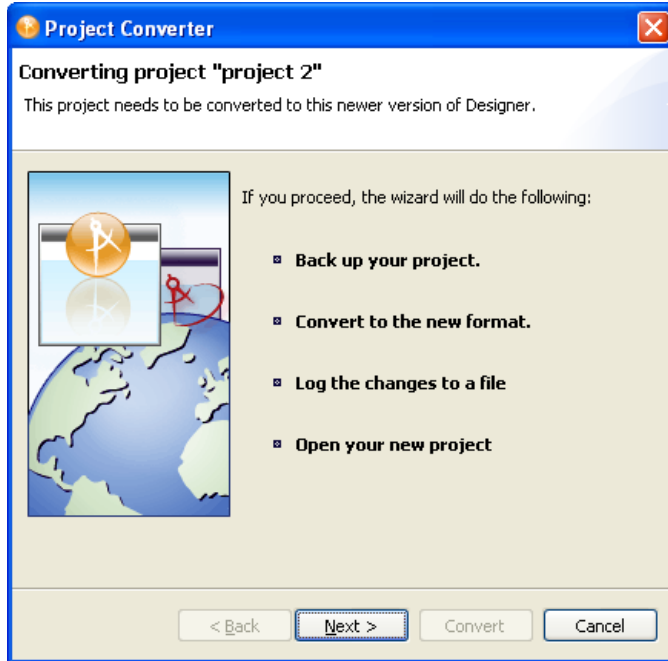


Designer closes the project to preform the upgrade.

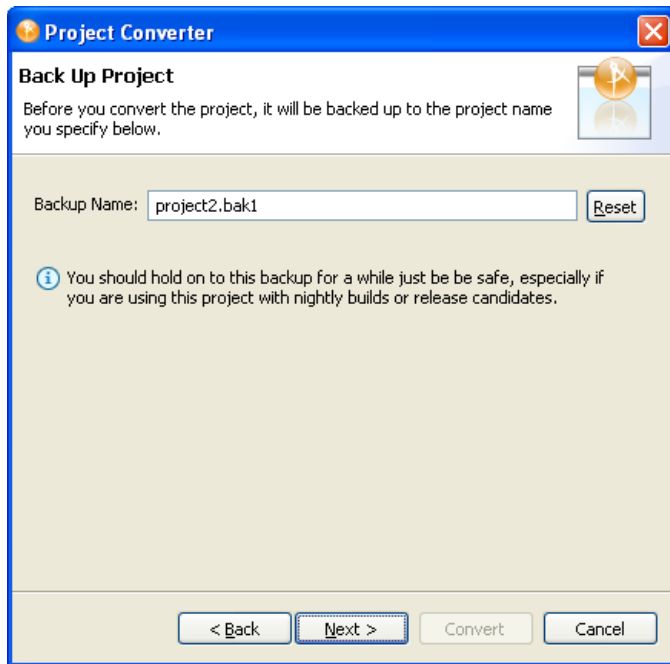
5 to open and convert the project, double-click *System Model* in the Project view.



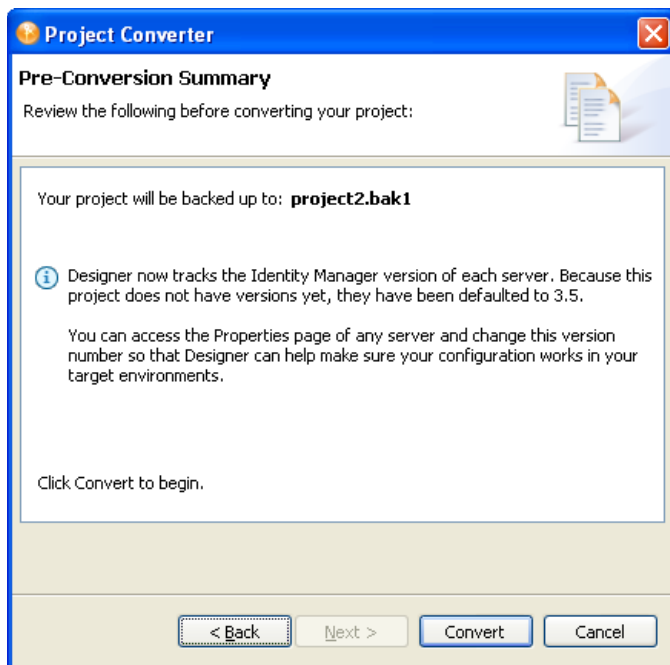
6 Read the tasks listed in the Project Converter message, then click *Next*.



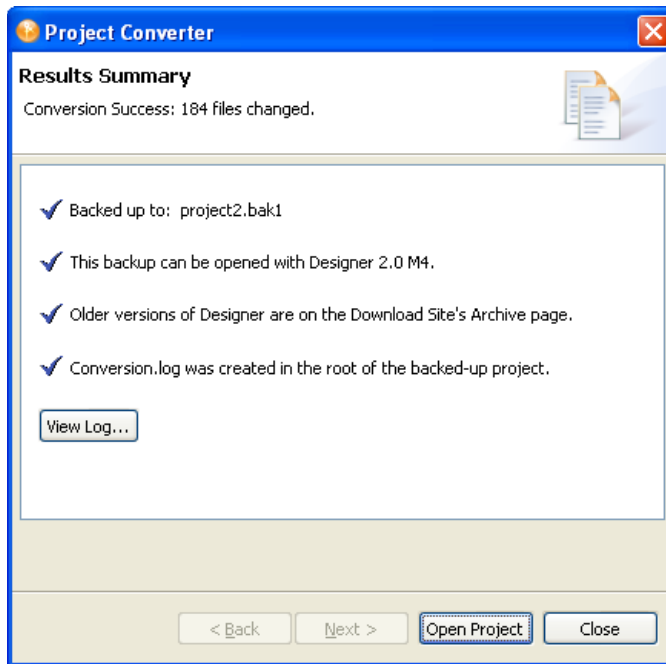
7 Specify the name of the backup project name, then click *Next*.



8 Read the project conversion summary, then click *Convert*.



- 9 Read the project conversion result summary, then click *Open Project*.



To view the log file that is generated, click *View Log*.

3.2 Upgrading the Driver by Using iManager

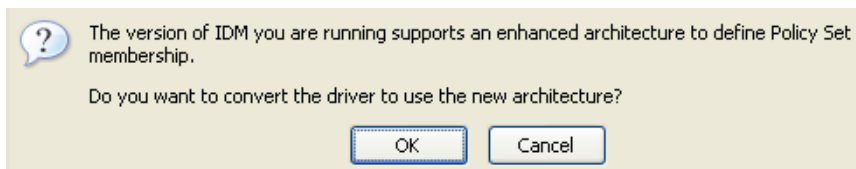
- 1 Make sure that you have updated your driver with all the patches for the version you are currently running.

To help minimize upgrade issues, we recommend that you complete this step on all drivers.

- 2 Back up the driver.

See [Chapter 10, "Backing Up the eDirectory Driver," on page 71](#).

- 3 Verify that Identity Manager 3.5.1 has been installed and that you have the current plug-ins installed.
- 4 Launch iManager.
- 5 Click *Identity Manager > Identity Manager Overview*.
- 6 Click *Search* to find the Driver Set object, then click the driver that you want to upgrade.
- 7 Read the message that is displayed, then click *OK*.



3.3 Upgrading from DirXML 1.x

IMPORTANT: This section applies to upgrading from DirXML[®] 1.x only.

Because you are upgrading the driver on two separate Identity Vault servers, you must complete the upgrade procedures for each server.

Installing the driver shim does not change your existing configuration. Your existing configuration continues to work with the new driver shim.

However, to take advantage of new features, you must upgrade your driver configuration, either by replacing your driver configuration with the new example configuration or by converting your existing configuration to Identity Manager 3.5.1 format and adding policies to it.

- ◆ To replace your existing configuration, import the new sample configuration for your existing driver objects.

The sample configuration contains all the newer features, such as support for Identity Manager Password Synchronization and Role-Based Entitlements.

- ◆ To convert an existing driver configuration so that you can edit it with the new Identity Manager plug-ins, see “[Upgrading a Driver Configuration from DirXML 1.1a to Identity Manager 3.5.1 Format](#)” in the *Novell Identity Manager 3.5.1 Administration Guide*.
- ◆ To add Identity Manager Password Synchronization functionality to an existing driver configuration, seev “[Upgrading Existing Driver Configurations to Support Password Synchronization](#)” in the *Novell Identity Manager 3.5.1 Administration Guide*.

The new policies for password synchronization are intended to support Universal Password and Distribution Password. If you are planning to synchronize only the NDS[®] Password, these policies should not be added to the driver configuration. NDS Password is synchronized by using Public Key and Private Key attributes instead of these policies.

3.4 Upgrade Issues for the eDirectory Driver

IMPORTANT: This section applies to upgrading from DirXML 1.x only.

If you are upgrading Identity Manager and the eDirectory driver, you might encounter data synchronization errors if your certificates have expired (or if one of the two certificates has expired).

If you create a user on the server that holds a valid certificate, the user won't be synchronized to the server containing the invalid certificate. Also, you might see the following error in DSTrace:

```
SSL handshake failed, X509_V_CERT_HAS_EXPIRED  
SSL handshake failed, SSL_ERROR_ZERO_RETURN,
```

If you create a user on the server that holds an expired certificate, the user is still synchronized to the server containing a valid certificate. Also, you might see the following error in DSTrace:

```
Error: 14094415: SSL Routines: SSL_READ_BYTES: sslv3 alert certificate  
expired.
```

To fix this issue, create new certificates.

Importing the Example Driver Configuration File

The Identity Manager Driver for LDAP includes an example configuration file that you can use as a starting point for creating the Driver object. When you import this file, Designer for Identity Manager or iManager creates and configures the objects and policies needed to make the driver work properly.

- ◆ [Section 4.1, “Using Designer to Import,” on page 23](#)
- ◆ [Section 4.2, “Using iManager to Import,” on page 25](#)

4.1 Using Designer to Import

You can import the basic driver configuration file for the eDirectory driver by using Designer. This basic file creates and configures the objects and policies needed to make the driver work properly.

You can create a driver or import the `eDirectory-IDM3_5_0-V1.xml` configuration file onto an existing driver. The following procedure explains one of several ways to import the sample configuration file:

- 1 Open a project in Designer.
- 2 In the Modeler, right-click the Driver Set object, then select *New > Driver*.
- 3 From the drop-down list, select *eDirectory*, then click *Run*.
- 4 Configure the driver by filling in the fields.
Specify information specific to your environment. For information on settings, see [Table 4-1 on page 23](#).
- 5 After specifying parameters, click *OK* to import the driver.
- 6 Customize and test the driver.
- 7 Deploy the driver into the Identity Vault.
See [“Deploying a Driver to an Identity Vault”](#) in the *Designer 2.1 for Identity Manager 3.5.1* guide.

Table 4-1 Settings for the eDirectory Driver

Item	Description
<i>Driver Name</i>	The object name to be assigned to this driver, or the existing driver for which you want to update the configuration.
<i>Remote Tree Address and Port</i>	Specify the DNS host name or IP address and port of the Identity Manager server in the remote tree. For example: 151.155.144.23:8196 hostname:8196

Item	Description
<i>Configure Data Flow</i>	<p>Bidirectional: Both eDirectory™ trees are authoritative sources of the data synchronized between them.</p> <p>Authoritative: The local tree is the authoritative source.</p> <p>Subordinate: The local tree is not an authoritative source.</p>
<i>Configuration Option</i>	<p>Mirrored: Synchronizes objects hierarchically between the local and remote trees.</p> <p>If you choose this option, use the same option for configuring both eDirectory trees you are synchronizing.</p> <p>This option in the driver configuration synchronizes User, Group, Organization, Country, and Organizational Unit objects. It also mirrors the structure of a subtree in the other tree.</p> <p>Flat: Synchronizes all Users and Groups into specific containers.</p> <p>This option synchronizes User and Group objects and places all users in one container and all groups in another container.</p> <p>This option is typically used in conjunction with the Department option (or a similar configuration) in the other tree.</p> <p>This option doesn't create the containers that hold the users and groups. You must create those manually.</p> <p>Department: Synchronize Users and Groups by department (OU).</p> <p>This option synchronizes User and Group objects and places all users and groups in a container based on the Department field in your management console.</p> <p>This configuration is typically used in conjunction with the Flat option (or a similar configuration) in the other tree.</p> <p>This option doesn't create the containers for each department. You must create those manually. They must be the same as the container specified during import.</p>
<i>Base Container</i>	<p>Used for Mirrored, Flat, and Department options.</p> <p>Specify the base container for synchronization in the local tree, for example Users.MyOrganization.</p> <p>If using with Mirrored: The local base container to mirror with the Remote Base Container above.</p> <p>If using with Flat: The container to place Users into.</p> <p>If using with Department: The parent of the departmental containers.</p>
<i>Password Sync Version</i>	<p>Determines whether the driver synchronizes by using public/private keys (for DirXML 1.0) or the distribution password and password policies (for Identity Manager 2.1 or later).</p>
<i>Password Failure Notification User</i>	<p>Specifies which user receives e-mail notifications when password updates fail.</p>

Item	Description
<i>Remote Base Container</i>	Used for Mirrored option only. Specify the base container for synchronization in the remote tree, for example Users.MyOrganization.
<i>Group Container</i>	Used for Flat only. Specify the base container for synchronization in the local tree to place Groups into, for example Groups.MyOrganization.



4.2 Using iManager to Import

Identity Manager provides an example configuration file (`eDirectory-IDM3_5_0-V1.xml`). You installed this file when you installed the Identity Manager Web components on an iManager server. Think of the example configuration file as a template that you import and customize or configure for your environment.

- 1 In iManager, select *Identity Manager Utilities > Import Configurations*.
- 2 Select a driver set, then click *Next*.

Where do you want to place the new drivers?

In an existing driver set

In a new driver set

If you place this driver in a new driver set, you must specify a driver set name, context, and associated server.

- 3 Select *eDirectory-IDM3_5_0-V1.xml*, then click *Next*.
- 4 Configure the driver by filling in the configuration parameters.
For information on the settings, see [Table 4-1 on page 23](#).
- 5 Define security equivalences by using a user object that has the rights that the driver needs to have on the server

The Admin user object is most often used for this task. However, you might want to create a DriversUser (for example) and assign security equivalence to that user. Whatever rights that the driver needs to have on the server, the DriversUser object must have the same security rights.

- 6 Identify all objects that represent administrative roles and exclude them from replication.
Exclude the security-equivalence object (for example, DriversUser) that you specified in [Step 5](#). If you delete the security-equivalence object, you have removed the rights from the driver. Therefore, the driver can't make changes to Identity Manager.
- 7 Click *Finish*.

Configuring the Driver

5

- ◆ Section 5.1, “Configuring Secure Identity Manager Data Transfers,” on page 27
- ◆ Section 5.2, “Configuring Driver Object Properties,” on page 29
- ◆ Section 5.3, “Configuring the Filter,” on page 31
- ◆ Section 5.4, “Configuring Rules on the Publisher Channel,” on page 32
- ◆ Section 5.5, “Using Driver Object Passwords,” on page 32
- ◆ Section 5.6, “Migrating or Copying Objects,” on page 33

5.1 Configuring Secure Identity Manager Data Transfers

All eDirectory driver communication is secured through SSL. To configure your eDirectory system to handle secure Identity Manager data transfers, run the NDS2NDS wizard in Novell iManager.

- ◆ Section 5.1.1, “Understanding eDirectory Driver Security,” on page 27
- ◆ Section 5.1.2, “Setting Up a KMO,” on page 28

5.1.1 Understanding eDirectory Driver Security

The following items can help you understand eDirectory driver security:

- ◆ The driver uses SSL sockets to provide authentication and a secure connection. SSL uses digital certificates to allow the parties to an SSL connection to authenticate one another. Identity Manager in turn uses Novell Certificate Server certificates for secure management of sensitive data.
- ◆ To use the driver, you must have the Novell Certificate Server running in each tree. We recommend that you use the Certificate Authority from one of the trees containing the driver to issue the certificates used for SSL. If your tree does not have a Certificate Authority, you need to create one. You can use an external Certificate Authority.
- ◆ The Novell implementation of SSL that the driver uses is based on Novell Secure Authentication Services (SAS) for eDirectory and NTLS for eDirectory 8.7.x. These must be installed and configured on the server where the driver runs. eDirectory usually does this automatically.
- ◆ To configure driver security, it is necessary to create and reference certificates in the eDirectory trees that will be connected using the driver. Certificate objects in eDirectory are called Key Material Objects (KMOs) because they securely contain both the certificate data (including the public key) and the private key associated with the certificate.

A minimum of two KMOs (one KMO per tree) must be created for use with the Identity Manager Driver for eDirectory. This section explains using a single KMO per tree.

The NDS2NDS Driver Certificate Wizard sets up the KMOs.

- ◆ For more information:
 - ◆ For an overview of Novell Certificate Server, see the [Novell Certificate Server online documentation \(http://www.novell.com/documentation/crtsrv20/index.html\)](http://www.novell.com/documentation/crtsrv20/index.html).

- ◆ For more information on CAs, and in particular for information about setting up Certificate Authorities in your trees, see [Setting Up Novell PKI Services \(http://www.novell.com/documentation/lg/ndsse/ndsseenu/data/h6172k4q.html\)](http://www.novell.com/documentation/lg/ndsse/ndsseenu/data/h6172k4q.html).

5.1.2 Setting Up a KMO

To configure your Identity Vault system to handle secure Identity Manager data transfers:

- 1 Find out the tree name or IP address of the destination server.
- 2 Launch iManager and authenticate to your first tree.
- 3 Click *Identity Manager Utilities > NDS2NDS Driver Certificates*.
- 4 At the Welcome page, enter the requested information for the first tree.

Default values are provided using objects in the tree that you authenticated to when you launched iManager. You must enter or confirm the following information:

- ◆ Driver DN: Type the distinguished name of the eDirectory driver (for example, EDir-Workforce.Employee Provisioning.Services.YourOrgName).
- ◆ The tree name: Specify the IP address for the Workforce Tree.
- ◆ A username for an account with Admin privileges (for example, Admin).
- ◆ The password for the user.
- ◆ The user's context (for example Services.YourOrgName).

- 5 Click *Next*.

The wizard uses the information you entered to authenticate to the first tree, verify the driver DN, and verify that the driver is associated with a server.

- 6 Specify the requested information for the second tree.

At the Welcome page, enter the requested information for the first tree.

Specify or confirm the following information:

- ◆ Driver DN: Type the distinguished name of the eDirectory driver (for example, EDir-Account.DriverSet.YourOrgName).
- ◆ The tree name: Type the tree name or IP address for the Account Tree.
- ◆ A username for an account with Admin privileges (for example, Admin).
- ◆ The password for the user.
- ◆ The user's context (for example, London.YourOrgName).

- 7 Click *Next*.

The wizard uses the information you entered to authenticate to the second tree, verify the driver DN, and verify that the driver is associated with a server.

- 8 Review the information on the Summary Page, then click *Finish*.

If KMOs already existed for these trees, the wizard deletes them and then does the following:

- ◆ Exports the trusted root of the CA in the first tree.
- ◆ Creates KMO objects.
- ◆ Issues a certificate signing request.
- ◆ Places certificate key pair names in the drivers' Authentication IDs.

5.2 Configuring Driver Object Properties

Typically, the driver's properties are automatically configured when you import the driver configuration file and run the Certificate Wizard.

To configure properties manually:

- 1 In iManager, click *Identity Manager > Identity Manager Overview*.
- 2 Locate the driver set that contains the eDirectory™ driver, then click the driver's icon.
- 3 From the *Identity Manager Driver Overview* page, click the eDirectory Driver object, which displays the driver configurations.
- 4 Locate the *Driver Module* section, then select *Java*.

Driver Module

- Java
 Native
 Connect to Remote Loader

Name:

com.vmp.nds.dirxml.driver.nds.DriverShimImpl

- 5 In the Name edit box, type the following eDirectory Driver Java class name:
`com.novell.nds.dirxml.driver.nds.DriverShimImpl`
- 6 Set parameters.

5.2.1 Authentication Parameters

Figure 5-1 Authentication Parameters

Authentication

SW3K-NDS.Novell

Authentication ID:	<input type="text"/>
Authentication context:	<input type="text" value="187.168.11.8196"/>
Remote loader connection parameters:	<input type="text" value="undefined"/>
Driver cache limit (kilobytes):	<input type="text" value="0"/>
Enter the application password:	<input type="password"/>
Reenter the application password:	<input type="password"/>
Enter the remote loader password:	<input type="password"/>
Reenter the remote loader password:	<input type="password"/>

Remove existing password

Provide information that allows the source server to communicate with the destination server.

Authentication ID

If you want the source server and destination server to exchange secure information (for example, passwords), run the NDS2NDS eDirectory Certificates Wizard. This wizard creates Key Material Objects (KMOs) and places the correct KMO name in the Authentication ID field.

The KMOs are Secure Socket Layer (SSL) certificates:

Figure 5-2 *SSL Certificates*



Authentication Context

In the Authentication Context field, enter the host name or IP address of the destination server as well as the decimal port number (for example, 187.168.1.1:8196).

You can specify a separate port for Subscriber and Publisher channels by specifying a second port number following a second colon. If a second port number is specified, the Publisher channel uses the second port number rather than using the same port number as the Subscriber channel (for example, 255.255.255.255:2000:2001).

If your server has multiple IP addresses, you can specify the IP address you want the Publisher channel to use. This requires specifying the remote IP address, the Subscriber channel port, the local IP address, and the Publisher channel port. For example, 137.65.134.81:2000:137.65.134.83:2000 specifies that the Subscriber channel will communicate with the remote tree on 137.65.134.81, port 2000, and that the Publisher channel will listen on address 137:65.134.83, port 2000.

NOTE: If you see “java.net.ConnectException: Connection Refused,” no port connection is available on the remote side. This error might be caused by one of the following:

- ◆ The driver on the remote side is not running.
 - ◆ The driver is running but is configured to use a different port.
-

Remote Loader Connection Parameters

The Remote Loader option isn’t needed (and isn’t used) for the Identity Manager Driver for eDirectory.

Driver Cache Limit

Don’t modify this field unless Novell Support asks you to do so.

Enter the application password

The application password on the eDirectory driver must match the Driver Object password of the driver in the other tree.

Scenario—Application Passwords: Server1 is in TreeA. Server 1 is running Identity Manager and the eDirectory driver. Server2 is in TreeB. Server2 is also running Identity Manager and the eDirectory driver. To Server1, the application is the eDirectory driver running on Server2. To

Server2, the application is the eDirectory driver running on Server1. The application password on Server1 is the same as the Driver Object password on Server2.

A Best Practice tip is to set Driver Object password on the eDirectory driver and the application password in the corresponding driver in a similar relationship to the Driver Object password and the Remote Loader password when using a Remote Loader.

Remove existing password

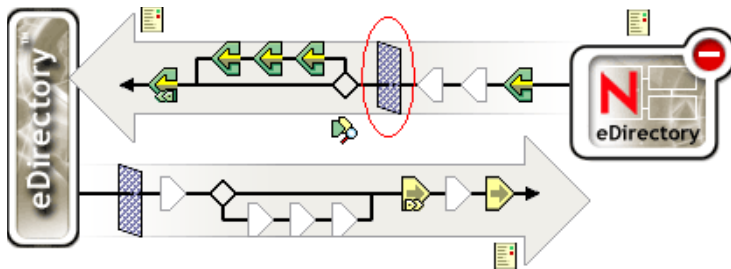
Remove existing password pertains to the application password. This option is rarely used. However, you might accidentally enter the application password when the driver in the other tree does not have a Driver Object password set. Or you might point the driver to a different driver in the other tree, a driver that doesn't have a Driver Object password set.

After a password is set it cannot be removed, only changed. If you set the Driver Object password in TreeA, you would thereafter need an application password on the driver in TreeB.

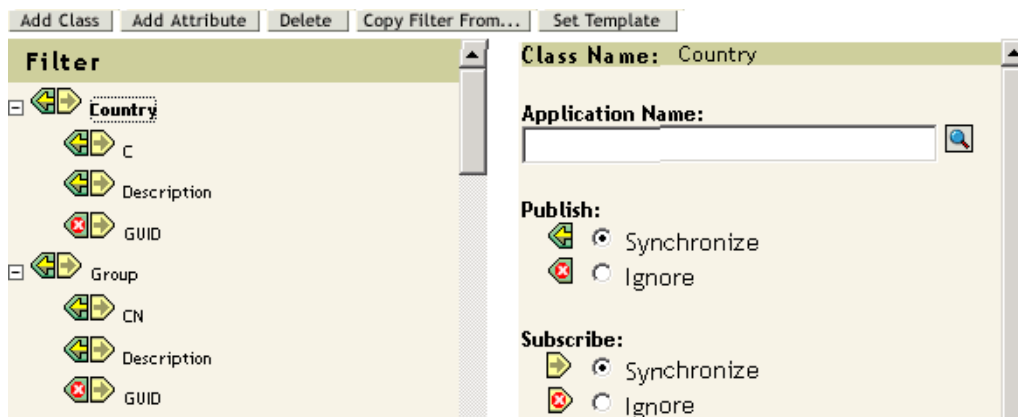
5.3 Configuring the Filter

One filter controls both the Publisher and the Subscriber channels. You should modify the filter to include object classes and attributes you want to be available for Identity Manager processing. To modify the filter:

- 1 In iManager, click *Identity Manager > Identity Manager Overview*.
- 2 Locate the driver set containing the eDirectory driver, then click the driver's icon to display the *Identity Manager Driver Overview* page.
- 3 Click the filter on the Publisher channel.



- 4 Customize the driver.



In this example, Country and Group are classes. To add a class, click *Add Class*, then select the class. To delete a class, select it, then click *Delete*.

In this example, CN is an attribute of the Group class. To add an attribute, select the class, click *Add Attribute*, then select the attribute.

To modify a class or attribute, select it, then select options in the right pane. In this example, the Country attribute is synchronized on the Publisher and Subscriber channels. However, the GUID attribute isn't synchronized on the Publisher channel.

To synchronize the GUID attribute, select it, then click *Synchronize* in the *Publish* section.

The GUID attribute is required for all classes that are set to Synchronize on the Subscriber channel.

In general, except for the GUID attribute, the Subscriber channel filter in one tree should match the Publisher channel filter in the other tree, and vice versa.

- 5 Click *Apply*, then click *OK*.

5.4 Configuring Rules on the Publisher Channel

The rules on a driver should generally be placed only on the Publisher object, not on the Subscriber object. The Matching and Placement policies cannot operate correctly on the Subscriber channel because the Subscriber channel is acting primarily as a source of events for the Publisher channel of the other tree.

It is sometimes desirable to place an Event Transform or Create Policy on the Subscriber channel in order to prevent sending unnecessary data across the channel. See “[Using Scope Filtering to Manage Users on Different Servers](#)” in the *Identity Manager 3.5.1 Installation Guide*.

5.5 Using Driver Object Passwords

In addition to the mandatory certificates needed to use SSL, for additional security you should configure the driver so that the Subscriber channel on one tree authenticates to the Publisher channel on the remote tree. The Driver object password in each tree should be set up to match the application password in the other tree.

To set the Identity Manager Driver object password in a tree:

- 1 In iManager, click *Identity Manager > Identity Manager Overview*.
- 2 Locate the driver set containing the eDirectory driver, then click the driver's icon.
- 3 From the Identity Manager Driver Overview page, click the eDirectory driver object.
- 4 Select *Driver Configuration*.
Select from a drop-down list or tab, depending on the iManager version and your environment.
- 5 Locate the *Driver Object Password* section.

Driver Object Password

Enter password:
Reenter password:

- 6 Type a Driver object password.

IMPORTANT: After a Driver object password is set, it can't be removed.

- 7 In the *Authentication* section, type the application password.

Authentication

5W3K-NDS.Novell

Authentication ID:	<input type="text"/>
Authentication context:	<input type="text" value="187.168.1.18196"/>
Remote loader connection parameters:	<input type="text" value="undefined"/>
Driver cache limit (kilobytes):	<input type="text" value="0"/>
Enter the application password:	<input type="password"/>
Reenter the application password:	<input type="password"/>

- 8 Click *Apply*, then click *OK*.

5.6 Migrating or Copying Objects

Although iManager doesn't have a Copy function, you can use the *Migrate from eDirectory* option to copy objects from one eDirectory tree to another. The scope of the copying depends on the policies of the driver. For example, depending on policies that apply to the driver, you can copy (sync) all the attributes from one eDirectory tree to another. Such a "copy" requires that you synchronize all the attributes across the trees, put objects in the same location during a migration, and not change any data during the migration.

A time stamp is always associated with a resync operation. A resync operation looks for objects that are already associated (have already been synchronized) but have been changed since the time stamp. It also attempts to look for objects that might have been created since the time stamp. Clicking *Resync* might cause new users to be synchronized.

Instead of using the *Resync* option to copy objects, you can use the *Migrate from eDirectory* option. This option enables you to specify and synchronize a list of objects. For each object in the list, iManager writes data to the directory. Identity Manager notes the changes and starts the synchronization process for listed objects.

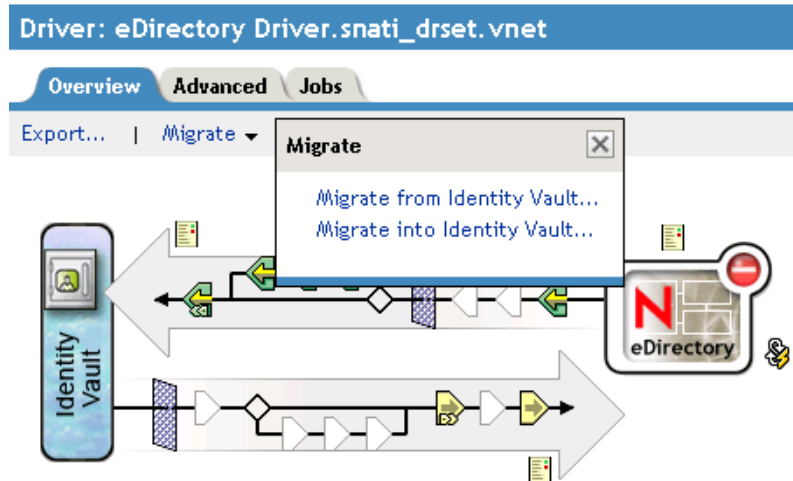
- 1 Make sure that Identity Manager 3.5.1 is installed on a server in the source eDirectory tree and on a server in the destination eDirectory tree.
- 2 Configure an Identity Manager Driver for eDirectory on the server in the source tree.
In the eDirectory driver's Authentication pane, provide the name or IP address and port of the destination server. See [Section 5.2, "Configuring Driver Object Properties," on page 29](#).
Select a migration option: *Flat*, *Mirrored*, or *Department*. To preserve the directory structure (including subcontainers and names) when data is migrated from the source tree to the destination tree, select *Mirrored*.
- 3 Configure an Identity Manager Driver for eDirectory on the server in the destination tree.
In the Authentication pane, provide the name or IP address and port of the source server.
- 4 Set up SSL between the two trees.

Using the NDS2NDS Wizard, create KMO certificates in both trees. See [Section 5.1.2, “Setting Up a KMO,”](#) on page 28.

To launch the NDS2NDS Wizard, in iManager select *Identity Manager Utilities > NDS-to-NDS Driver Certificates*.

- 5 In iManager, select *Identity Manager*, click *Identity Manager Overview*.
- 6 Search for and select a driver set, then click the driver.
- 7 Select *Migrate*, then click a migration option.

Identity Manager Driver Overview



With eDirectory-to-eDirectory migrations, migrate from the source tree to the destination tree.

The *Migrate into Identity Vault* option doesn't work with the Identity Manager Driver for eDirectory.

- 8 Select objects.
For example, select a User object or a Container object. You can search for or browse to the objects. Also, you can add multiple objects.
- 9 Click *OK* twice.
The client (for example, iManager) writes a value to each object in the list. This change event causes Identity Manager to push the data into your destination tree.

Activating the eDirectory Driver

6

Activate the driver within 90 days of installation. Otherwise, the driver will stop running.

For information on activation, refer to “[Activating Novell Identity Manager Products](#)” in the *Identity Manager 3.5.1 Installation Guide*.

- ♦ [Section 7.1, “Starting, Stopping, or Restarting the eDirectory Driver,” on page 37](#)
- ♦ [Section 7.2, “Migrating and Resynchronizing Data,” on page 37](#)
- ♦ [Section 7.3, “Password Synchronization,” on page 38](#)
- ♦ [Section 7.4, “Which Attributes Are Synchronized,” on page 39](#)
- ♦ [Section 7.5, “Using the DirXML Command Line Utility,” on page 40](#)
- ♦ [Section 7.6, “Viewing Driver Version Information,” on page 40](#)
- ♦ [Section 7.7, “Reassociating a Driver Set Object with a Server Object,” on page 45](#)
- ♦ [Section 7.8, “Changing the Driver Configuration,” on page 46](#)
- ♦ [Section 7.9, “Storing Driver Passwords Securely with Named Passwords,” on page 46](#)
- ♦ [Section 7.10, “Adding a Driver Heartbeat,” on page 53](#)

7.1 Starting, Stopping, or Restarting the eDirectory Driver

In Designer:

- 1 Open a project in the Modeler, then right-click the driver icon or driver line.
- 2 Select *Live > Start Driver, Stop Driver, or Restart Driver*.

To start the eDirectory driver in iManager:

- 1 If you changed default data locations during configuration, ensure that the new locations exist before you start the driver.
- 2 Click *Identity Manager > Identity Manager Overview*.
- 3 Browse to the driver set where the driver exists, then click *Search*.
- 4 Click the driver status indicator in the upper right corner of the driver icon, then click *Start driver, Stop driver, or Restart driver*.

If a change log is available, the driver processes all the changes in the change log. To force an initial synchronization, see [“Migrating and Resynchronizing Data” on page 37](#).

7.2 Migrating and Resynchronizing Data

Identity Manager synchronizes data as it changes. If you want to synchronize all data immediately, you can choose from the following options:

- ♦ **Migrate Data from the Identity Vault:** Allows you to select containers or objects you want to migrate from an Identity Vault to an eDirectory server. When you migrate an object, the Metadirectory engine applies all of the Matching, Placement, and Create policies, as well as the Subscriber filter, to the object.
- ♦ **Migrate Data into the Identity Vault:** Allows you to define the criteria that Identity Manager uses to migrate objects from an eDirectory server into an Identity Vault. When you migrate an object, the Metadirectory engine applies all of the Matching, Placement, and Create policies, as

well as the Publisher filter, to the object. Objects are migrated into the Identity Vault by using the order you specify in the Class list.

- ♦ **Synchronize:** Identity Manager looks in the Subscriber class filter and processes all objects for those classes. Associated objects are merged. Unassociated objects are processed as Add events.

To use one of the options:

- 1 In iManager, click *Identity Manager > Identity Manager Overview*.
- 2 Browse to and select the driver set where the driver exists, then click *Search*.
- 3 Click the driver icon.
- 4 Click the appropriate migration button.

For more information, see [Chapter 8, “Synchronizing Objects,” on page 55](#).

7.3 Password Synchronization

This section contains information that is specific to the Identity Manager Driver for eDirectory, and assumes that you are familiar with the information in “[Implementing Password Synchronization](#)” in the *Novell Identity Manager 3.5.1 Administration Guide* [Novell Identity Manager 3.5.1 Administration Guide](#).

- ♦ The driver shim continues to work as in earlier releases. In Identity Manager 2.0, new policies were added to the sample driver configuration to support Identity Manager Password Synchronization, including synchronizing Universal Password.
- ♦ If you decide to enforce password policies in multiple trees, make sure that the Advanced Password Rules in the password policies are compatible in each tree, so that password synchronization can be successful.

If you enforce incompatible password policies in multiple eDirectory trees, and choose to set a password back if it does not comply (with the option *If password does not comply, enforce Password Policy on the connected system by resetting user’s password to the Distribution Password*), you could encounter a loop in which each Identity Vault server tries to change a noncompliant password.

Information about password policies is in “Managing Passwords by Using Password Policies,” in the [Password Management Administration Guide \(http://www.novell.com/documentation/password_management/index.html\)](http://www.novell.com/documentation/password_management/index.html).

- ♦ If the filter for the driver has the setting *Synchronize* for the Public Key and Private Key attributes, the NDS[®] password is synchronized between trees, regardless of any other settings you have created.

If you want to synchronize passwords using Universal Password, make sure you set the filter on both eDirectory drivers to *Ignore* for the Public Key and Private Key attributes for all classes that you want to synchronize Universal Password.

- ♦ To add Identity Manager Password Synchronization functionality to an existing driver configuration, see “[Upgrading Existing Driver Configurations to Support Password Synchronization](#)” in the *Novell Identity Manager 3.5.1 Administration Guide*.

The new policies for password synchronization are intended to support Universal Password and Distribution Password. If you are planning to synchronize only the NDS Password, these policies should not be added to the driver configuration. NDS Password is synchronized by using Public Key and Private Key attributes instead of these policies.

- ◆ The Check Password Status task in iManager does not work for a connected system if the Password Policy has Universal Password enabled and does not have the setting selected for synchronizing Universal Password with NDS Password.

The Check Password Status task lets you see whether a user’s password in Identity Manager is synchronized with the password on connected systems.

If you are using the Identity Manager Driver for eDirectory, and the password policy for a user specifies in the Configuration Options tab that the NDS Password should not be updated when the Universal Password is updated, then the Check Password Status task for that user always shows that the password is not synchronized. The password status is shown as not synchronized, even if the Identity Manager Distribution Password and the Universal Password on the connected system are in fact the same.

This is because the Identity Vault check-password functionality is checking the NDS Password at this time, instead of going through NMAST[™] to refer to the Universal Password.

By default, the NDS Password is updated when the Universal Password is updated in the password policy. If you select this option, Check Password Status should be accurate for the connected system.

- ◆ To use the driver, you must have the Novell[®] Certificate Server[™] running on each server that hosts the driver. You must also create a Certificate Authority (CA) for SSL encryption to work. We recommend that the certificates used for SSL be issued by the Certificate Authority from one of the trees containing the driver. If your tree does not have a Certificate Authority, create one. You can use an external Certificate Authority.

For instructions on creating CAs and configuring the Certificate Server, refer to [Section 5.1, “Configuring Secure Identity Manager Data Transfers,” on page 27](#).

7.4 Which Attributes Are Synchronized

The filter for the sample driver configuration synchronizes the following attributes:

Table 7-1 eDirectory Driver Attributes That Are Synchronized

accessCardNumber	Initials	preferredDeliveryMethod
ACL	instantMessagingID	preferredName
assistant	internationaliSDNNumber	Private Key
assistantPhone	Internet EMail Address	Public Key
businessCategory	jackNumber	registeredAddress
city	jobCode	roomNumber
CN	L	S
co	Language	SA
company	Mailbox ID	Security Equals
costCenter	Mailbox Location	Security Flags
costCenterDescription	mailstop	See Also
departmentNumber	manager	siteLocation

Description	managerWorkforceID	Surname
destinationIndicator	mobile	Telephone Number
directReports	NSCP:employeeNumber	teletexTerminalIdentifier
E-Mail Address	otherPhoneNumber	telexNumber
employeeStatus	O	Timezone
employeeType	OU	Title
Equivalent To Me	pager	tollFreePhoneNumber
Facsimile Telephone Number	personalTitle	UID
Full Name	photo	uniqueID
Generational Qualifier	Physical Delivery Office Name	vehicleInformation
Given Name	Postal Address	workforceID
Group Membership	Postal Code	x121Address
Higher Privileges	Postal Office Box	x500UniqueIdentifier

7.5 Using the DirXML Command Line Utility

The DirXML[®] Command Line utility provides command line access to manage the driver. This utility is not a replacement for iManager or Designer. The primary use of this utility is to allow you to create platform-specific scripts to manage the driver.

For example, you could create a shell script on Linux to check the status of the driver. See [Appendix A, “The DirXML Command Line Utility,” on page 75](#) for information about the DirXML Command Line utility. For daily tasks, use iManager or Designer.

7.6 Viewing Driver Version Information

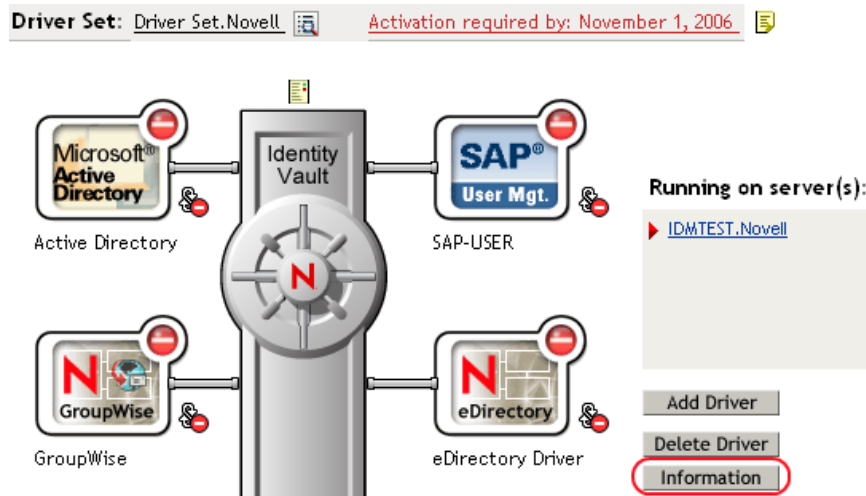
The Versioning Discovery tool exists only in iManager.

- ◆ [Section 7.6.1, “Viewing a Hierarchical Display of Version Information,” on page 40](#)
- ◆ [Section 7.6.2, “Viewing the Version Information As a Text File,” on page 42](#)
- ◆ [Section 7.6.3, “Saving Version Information,” on page 44](#)

7.6.1 Viewing a Hierarchical Display of Version Information

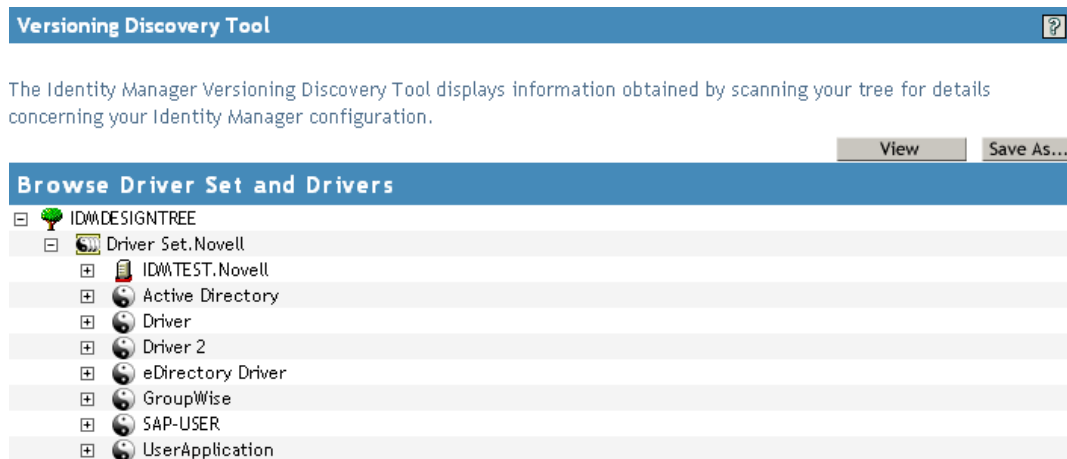
- 1 To find your Driver Set object in iManager, click *Identity Manager > Identity Manager Overview*, then click *Search*.

2 In the Identity Manager Overview, click *Information*.



You can also select *Identity Manager Utilities > Versions Discovery*, browse to and select the Driver Set object, then click *OK*.

3 View a top-level or unexpanded display of version information.



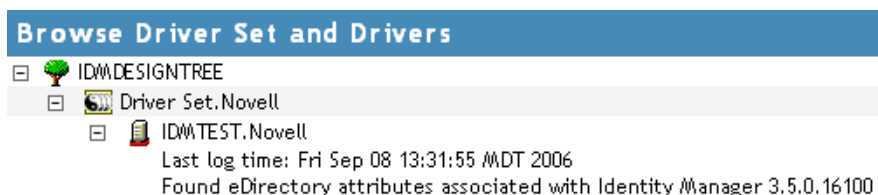
The unexpanded hierarchical view displays the following:

- ◆ The eDirectory™ tree that you are authenticated to
- ◆ The Driver Set object that you selected
- ◆ Servers that are associated with the Driver Set object

If the Driver Set object is associated with two or more servers, you can view Identity Manager information on each server.

- ◆ Drivers

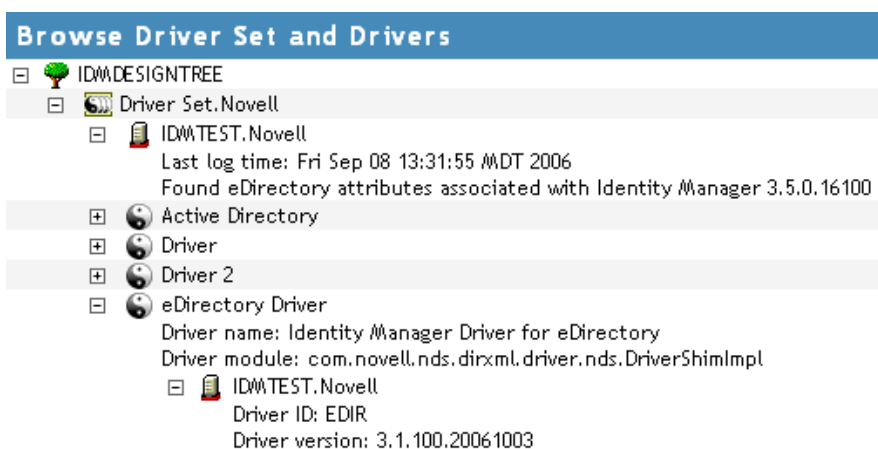
- 4 View version information related to servers by expanding the server icon.



The expanded view of a top-level server icon displays the following:

- ◆ Last log time
- ◆ Version of Identity Manager that is running on the server

- 5 View version information related to drivers by expanding the driver icon.



The expanded view of a top-level driver icon displays the following:

- ◆ The driver name
- ◆ The driver module (for example, com.novell.nds.dirxml.driver.delimitedtext.DelimitedTextDriver)

The expanded view of a server under a driver icon displays the following:

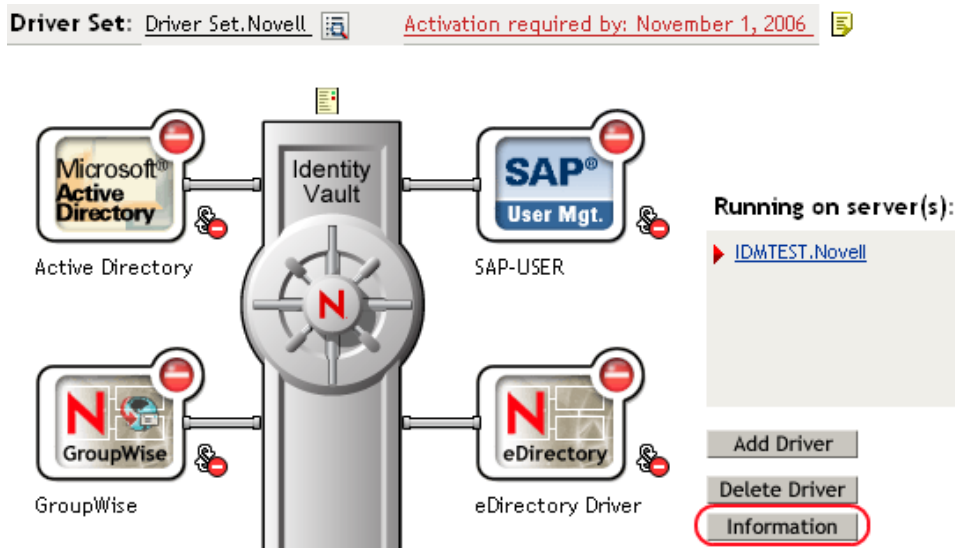
- ◆ The driver ID
- ◆ The version of the instance of the driver running on that server

7.6.2 Viewing the Version Information As a Text File

Identity Manager publishes version information to a file. You can view this information in text format. The textual representation is the same information contained in the hierarchical view.

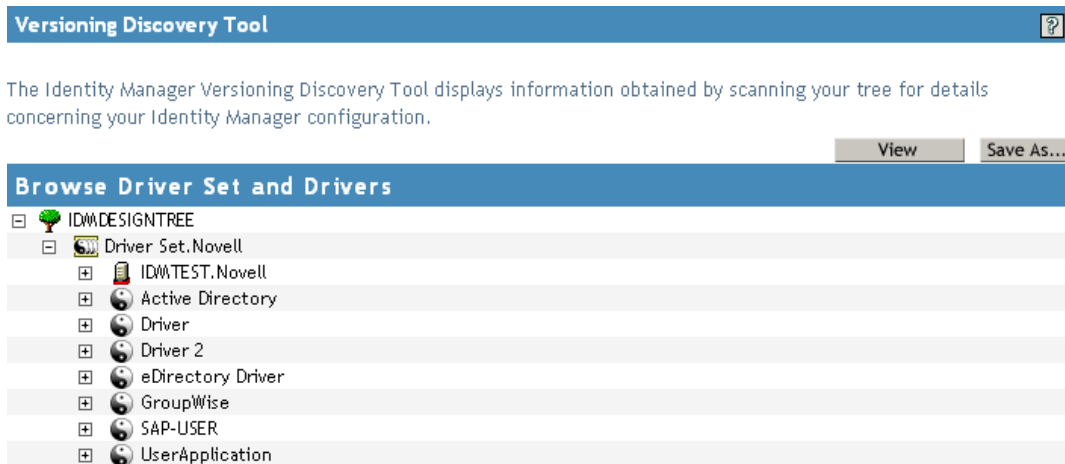
- 1 To find your Driver Set object in iManager, click *Identity Manager > Identity Manager Overview*, then click *Search*.

2 In the Identity Manager Overview, click *Information*.



You can also select *Identity Manager Utilities > Versions Discovery*, browse to and select the Driver Set object, then click *Information*.

3 In the Versioning Discovery Tool dialog box, click *View*.



The information is displayed as a text file in the Report Viewer window.

Versioning Discovery Tool - Report Viewer

```
Identity Manager Version Discovery Tool v2.0
Novell, Inc. Copyright 2003, 2004

Version Query started Saturday, January 20, 2007 11:02:52 AM MST

Parameter Summary:
  Default server's DN:  IDMTEST.Novell
  Default server's IP address:  137.65.151.208
  Logged in as admin, context Novell
  Tree name:  IDMDSIGNTREE
  Found 7 Identity Manager Drivers

Driver Set:  Driver Set.Novell
  Driver Set running on Identity Vault:  IDMTEST.Novell
  Last log time:  Fri Sep 08 13:31:55 MDT 2006
  Found eDirectory attributes associated with Identity Manager 3.5.0.1
Driver:  Active Directory.Driver Set.Novell
  Driver name:  Identity Manager Driver for Active Directory and Excha
  Driver module:  addriver.dll
  Driver Set running on Identity Vault:  IDMTEST.Novell
  Didn't find any DirXML-DriverVersion attributes associated w:
  This may mean the Metadirectory engine is older than
  It does not indicate anything about the version of t
Driver:  Driver.Driver Set.Novell
  Driver name:  Identity Manager Driver for Peoplesoft
  Driver module:  NPSShim.dll
  Driver Set running on Identity Vault:  IDMTEST.Novell
```

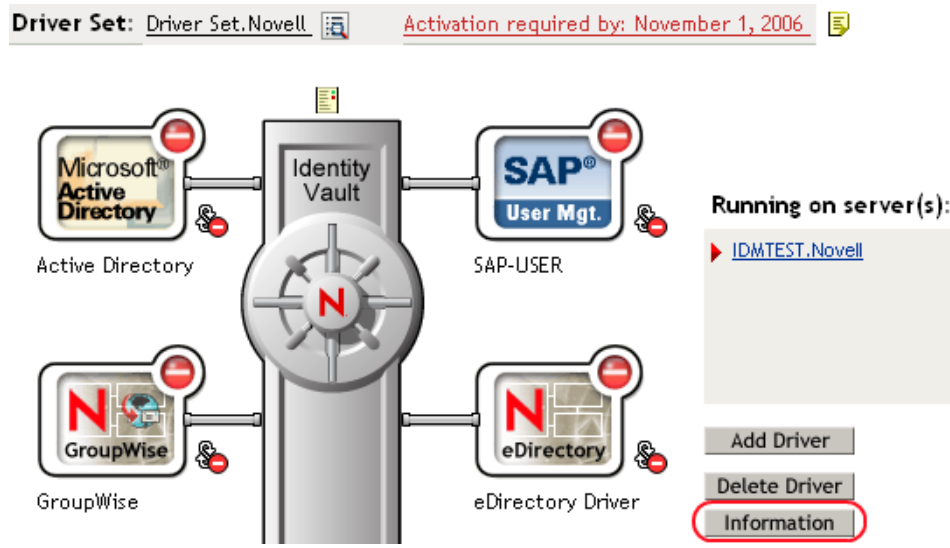
OK

7.6.3 Saving Version Information

You can save version information to a text file on your local or network drive.

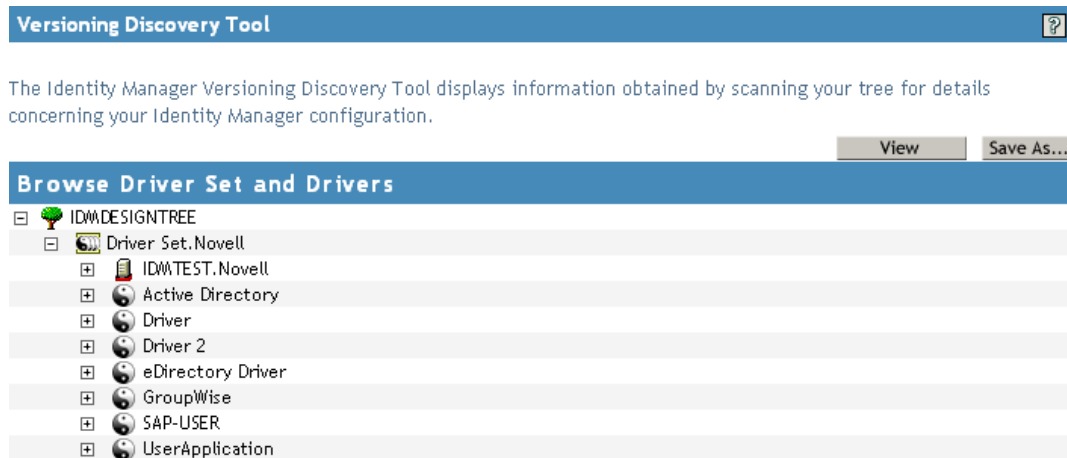
- 1 To find the Driver Set object in iManager, click *Identity Manager > Identity Manager Overview*, then click *Search*.

2 In the Identity Manager Overview, click *Information*.



You can also select *Identity Manager Utilities > Versions Discovery*, browse to and select the Driver Set object, then click *Information*.

3 In the Versioning Discovery Tool dialog box, click *Save As*.



4 In the File Download dialog box, click *Save*.

5 Navigate to the desired directory, type a filename, then click *Save*.

Identity Manager saves the data to a text file.

7.7 Reassociating a Driver Set Object with a Server Object

The driver set object should always be associated with a server object. If the driver set is not associated with a server object, none of the drivers in the driver set can start.

If the link between the driver set object and the server object becomes invalid, you see one of the following conditions:

- ◆ When upgrading eDirectory your Identity Manager server, you get the error UniqueSPIException error -783.
- ◆ No server is listed next to the driver set in the Identity Manager Overview window.
- ◆ A server is listed next to the driver set in the Identity Manager Overview window, but the name is garbled text.

To resolve this issue, disassociate the driver set object and the server object, then reassociate them.

- 1 In iManager click *Identity Manager > Identity Manager Overview*, then click *Search* to find the driver set object that the driver should be associated with.
- 2 Click the *Remove server* icon, then click *OK*.
- 3 Click the *Add server* icon, then browse to and select the server object.
- 4 Click *OK*.

7.8 Changing the Driver Configuration

If you need to change the driver configuration, Identity Manager allows you to make the change through Designer or iManager.

To change the driver configuration in Designer:

- 1 Open a project in the Modeler.
- 2 Right-click the driver line, then select *Properties*.

To change the driver configuration in iManager:

- 1 Click *Identity Manager > Identity Manager Overview*, then click *Search* to search for the driver set that is associated with the driver.
- 2 Browse to the driver, then click the upper right corner of the driver icon.
- 3 Click *Edit Properties*.

For a listing of all of the configuration fields, see [Appendix B, “Properties of the eDirectory Driver,” on page 89](#).

7.9 Storing Driver Passwords Securely with Named Passwords

Identity Manager allows you to store multiple passwords securely for a particular driver. This functionality is referred to as Named Passwords. Each different password is accessed by a key, or name.

You can also use the Named Passwords feature to store other pieces of information securely, such as a user name.

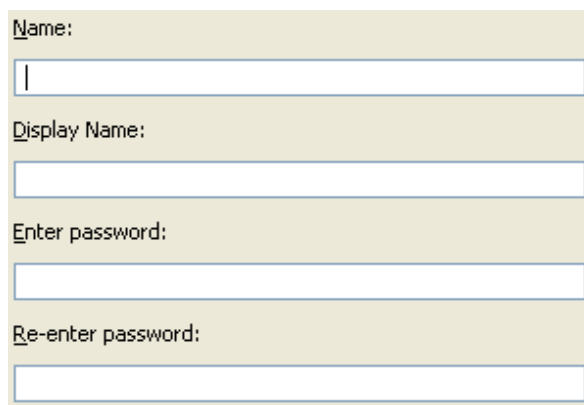
To use a named password in a driver policy, you refer to it by the name of the password, instead of using the actual password, and the Metadirectory engine sends the password to the driver. The

method described in this section for storing and retrieving named passwords can be used with any driver without making changes to the driver shim.

- ◆ [Section 7.9.1, “Using Designer to Configure Named Passwords,” on page 47](#)
- ◆ [Section 7.9.2, “Using iManager to Configure Named Passwords,” on page 47](#)
- ◆ [Section 7.9.3, “Using Named Passwords in Driver Policies,” on page 49](#)
- ◆ [Section 7.9.4, “Configuring Named Passwords by Using the DirXML Command Line Utility,” on page 50](#)

7.9.1 Using Designer to Configure Named Passwords

- 1 Right-click the Driver object, then select *Properties*.
- 2 Select *Named Password*, then click *New*.



Name:

Display Name:

Enter password:

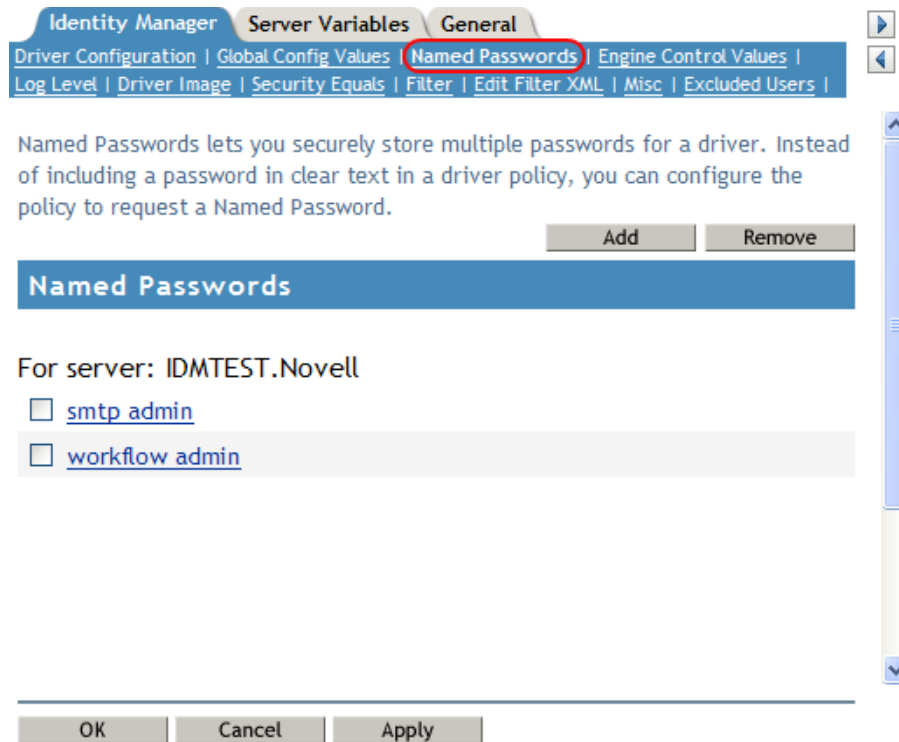
Re-enter password:

- 3 Specify the *Name* of the named password.
- 4 Specify the *Display name* of the named password.
- 5 Specify the named password, then re-enter the password.
- 6 Click *OK* twice.

7.9.2 Using iManager to Configure Named Passwords

- 1 Click *Identity Manager > Identity Manager Overview*.
- 2 Click *Search* to search for the driver set that is associated with the driver.
- 3 In the Identity Manager Overview, click the upper right corner of the driver icon, then click *Edit properties*.
- 4 On the Modify Object page on the Identity Manager tab, click *Named Passwords*.

The Named Passwords page appears, listing the current named passwords for this driver. If you have not set up any named passwords, the list is empty.



- 5 To add a named password, click *Add*, complete the fields, then click *OK*.

Named Password

Named Passwords lets you securely store multiple passwords for a driver. Instead of including a password in clear text in a driver policy, you can configure the policy to request a Named Password.

Name:

Display name:

Enter password:

Reenter password:

- 6 Specify a name, display name, and a password, then click *OK* twice.
You can use this feature to store other kinds of information securely, such as a username.

7 Click *OK* to restart the driver and have the changes take effect.

To remove a Named Password, select the password name, then click *Remove*. The password is removed without prompting you to confirm the action.

7.9.3 Using Named Passwords in Driver Policies

- ♦ “Making a Call to a Named Password” on page 49
- ♦ “Referencing a Named Password” on page 49

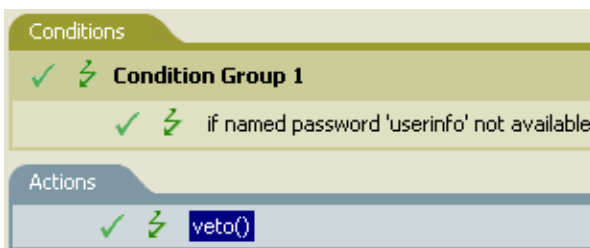
Making a Call to a Named Password

Policy Builder allows you to make a call to a named password. Create a new rule and select Named Password as the condition, then set an action, depending upon if the Named Password is available or not available.

- 1 In Designer, launch Policy Builder, right-click, then click *New > Rule*.
- 2 Specify the name of the rule, then click *Next*.
- 3 Select the condition structure, then click *Next*.
- 4 Select *named password* for the *Condition*.
- 5 Browse to and select the named password that is stored on the driver.
In this example, the named password is *userinfo*.
- 6 Select whether the Operator is available or not available.
- 7 Select an action for the *Do* field.
In this example, the action is *veto*.

The example indicates that if the *userinfo* named password is not available, then the event is vetoed.

Figure 7-1 A Policy Using Named Passwords



Referencing a Named Password

The following example shows how a named password can be referenced in a driver policy on the Subscriber channel in XSLT:

```
<xsl:value-of  
select="query:getNamedPassword($srcQueryProcessor, 'mynamedpassword')"  
xmlns:query="http://www.novell.com/java/  
com.novell.nds.dirxml.driver.XdsQueryProcessor/>
```

7.9.4 Configuring Named Passwords by Using the DirXML Command Line Utility

- ♦ “Creating a Named Password in the DirXML Command Line Utility” on page 50
- ♦ “Removing a Named Password by Using the DirXML Command Line Utility” on page 51

Creating a Named Password in the DirXML Command Line Utility

- 1 Run the DirXML Command Line utility.

For information, see [Appendix A, “The DirXML Command Line Utility,”](#) on page 75.

- 2 Enter your username and password.

The following list of options appears.

```
DirXML commands
1: Start driver
2: Stop driver
3: Driver operations...
4: Driver set operations...
5: Log events operations...
6: Get DirXML version
7: Job operations...
99: Quit
Enter choice:
```

- 3 Enter 3 for driver operations.

A numbered list of drivers appears.

- 4 Enter the number for the driver you want to add a named password to.

The following list of options appears.

```
Select a driver operation for:
driver_name
1: Start driver
2: Stop driver
3: Get driver state
4: Get driver start option
5: Set driver start option
6: Resync driver
7: Migrate from application into DirXML
8: Submit XDS command document to driver
9: Submit XDS event document to driver
10: Queue event for driver
11: Check object password
12: Initialize new driver object
13: Passwords operations
14: Cache operations
99: Exit
Enter choice:
```

- 5 Enter 13 for password operations.

The following list of options appears.

```
Select a password operation
```

```
1: Set shim password
2: Reset shim password
3: Set Remote Loader password
4: Clear Remote Loader password
5: Set named password
6: Clear named password(s)
7: List named passwords
8: Get passwords state
99: Exit
Enter choice:
```

6 Enter 5 to set a new named password.

The following prompt appears:

```
Enter password name:
```

7 Enter the name by which you want to refer to the named password.

8 Enter the actual password that you want to secure at the following prompt:

```
Enter password:
```

The characters you type for the password are not displayed.

9 Confirm the password by entering it again at the following prompt:

```
Confirm password:
```

10 After you enter and confirm the password, you are returned to the password operations menu.

11 After completing this procedure, use the 99 option twice to exit the menu and quit the DirXML Command Line Utility.

Removing a Named Password by Using the DirXML Command Line Utility

This option is useful if you no longer need named passwords that you previously created.

1 Run the DirXML Command Line utility.

For information, see [Appendix A, “The DirXML Command Line Utility,” on page 75](#).

2 Enter your username and password.

The following list of options appears.

```
DirXML commands
1: Start driver
2: Stop driver
3: Driver operations...
4: Driver set operations...
5: Log events operations...
6: Get DirXML version
7: Job operations
99: Quit
Enter choice:
```

3 Enter 3 for driver operations.

A numbered list of drivers appears.

4 Enter the number for the driver you want to remove named passwords from.

The following list of options appears.

Select a driver operation for:
driver_name
1: Start driver
2: Stop driver
3: Get driver state
4: Get driver start option
5: Set driver start option
6: Resync driver
7: Migrate from application into DirXML
8: Submit XDS command document to driver
9: Submit XDS event document to driver
10: Queue event for driver
11: Check object password
12: Initialize new driver object
13: Passwords operations
14: Cache operations
99: Exit
Enter choice:

5 Enter 13 for password operations.

The following list of options appears.

Select a password operation
1: Set shim password
2: Reset shim password
3: Set Remote Loader password
4: Clear Remote Loader password
5: Set named password
6: Clear named password(s)
7: List named passwords
8: Get passwords state
99: Exit
Enter choice:

6 (Optional) Enter 7 to see the list of existing named passwords.

The list of existing named passwords is displayed.

This step can help you make sure you are removing the correct password.

7 Enter 6 to remove one or more named passwords.

8 Enter No to remove a single named password at the following prompt:

Do you want to clear all named passwords? (yes/no):

9 Enter the name of the named password you want to remove at the following prompt:

Enter password name:

After you enter the name of the named password you want to remove, you are returned to the password operations menu:

Select a password operation
1: Set shim password
2: Reset shim password
3: Set Remote Loader password
4: Clear Remote Loader password
5: Set named password

```
6: Clear named password(s)
7: List named passwords
8: Get passwords state
99: Exit
Enter choice:
```

10 (Optional) Enter 7 to see the list of existing named passwords.

This step lets you verify that you have removed the correct password.

11 After completing this procedure, use the 99 option twice to exit the menu and quit the DirXML Command Line utility.

7.10 Adding a Driver Heartbeat

The driver heartbeat is a feature of the Identity Manager drivers that ship with Identity Manager 2 and later. Its use is optional. The driver heartbeat is configured by using a driver parameter with a time interval specified. If a heartbeat parameter exists and has an interval value other than 0, the driver sends a heartbeat document to the Metadirectory engine if no communication occurs on the Publisher channel for the specified interval of time.

The intent of the driver heartbeat is to give you a trigger to allow you to initiate an action at regular intervals, if the driver does not communicate on the Publisher channel as often as you want the action to occur. To take advantage of the heartbeat, you must customize your driver configuration or other tools. The Metadirectory engine accepts the heartbeat document but does not take any action because of it.

For most drivers, a driver parameter for heartbeat is not used in the sample configurations, but you can add it.

A custom driver that is not provided with Identity Manager can also provide a heartbeat document, if the driver developer has written the driver to support it.

To configure the heartbeat:

- 1** In iManager, click *Identity Manager > Identity Manager Overview*.
- 2** Browse to and select your driver set object, then click *Search*.
- 3** In the Identity Manager Overview, click the upper right corner of the driver icon, then click *Edit properties*.
- 4** On the Identity Manager tab, click *Driver Configuration*, scroll to *Driver Parameters*, then look for Heart Beat or a similar display name.

If a driver parameter already exists for heartbeat, you can change the interval and save the changes. Configuration is then complete.

The value of the interval cannot be less than 1. A value of 0 means that the feature is turned off.

The unit of time is usually minutes; however, some drivers might choose to implement it differently, such as using seconds.

- 5** If a driver parameter does not exist for heartbeat, click *Edit XML*.
- 6** Add a driver parameter entry similar to the following example, as a child of `<publisher-options>`.

```
<pub-heartbeat-interval display-name="Heart Beat">10</pub-
heartbeat-interval>
```

TIP: If the driver does not produce a heartbeat document after being restarted, check the placement of the driver parameter in the XML.

- 7 Save the changes, then make sure the driver is stopped and restarted.

After you have added the driver parameter, you can edit the time interval by using the graphical view. Another option is to create a reference to a global configuration value (GCV) for the time interval. Like other global configuration values, the driver heartbeat can be set at the driver set level instead of on each individual Driver object. If a driver does not have a particular global configuration value, and the Driver Set object does have it, the driver inherits the value from the Driver Set object.

Synchronizing Objects

8

This section explains driver and object synchronization in DirXML[®] 1.1a, Identity Manager 2.0, and Identity Manager 3.x. Driver synchronization was not available for DirXML 1.0 and DirXML 1.1.

After the driver is created, instead of waiting for objects to be modified or created, the data between the two connected systems can be sent through the synchronization process.

- ♦ [Section 8.1, “What Is Synchronization?” on page 55](#)
- ♦ [Section 8.2, “When Does Synchronization Occur?” on page 55](#)
- ♦ [Section 8.3, “How Does the Metadirectory Engine Decide Which Object to Synchronize?” on page 56](#)
- ♦ [Section 8.4, “How Synchronization Works,” on page 57](#)

8.1 What Is Synchronization?

The actions commonly referred to as “synchronization” in Identity Manager refer to several different but related actions:

- ♦ Synchronization (or merging) of attribute values of an object in the Identity Vault with the corresponding attribute values of an associated object in a connected system.
- ♦ Migration of all Identity Vault objects and classes that are included in the filter on the Subscriber channel.
- ♦ Generation of the list of objects to submit to the driver’s Subscriber channel for synchronization or migration in response to a user request (a manual synchronization).
- ♦ Generation of the list of objects to submit to the driver’s Subscriber channel for synchronization or migration in response to enabling a formerly disabled driver, or in response to a cache error.

8.2 When Does Synchronization Occur?

The Metadirectory engine synchronizes objects or merges them in the following circumstances:

- ♦ When a `<sync>` event element is submitted on the Subscriber or Publisher channel.
- ♦ When a `<sync>` event element is submitted on the Subscriber channel in the following circumstances:
 - ♦ The state of the object’s association value is set to “manual” or “migrate.” (This causes an eDirectory™ event, which in turn causes the Identity Manager caching system to queue an object synchronization command in the affected driver’s cache.)
 - ♦ An object synchronization command is read from the driver’s cache.
- ♦ When a `<sync>` event element is submitted on the Publisher channel in the following circumstances:
 - ♦ A driver submits a `<sync>` event element. No known driver currently does this.

- ◆ The Metadirectory engine submits a <sync> event element for each object found as the result of a migrate-into-NDS query. The engine submits these <sync> events by using the Subscriber thread, but processes them by using the Publisher channel filter and policies.
- ◆ When an <add> event (real or synthetic) is submitted on a channel, and the channel Matching policy finds a matching object in the target system.
- ◆ When an <add> event with an association is submitted on the Subscriber channel. This normally occurs only in exceptional cases, such as the bulk load of objects into eDirectory with DirXML-Associations attribute values.
- ◆ When an <add> event is submitted on the Publisher channel, and an object is found in eDirectory that already has the association value reported with the <add> event.

The Metadirectory engine generates synchronization requests for zero or more objects in the following cases:

- ◆ The user issues a manual driver synchronization request. This corresponds to the *Resync* button in the Driver Set property page in ConsoleOne[®], or to the *Synchronize* button on the iManager Identity Manager Driver Overview page.
- ◆ The Metadirectory engine encounters an error with the driver's cache and cannot recover from the cache error. The driver's cache is deleted, and the engine generates object synchronization commands as detailed in [Section 8.3, "How Does the Metadirectory Engine Decide Which Object to Synchronize?," on page 56](#).

8.3 How Does the Metadirectory Engine Decide Which Object to Synchronize?

The Metadirectory engine processes both manually initiated and automatically initiated synchronization requests in the same manner. The only difference in the processing of manually initiated versus automatically initiated driver synchronization requests is the starting filter time used to filter objects being considered for synchronization.

The starting filter time is used to filter objects that have modification or creation times that are older than the starting time specified in the synchronization request.

For automatically initiated driver synchronization, the starting filter time is obtained from the time stamps of cached eDirectory events. In particular, the starting filter time is the earliest time for the cached events that haven't yet been successfully processed by the driver's Subscriber channel.

For manually initiated driver synchronization, the default starting filter time is the earliest time in the eDirectory database. In Identity Manager 2 and Identity Manager 3, an explicit starting filter time can also be set. DirXML[®] 1.1a has no facility to set the starting filter time value for synchronization when manually initiating driver synchronization.

The Metadirectory engine creates a list of objects to be synchronized on the Subscriber channel in the following manner:

1. It finds all objects that:
 - ◆ Have an entry modification time stamp greater than or equal to the starting filter time and
 - ◆ Exist in the filter on the Subscriber channel.

2. It finds all objects that have an entry creation time stamp greater than or equal to the starting filter time.
3. It adds a `synchronize object` command to the following:
 - ♦ The driver cache for each unique object found that has an entry modification time stamp greater than or equal to the starting filter time
 - ♦ All objects and classes that are in the Subscriber filter channel in the driver being synchronized

8.4 How Synchronization Works

After the Metadirectory engine determines that an object is to be synchronized, the following processes occur:

1. Each system (the Identity Vault and the connected system) is queried for all attribute values in the appropriate filters.
 - ♦ eDirectory is queried for all values in the Subscriber filter, and for values that are marked for synchronization in Identity Manager 2.x and Identity Manager 3.x.
 - ♦ The connected system is queried for all values in the Publisher filter, and for values that are marked for synchronization in Identity Manager 2.x and Identity Manager 3.x.
2. The returned attribute values are compared, and modification lists are prepared for the Identity Vault and the connected system according to [Table 8-1 on page 58](#), [Table 8-2 on page 60](#), and [Table 8-3 on page 61](#).

In the tables the following pseudo-equations are used:

- ♦ “Left = Right” indicates that the left side receives all values from the right side.
- ♦ “Left = Right[1]” indicates that the left side receives one value from the right side. If there is more than one value, it is indeterminate.
- ♦ “Left += Right” indicates that the left side adds the right side values to the left side’s existing values.
- ♦ “Left = Left + Right” indicates that the left side receives the union of the values of the left and right sides.

Identity Manager has three different combinations of selected items in the filter, and each one creates a different output.

- ♦ [Section 8.4.1, “Scenario One,” on page 57](#)
- ♦ [Section 8.4.2, “Scenario Two,” on page 59](#)
- ♦ [Section 8.4.3, “Scenario Three,” on page 60](#)

8.4.1 Scenario One

The attribute is set to *Synchronize* on the Publisher and Subscriber channels, and the merge authority is set to *Default*.

Figure 8-1 Scenario One

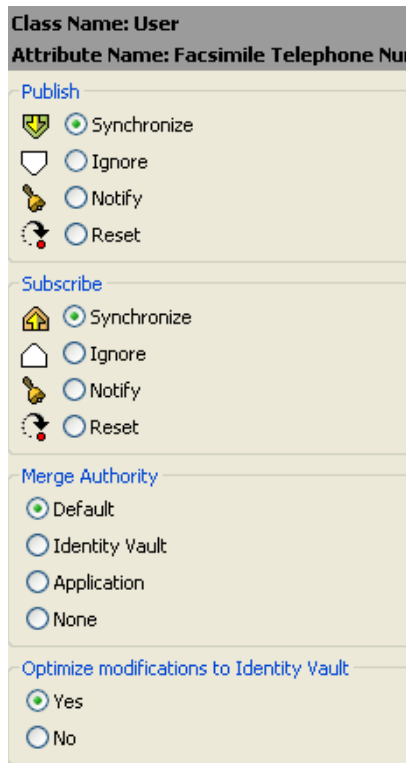


Table 8-1 on page 58 contains the values that the Metadirectory engine synchronizes when the attribute is sent through a filter that is set to the configuration for Scenario One. The table shows different outputs, depending upon the following:

- ◆ Whether the attribute comes from the Identity Vault or the Application
- ◆ If the attribute is single-valued or multi-valued, and if the attribute is empty or non-empty.
- ◆ If the attribute is empty or non-empty

Table 8-1 Output of Scenario One

	Identity Vault single-valued empty	Identity Vault single-valued non-empty	Identity Vault multi-valued empty	Identity Vault multi-valued non-empty
Application single-valued empty	No change	App = Identity Vault	No change	App = Identity Vault[1]
Application single-valued non-empty	Identity Vault = App	App = Identity Vault	Identity Vault = App	Identity Vault + = App
Application multi-valued empty	No change	App = Identity Vault	No change	App = Identity Vault

	Identity Vault single-valued empty	Identity Vault single-valued non-empty	Identity Vault multi-valued empty	Identity Vault multi-valued non-empty
Application multi-valued non-empty	Identity Vault = App[1]	App + = Identity Vault	Identity Vault = App	App = App + Identity Vault Identity Vault = App + Identity Vault

8.4.2 Scenario Two

The attribute is set to *Synchronize* only on the Subscriber channel, or it is set to *Synchronize* on both the Subscriber and Publisher channels. The merge authority is set to *Identity Vault*.

Figure 8-2 Scenario Two

Class Name: User

Attribute Name: Description

Publish

Synchronize

Ignore

Notify

Reset

Subscribe

Synchronize

Ignore

Notify

Reset

Merge Authority

Default

Identity Vault

Application

None

Optimize modifications to Identity Vault

Yes

No

Table 8-2 on page 60 contains the values that the Metadirectory engine synchronizes when the attribute is sent through a filter that is set to the configuration for Scenario Two. The table shows different outputs depending upon the following:

- ◆ Whether the attribute comes from the Identity Vault or the Application
- ◆ If the attribute is single-valued or multi-valued
- ◆ If the attribute is empty or non-empty

Table 8-2 Output of Scenario Two

	Identity Vault single-valued empty	Identity Vault single-valued non-empty	Identity Vault multi-valued empty	Identity Vault multi-valued non-empty
Application single-valued empty	No change	App = Identity Vault	No change	App = Identity Vault[1]
Application single-valued empty	App = empty	App = Identity Vault	Identity Vault = App	App = Identity Vault[1]
Application multi-valued empty	No change	App = Identity Vault	No change	App = Identity Vault
Application multi-valued non-empty	App = empty	App = Identity Vault	App = empty	App = Identity Vault

8.4.3 Scenario Three

The attribute is set to *Synchronize* on the Publisher channel, or the merge authority is set to *Application*.

Figure 8-3 Scenario Three

Class Name: User

Attribute Name: DirXML-ADAliasName

Publish

Synchronize

Ignore

Notify

Reset

Subscribe

Synchronize

Ignore

Notify

Reset

Merge Authority

Default

Identity Vault

Application

None

Optimize modifications to Identity Vault

Yes

No

Table 8-3 on page 61 contains the values that the Metadirectory engine synchronizes when the attribute is sent through a filter that is set to the configuration for Scenario Three. The table shows different outputs depending upon the following:

- ◆ Whether the attribute comes from the Identity Vault or the Application
- ◆ If the attribute is single-valued or multi-valued
- ◆ If the attribute is empty or non-empty

Table 8-3 *Output of Scenario Three*

	Identity Vault single-valued empty	Identity Vault single-valued non-empty	Identity Vault multi-valued empty	Identity Vault multi-valued non-empty
Application single-valued empty	No change	Identity Vault = empty	No change	Identity Vault = empty
Application single-valued non-empty	Identity Vault = App	Identity Vault = App	Identity Vault = App	Identity Vault = App
Application multi-valued empty	No change	Identity Vault = empty	No change	Identity Vault = empty
Application multi-valued non- empty	Identity Vault = App[1]	Identity Vault = App[1]	Identity Vault = App	Identity Vault = App

Troubleshooting the eDirectory Driver

9

- ♦ [Section 9.1, “Troubleshooting Driver Processes,” on page 63](#)
- ♦ [Section 9.2, “Creating Functionality on the Driver Parameters Page,” on page 69](#)

9.1 Troubleshooting Driver Processes

Viewing driver processes is necessary to analyze unexpected behavior. To view the driver processing events, use DSTrace. You should only use it during testing and troubleshooting the driver. Running DSTrace while the drivers are in production increases the utilization on the Identity Manager server and can cause events to process very slowly.

9.1.1 Viewing Driver Processes

To see the driver processes in DSTrace, values are added to the Driver Set object and the Driver object. You can do this in Designer or iManager.

- ♦ [“Adding Trace Levels in Designer” on page 63](#)
- ♦ [“Adding Trace Levels in iManager” on page 65](#)
- ♦ [“Capturing Driver Processes to a File” on page 66](#)

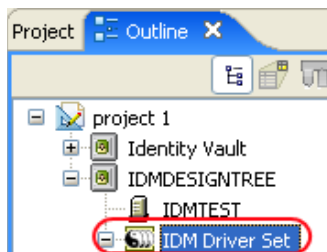
Adding Trace Levels in Designer

You can add trace levels to the Driver Set object or to each Driver object.

- ♦ [“Driver Set” on page 63](#)
- ♦ [“Driver” on page 64](#)

Driver Set

- 1 In an open project in Designer, select the Driver Set object in the *Outline* view.



- 2 Right-click, select *Properties*, then click *5. Trace*.
- 3 Set the parameters for tracing, then click *OK*.

Parameter	Description
Driver trace level	As the Driver object trace level increases, the amount of information displayed in DSTrace increases. Trace level 1 shows errors, but not the cause of the errors. To see password synchronization information, set the trace level to 5.
XSL trace level	DSTrace displays XSL events. Set this trace level only when troubleshooting XSL style sheets. If you do not want to see XSL information, set the level to zero.
Java debug port	Allows developers to attach a Java debugger.
Java trace file	When a value is set in this field, all Java information for the Driver Set object is written to a file. The value for this field is the path for that file. As long as the file is specified, Java information is written to this file. If you do not need to debug Java, leave this field blank.
Trace file size limit	Allows you to set a limit for the Java trace file. If you set the file size to <i>Unlimited</i> , the file grows in size until no disk space remains.

If you set the trace level on the Driver Set object, all drivers appear in the DSTrace logs.

Driver

- 1 In an open project in Designer, select the Driver object in the *Outline* view.
- 2 Right-click, select *Properties*, then click *8. Trace*.
- 3 Set the parameters for tracing, then click *OK*.

Parameter	Description
Trace level	As the Driver object trace level increases, the amount of information displayed in DSTrace increases. Trace level 1 shows errors, but not the cause of the errors. To see password synchronization information, set the trace level to 5. if you select <i>Use setting from Driver Set</i> , the value is taken from the Driver Set object.
Trace file	Specify a filename and location for where the Identity Manager information is written for the selected driver. if you select <i>Use setting from Driver Set</i> , the value is taken from the Driver Set object.
Trace file size limit	Allows you to set a limit for the Java trace file. If you set the file size to <i>Unlimited</i> , the file grows in size until no disk space remains. If you select <i>Use setting from Driver Set</i> , the value is taken from the Driver Set object.
Trace name	The driver trace messages are prepended with the value entered instead of the driver name. Use this option if the driver name is very long.

If you set the parameters only on the Driver object, only information for that driver appears in the DSTrace log.

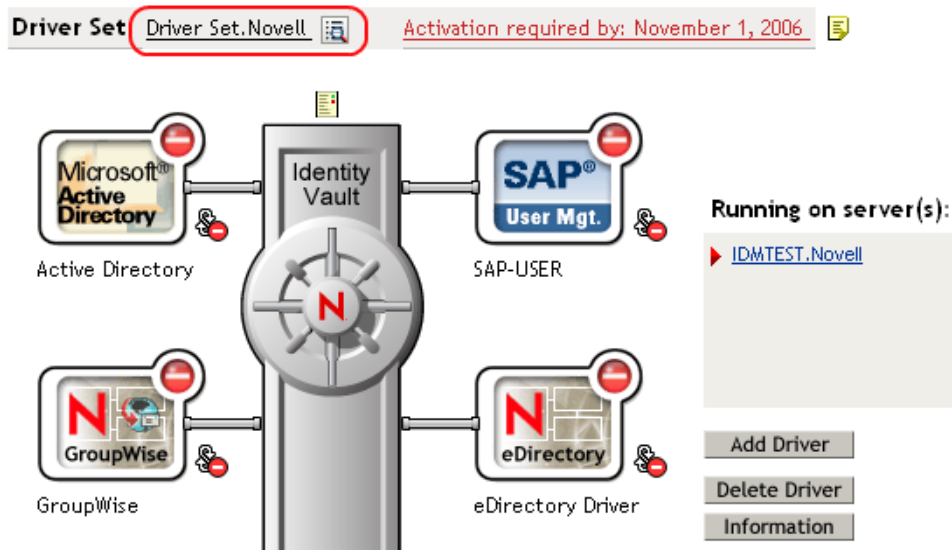
Adding Trace Levels in iManager

You can add trace levels to the Driver Set object or to each Driver object.

- ♦ “Driver Set” on page 65
- ♦ “Driver” on page 65

Driver Set

- 1 In iManager, select *Identity Manager > Identity Manager Overview*.
- 2 Browse to the Driver Set object, then click *Search*.
- 3 Click the driver set name.



- 4 Select the *Misc* tab for the Driver Set object.
- 5 Set the parameters for tracing, then click *OK*.

Driver

- 1 In iManager, select *Identity Manager > Identity Manager Overview*.
- 2 Browse to the Driver Set object where the Driver object resides, then click *Search*.
- 3 Click the upper right corner of the Driver object, then click *Edit properties*.
- 4 Select the *Misc* tab for the Driver object.
- 5 Set the parameters for tracing, then click *OK*.

NOTE: The option *Use setting from Driver Set* does not exist in iManager.

Capturing Driver Processes to a File

You can save driver processes to a file by using the parameter on the Driver object or by using DSTrace. The parameter on the Driver object is the *Trace file* parameter, under the *MISC* tab.

The driver processes that are captured through DSTrace are the processes that occur on the Identity Manager engine. If you use the Remote Loader, you need to capture a trace on the Remote Loader at the same time as you are capturing the trace on the Identity Manager engine.

The following methods help you capture and save Identity Manager processes through DSTrace on different platforms.

- ◆ “NetWare” on page 66
- ◆ “Windows” on page 66
- ◆ “UNIX” on page 67
- ◆ “iMonitor” on page 67
- ◆ “Remote Loader” on page 68

NetWare

Use `dstrace.nlm` to display trace messages on the system console or trace messages to a file (`sys:\system\dstrace.log`). Use `dstrace.nlm` to display the trace messages to a screen labeled DSTrace Console.

- 1 Enter `dstrace.nlm` at the server console to load `dstrace.nlm` into memory.
- 2 Enter `dstrace screen on` at the server console to allow trace messages to appear on the DSTrace Console screen.
- 3 Enter `dstrace file on` at the server console to capture trace messages sent to the DSTrace Console to the `dstrace.log` file.
- 4 (Optional) Enter `dstrace -all` at the server console to make it easier to read the trace log.
- 5 Enter `dstrace +dxml dstrace +dvr` at the server console to display Identity Manager events.
- 6 Enter `dstrace +tags dstrace +time` at the server console to display message tags and time stamps.
- 7 Toggle to the DSTrace Console screen and watch for the event to pass.
- 8 Toggle back to the server console.
- 9 Enter `dstrace file off` at the server console.
This stops capturing trace messages to the log file. It also stops logging information into the file.
- 10 Open the `dstrace.log` in a text editor and search for the event or the object you modified.

Windows

- 1 Open the Control Panel, select *NDS Services* > `dstrace.dlm`, then click *Start* to display the NDS Server Trace utility window.
- 2 Click *Edit* > *Options*, then click *Clear All* to clear all of the default flags.
- 3 Select *DirXML* and *DirXML Drivers*.
- 4 Click OK.

- 5 Click *File > New*.
- 6 Specify the filename and location where you want the DSTrace information saved, then click *Open*.
- 7 Wait for the event to occur.
- 8 Click *File > Close*.
This stops the information from being written to the log file.
- 9 Open the file in a text editor and search for the event or the object you modified.

UNIX

- 1 Enter `ndstrace` to start the ndstrace utility.
- 2 Enter `set ndstrace=nodebug` to turn off all trace flags currently set.
- 3 Enter `set ndstrace on` to display trace messages to the console.
- 4 Enter `set ndstrace file on` to capture trace messages to the `ndstrace.log` file in the directory where eDirectory is installed. By default it is `/var/nds`.
- 5 Enter `set ndstrace=+dxml` to display the Identity Manager events.
- 6 Enter `set ndstrace=+dvrs` to display the Identity Manager driver events.
- 7 Wait for the event to occur.
- 8 Enter `set ndstrace file off` to stop logging information to the file.
- 9 Enter `exit` to quite the ndstrace utility.
- 10 Open the file in a text editor. Search for the event or the object that was modified.

iMonitor

iMonitor allows you to get DSTrace information from a Web browser. It does not matter where Identity Manager is running. The following files run iMonitor:

- ◆ `ndsimon.nlm` runs on NetWare®.
- ◆ `ndsimon.dlm` runs on Windows.
- ◆ `ndsmonitor` runs on UNIX*.

- 1 Access iMonitor from `http://server_ip:8008/nds`.
Port 8008 is the default.
- 2 Specify a username and password with administrative rights, then click *Login*.
- 3 Select *Trace Configuration* on the left side.
- 4 Click *Clear All*.
- 5 Select *DirXML* and *DirXML Drivers*.
- 6 Click *Trace On*.
- 7 Select *Trace History* on the left side.
- 8 Click the document with the *Modification Time of Current* to see a live trace.
- 9 Change the *Refresh Interval* if you want to see information more often.
- 10 Select *Trace Configuration* on the left side, then click *Trace Off* to turn the tracing off.
- 11 Select *Trace History* to view the trace history.

The files are distinguished by the time stamp.

If you need a copy of the HTML file, the default location is:

- ◆ NetWare: `sys:\system\ndsimon\dstrace*.htm`
- ◆ Windows: `Drive_letter:\novell\nds\ndsimon\dstrace*.htm`
- ◆ UNIX: `/var/nds/dstrace/*.htm`

Remote Loader

You can capture the events that occur on the machine running the Remote Loader service.

- 1 Launch the Remote Loader Console by clicking the icon.
- 2 Select the driver instance, then click *Edit*.
- 3 Set the *Trace Level* to 3 or above.
- 4 Specify a location and file for the trace file.
- 5 Specify the amount of disk space that the file is allowed.
- 6 Click *OK* twice to save the changes.

You can also enable tracing from the command line by using the following switches. For more information, see “[Configuring the Remote Loader](#)” in the *Novell Identity Manager 3.5.1 Administration Guide*.

Table 9-1 Command Line Tracing Switches

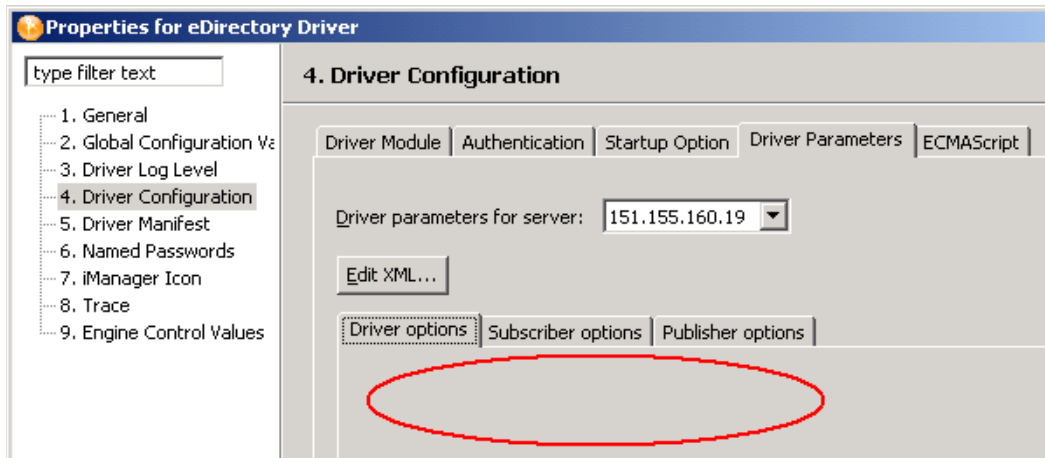
Option	Short Name	Parameter	Description
-trace	-t	integer	Specifies the trace level. This is used only when hosting an application shim. Trace levels correspond to those used on the Identity Manager server. Example: <code>-trace 3</code> or <code>-t3</code>
-tracefile	-tf	filename	Specify a file to write trace messages to. Trace messages are written to the file if the trace level is greater than zero. Trace messages are written to the file even if the trace window is not open. Example: <code>-tracefile c:\temp\trace.txt</code> or <code>-tf c:\temp\trace.txt</code>

Option	Short Name	Parameter	Description
-tracefilemax	-tfm	size	<p>Specifies the approximate maximum size that trace file data can occupy on disk. If you specify this option, Identity Manager creates a trace file with the name specified by using the tracefile option and up to 9 additional “roll-over” files. The roll-over files are named by using the base of the main trace filename plus “_n”, where n is 1 through 9.</p> <p>The size parameter is the number of bytes. Specify the size by using the suffixes K, M, or G for kilobytes, megabytes, or gigabytes.</p> <p>If the trace file data is larger than the specified maximum when the Remote Loader is started, the trace file data remains larger than the specified maximum until roll-over is completed through all 10 files.</p> <p>Example: <code>-tracefilemax 1000M</code> or <code>-tfm 1000M</code></p>

9.2 Creating Functionality on the Driver Parameters Page

If no parameters appear on the eDirectory driver’s *Driver Parameters* tab:

Figure 9-1 Empty Tabs on the Driver Parameters Page

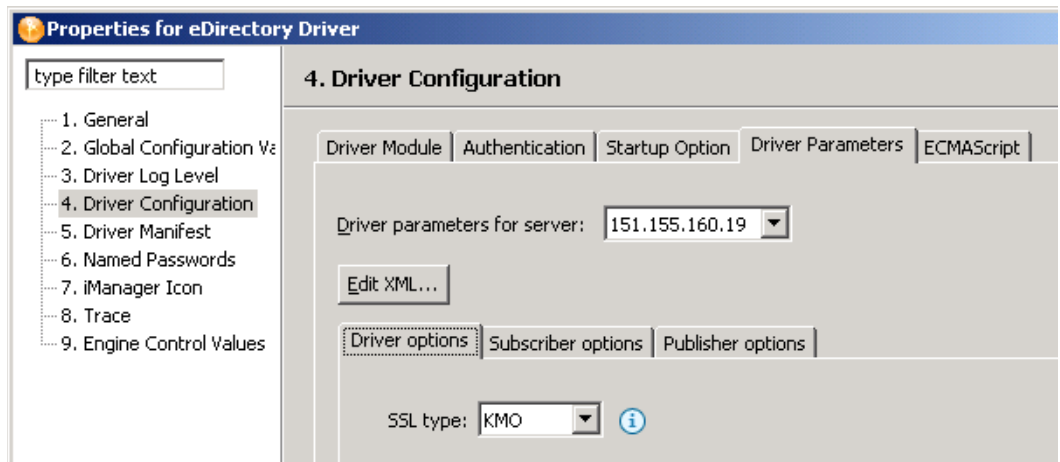


- 1 On the Novell Support Web site, open [Novell Support TID 2970417](http://www.novell.com/support/search.do?cmd=displayKC&docType=kc&externalId=InfoDocument-2970417&sliceId=&dialogID=29785513&stateId=0%20%2029791519) (<http://www.novell.com/support/search.do?cmd=displayKC&docType=kc&externalId=InfoDocument-2970417&sliceId=&dialogID=29785513&stateId=0%20%2029791519>).

Although this TID was written for Identity Manager 2.0.1, the information applies to Identity Manager 3.5.1.

- 2 Download `idm201edirir1.tgz`.
- 3 Unzip the file.
- 4 Open `options.xml` with a text editor, then copy the entire text.
- 5 In Designer or iManager, click *Edit XML* on the Driver Parameters page.

- 6 Overwrite the text in the Driver Parameters XML window by pasting the contents of `options.xml` into the window.
- 7 Click *OK*.



Settings and values appear on the *Driver options*, *Subscriber options*, and *Publisher options* tabs.

Backing Up the eDirectory Driver

10

You can use Designer for Identity Manager or iManager to create an XML file of the driver. The file contains all of the information that you entered into the driver during configuration. If the driver becomes corrupted, you can restore the configuration information by importing the exported file.

IMPORTANT: If the driver has been deleted, all of the associations on the objects are purged. When you import the XML file, the migration process creates new associations.

Not all server-specific information stored on the driver is contained in the XML file. Make sure that this information is documented through the Document Generation process in Designer. See “[Documenting Projects](#)” in the *Designer 2.1 for Identity Manager 3.5.1* guide.

- ♦ [Section 10.1, “Exporting the Driver in Designer,” on page 71](#)
- ♦ [Section 10.2, “Exporting the Driver in iManager,” on page 71](#)

10.1 Exporting the Driver in Designer

- 1 Open a project in Designer, then right-click the Driver object.
- 2 Select *Export to Configuration File*.
- 3 Specify a unique name for the configuration file, browse to location where it should be saved, then click *Save*.
- 4 Click *OK* in the Export Configuration Results window.

10.2 Exporting the Driver in iManager

- 1 In iManager, select *Identity Manager > Identity Manager Overview*.
- 2 Browse to and select the Driver Set object, then click *Search*.
- 3 Click the driver icon.
- 4 Select *Export* in the Identity Manager Driver Overview window.
- 5 Browse to and select the Driver object that you want to export, then click *Next*.
- 6 Select *Export all policies, linked to the configuration or not* or select *Only export policies that are linked to the configuration*, depending upon the information you want to have stored in the XML file.
- 7 Click *Next*.
- 8 Click *Save As*, then click *Save*.
- 9 Browse to and select a location to save the XML file, then click *Save*.
- 10 Click *Finish*.

Security: Best Practices

11

To secure the driver and the information it is synchronizing, see “[Security: Best Practices](#)” in the *Novell Identity Manager 3.5.1 Administration Guide*.

The DirXML Command Line Utility

A

The DirXML[®] Command Line utility allows you to use a command line interface to manage the driver. You can create scripts that have the commands to manage the driver.

The utility and scripts are installed on all platforms during the Identity Manager installation. The utility is installed to the following locations:

- ♦ Windows: \Novell\Nds\dxcmd.bat
- ♦ NetWare[®]: sys:\system\dxcmd.ncf
- ♦ UNIX: /usr/bin/dxcmd

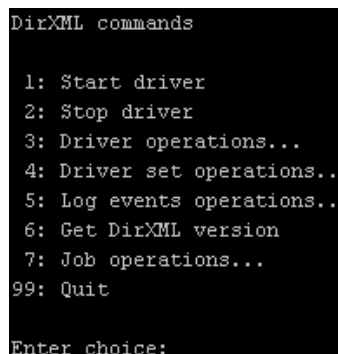
Either of the following methods enable you to use the DirXML Command Line utility:

- ♦ [Section A.1, “Interactive Mode,” on page 75](#)
- ♦ [Section A.2, “Command Line Mode,” on page 84](#)

A.1 Interactive Mode

The interactive mode provides a text interface to control and use the DirXML Command Line utility.

- 1 At the console, enter dxcmd.
- 2 Enter the name of a user with sufficient rights to the Identity Manager objects, such as admin.novell.
- 3 Enter the user’s password.



```
DirXML commands
1: Start driver
2: Stop driver
3: Driver operations...
4: Driver set operations...
5: Log events operations...
6: Get DirXML version
7: Job operations...
99: Quit
Enter choice:
```

- 4 Enter the number of the command that you want to perform.
[Table A-1 on page 76](#) contains the list of options and what functionality is available.
- 5 To quit the utility, enter 99.

NOTE: If you are running eDirectory[™] 8.8 on UNIX or Linux, you must specify the -host and -port parameters. For example, dxcmd -host 10.0.0.1 -port 524. If the parameters are not specified, a jclient error occurs.

```
novell.jclient.JCException: connect (to address) 111 UNKNOWN ERROR
```

By default, eDirectory 8.8 is not listening to localhost. The DirXML Command Line utility needs to resolve the server IP address or hostname and the port to be able to authenticate.

Table A-1 *Interactive Mode Options*

Option	Description
1: <i>Start Driver</i>	Starts the driver. If more than one driver exists, each driver is listed with a number. Enter the number of the driver to start the driver.
2: <i>Stop Driver</i>	Stops the driver. If more than one driver exists, each driver is listed with a number. Enter the number of the driver to stop the driver.
3: <i>Driver operations</i>	Lists the operations available for the driver. If more than one driver exists, each driver is listed with a number. Enter the number of the driver to see the operations available. See Table A-2 on page 77 for a list of operations.
4: <i>Driver set operations</i>	Lists the operations available for the driver set. <ul style="list-style-type: none">◆ 1: Associate driver set with server◆ 2: Disassociate driver set from server◆ 99: Exit
5: <i>Log events operations</i>	Lists the operations available for logging events through Novell® Audit. See Table A-5 on page 81 for a description of these options.
6: <i>Get DirXML version</i>	Lists the installed version of Identity Manager.
7: <i>Job operations</i>	Manages jobs created for Identity Manager.
99: <i>Quit</i>	Exits the DirXML Command Line utility

Figure A-1 *Driver Options*

```
Select a driver operation for:
Active Directory.Driver Set.Novell.IDMDESIGNTREE.

1: Start driver
2: Stop driver
3: Get driver state
4: Get driver start option
5: Set driver start option
6: Resync driver
7: Migrate from application into DirXML
8: Submit XDS command document to driver
9: Submit XDS event document to driver
10: Queue event for driver
11: Check object password
12: Initialize new driver object
13: Passwords operations
14: Cache operations
99: Exit

Enter choice:
```

Table A-2 *Driver Options*

Options	Description
1: <i>Start driver</i>	Starts the driver.
2: <i>Stop driver</i>	Stops the driver.
3: <i>Get driver state</i>	Lists the state of the driver. <ul style="list-style-type: none">◆ 0 - Driver is stopped◆ 1 - Driver is starting◆ 2 - Driver is running◆ 3 - Driver is stopping
4: <i>Get driver start option</i>	Lists the current driver start option. <ul style="list-style-type: none">◆ 1 - Disabled◆ 2 - Manual◆ 3 - Auto
5: <i>Set driver start option</i>	Changes the start option of the driver. <ul style="list-style-type: none">◆ 1 - Disabled◆ 2 - Manual◆ 3 - Auto◆ 99 - Exit
6: <i>Resync driver</i>	<p>Forces a resynchronization of the driver. It prompts for a time delay: <i>Do you want to specify a minimum time for resync? (yes/no)</i>.</p> <p>If you enter <i>Yes</i>, specify the date and time you want the resynchronization to occur: <i>Enter a date/time (format 9/27/05 3:27 PM)</i>.</p> <p>If you enter <i>No</i>, the resynchronization occurs immediately.</p>
7: <i>Migrate from application into DirXML</i>	<p>Processes an XML document that contains a query command: <i>Enter filename of XDS query document:</i></p> <p>Create the XML document that contains a query command by using the Novell <code>nds.dtd</code> (http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/ref/ndsstdtd/query.html).</p> <p>Examples:</p> <p>NetWare: <code>sys:\files\query.xml</code></p> <p>Windows: <code>c:\files\query.xml</code></p> <p>Linux: <code>/files/query.xml</code></p>

Options	Description
8: <i>Submit XDS command document to driver</i>	<p>Processes an XDS command document:</p> <p><i>Enter filename of XDS command document:</i></p> <p>Examples:</p> <p>NetWare: <code>sys:\files\user.xml</code></p> <p>Windows: <code>c:\files\user.xml</code></p> <p>Linux: <code>/files/user.xml</code></p> <p><i>Enter name of file for response:</i></p> <p>Examples:</p> <p>NetWare: <code>sys:\files\user.log</code></p> <p>Windows: <code>c:\files\user.log</code></p> <p>Linux: <code>/files/user.log</code></p>
9: <i>Submit XDS event document to driver</i>	<p>Processes an XDS event document:</p> <p><i>Enter filename of XDS event document:</i></p> <p>Examples:</p> <p>NetWare: <code>sys:\files\add.xml</code></p> <p>Windows: <code>c:\files\add.xml</code></p> <p>Linux: <code>/files/add.xml</code></p>
10: <i>Queue event for driver</i>	<p>Adds an event to the driver queue</p> <p><i>Enter filename of XDS event document:</i></p> <p>Examples:</p> <p>NetWare: <code>sys:\files\add.xml</code></p> <p>Windows: <code>c:\files\add.xml</code></p> <p>Linux: <code>/files/add.xml</code></p>
11: <i>Check object password</i>	<p>Validates that an object's password in the connected system is associated with a driver. It matches the object's eDirectory password (Distribution Password, used with Universal Password).</p> <p><i>Enter user name:</i></p>
12: <i>Initialize new driver object</i>	<p>Performs an internal initialization of data on a new Driver object. This is only for testing purposes.</p>
13: <i>Password operations</i>	<p>Nine Password options are available. See Table A-3 on page 79 for a description of these options.</p>
14: <i>Cache operations</i>	<p>Five Cache operations are available. See Table A-4 on page 80 for a descriptions of these options.</p>

Options	Description
99: <i>Exit</i>	Exits the driver options.

Figure A-2 Password Operations

```
Select a password operation

1: Set shim password
2: Clear shim password
3: Set Remote Loader password
4: Clear Remote Loader password
5: Set named password
6: Clear named password(s)
7: List named passwords
8: Get passwords state
99: Exit

Enter choice:
```

Table A-3 Password Operations

Operation	Description
1: <i>Set shim password</i>	Sets the application password. This is the password of the user account you are using to authenticate into the connected system with.
2: <i>Clear shim password</i>	Clears the application password.
3: <i>Set Remote Loader password</i>	The Remote Loader password is used to control access to the Remote Loader instance. Enter the Remote Loader password, then confirm the password by typing it again.
4: <i>Clear Remote Loader password</i>	Clears the Remote Loader password so no Remote Loader password is set on the Driver object.
5: <i>Set named password</i>	Allows you to store a password or other pieces of security information on the driver. See Section 7.9, "Storing Driver Passwords Securely with Named Passwords," on page 46 for more information. Lists four prompts: <ul style="list-style-type: none"> ◆ <i>Enter password name:</i> ◆ <i>Enter password description:</i> ◆ <i>Enter password:</i> ◆ <i>Confirm password:</i>

Operation	Description
6: <i>Clear named passwords</i>	<p>Clears a specified named password or all named passwords that are stored on the Driver object: <i>Do you want to clear all named passwords? (yes/no)</i>.</p> <p>If you enter <i>Yes</i>, all Named Passwords are cleared. If you enter <i>No</i>, you are prompted to specify the password name that you want to clear.</p>
7: <i>List named passwords</i>	Lists all named passwords that are stored on the Driver object. It lists the password name and the password description.
8: <i>Get password state</i>	<p>Lists if a password is set for:</p> <ul style="list-style-type: none"> ◆ Driver Object password ◆ Application password ◆ Remote loader password <p>The dxcmnd utility enables you to set the Application password and the Remote Loader password. You cannot set the Driver Object password with this utility. It displays whether the password has been set.</p>
99: <i>Exit</i>	Exits the current menu and takes you back to the Driver options.

Figure A-3 Cache Operations

```

Enter choice: 14

Select a cache operation

1: Get driver cache limit
2: Set driver cache limit
3: View cached transactions
4: Delete cached transactions
99: Exit

Enter choice:

```

Table A-4 Cache Operations

Operation	Description
1: <i>Get driver cache limit</i>	Displays the current cache limit that is set for the driver.
2: <i>Set driver cache limit</i>	Sets the driver cache limit in kilobytes. A value of 0 is unlimited.

Operation	Description
3: <i>View cached transactions</i>	<p>A text file is created with the events that are stored in cache. You can select the number of transactions to view.</p> <ul style="list-style-type: none"> ◆ <i>Enter option token (default=0):</i> ◆ <i>Enter maximum transactions records to return (default=1):</i> ◆ <i>Enter name of file for response:</i>
4: <i>Delete cached transactions</i>	<p>Deletes the transactions stored in cache.</p> <ul style="list-style-type: none"> ◆ <i>Enter position token (default=0):</i> ◆ <i>Enter event-id value of first transaction record to delete (optional):</i> ◆ <i>Enter number of transaction records to delete (default=1):</i>
99: <i>Exit</i>	Exits the current menu and takes you back to the Driver options.

Figure A-4 Log Event Operations

```

Select a log events operation

1: Set driver set log events
2: Reset driver set log events
3: Set driver log events
4: Reset driver log events
99: Exit

Enter choice:

```

Table A-5 Log Events Operations

Operation	Description
1: <i>Set driver set log events</i>	<p>Allows you to log driver set events through Novell Audit. You can select 49 items to log. See Table A-6 on page 82 for a list of these options.</p> <p>Type the number of the item you want to log. After the items are selected, enter 99 to accept the selections.</p>
2: <i>Reset driver set log events</i>	Resets all log event options.
3: <i>Set driver log events</i>	<p>Allows you to log driver events through Novell Audit. You can select 49 items to log. See Table A-6 on page 82 for a list of these options.</p> <p>Type the number of the item you want to log. After the items are selected, enter 99 to accept the selections.</p>

Operation	Description
4: <i>Reset driver log events</i>	Resets all of the log event options.
99: <i>Exit</i>	Exits the log events operations menu.

Table A-6 *Driver Set and Driver Log Events*

Options
1: Status success
2: Status retry
3: Status warning
4: Status error
5: Status fatal
6: Status other
7: Query elements
8: Add elements
9: Remove elements
10: Modify elements
11: Rename elements
12: Move elements
13: Add-association elements
14: Remove-association elements
15: Query-schema elements
16: Check-password elements
17: Check-object-password elements
18: Modify-password elements
19: Sync elements
20: Pre-transformed XDS document from shim
21: Post input transformation XDS document
22: Post output transformation XDS document
23: Post event transformation XDS document
24: Post placement transformation XDS document
25: Post create transformation XDS document
26: Post mapping transformation <inbound> XDS document
27: Post mapping transformation <outbound> XDS document

Options

- 28: Post matching transformation XDS document
 - 29: Post command transformation XDS document
 - 30: Post-filtered XDS document <Publisher>
 - 31: User agent XDS command document
 - 32: Driver resync request
 - 33: Driver migrate from application
 - 34: Driver start
 - 35: Driver stop
 - 36: Password sync
 - 37: Password request
 - 38: Engine error
 - 39: Engine warning
 - 40: Add attribute
 - 41: Clear attribute
 - 42: Add value
 - 43: Remove value
 - 44: Merge entire
 - 45: Get named password
 - 46: Reset Attributes
 - 47: Add Value - Add Entry
 - 48: Set SSO Credential
 - 49: Clear SSO Credential
 - 50: Set SSO Passphrase
 - 51: User defined IDs
 - 99: Accept checked items
-

Table A-7 Job Scheduler Operations

Options	Description
1: <i>Get available job definitions</i>	Allows you to select an existing job. <ul style="list-style-type: none">◆ <i>Enter the job number:</i>◆ <i>Do you want to filter the job definitions by containment? Enter Yes or No</i>◆ <i>Enter name of the file for response:</i> Examples: NetWare: <code>sys:\files\user.log</code> Windows: <code>c:\files\user.log</code> Linux: <code>/files/user.log</code>
2: <i>Operations on specific job object</i>	Allows you to perform operations for a specific job.

A.2 Command Line Mode

The command line mode allows you to use script or batch files. [Table A-8 on page 84](#) lists the different options that are available.

To use the command line options, decide which items you want to use and string them together.

Example: `dxcmd -user admin.headquarters -host 10.0.0.1 -password n0vell -start test.driverset.headquarters`

This example command starts the driver.

Table A-8 Command Line Options

Option	Description
Configuration	
<code>-user <user name></code>	Specify the name of a user with administrative rights to the drivers you want to test.
<code>-host <name or IP address></code>	Specify the IP address of the server where the driver is installed.
<code>-password <user password></code>	Specify the password of the user specified above.
<code>-port <port number></code>	Specify a port number, if the default port is not used.
<code>-q <quiet mode></code>	Displays very little information when a command is executed.
<code>-v <verbose mode></code>	Displays detailed information when a command is executed.
<code>-s <stdout></code>	Writes the results of the <code>dxcmd</code> command to <code>stdout</code> .

Option	Description
-? <show this message>	Displays the help menu.
-help <show this message>	Displays the help menu.
Actions	
-start <driver dn>	Starts the driver.
-stop <driver dn>	Stops the driver.
-getstate <driver dn>	Shows the state of the driver as running or stopped.
-getstartoption <driver dn>	Shows the startup option of the driver.
-setstartoption <driver dn> <disabled manual auto> <resync noresync>	Sets how the driver starts if the server is rebooted. Sets whether the objects are to be resynchronized when the driver restarts.
-getcachelimit <driver dn>	Lists the cache limit set for the driver.
-setcachelimit <driver dn> <0 or positive integer>	Sets the cache limit for the driver.
-migrateapp <driver dn> <filename>	Processes an XML document that contains a query command. Create the XML document that contains a query command by using the Novell <code>nds.dtd</code> (http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_dtd/data/dtdndsoverview.html#dtdndsoverview).
-setshimpassword <driver dn> <password>	Sets the application password. This is the password of the user account you are using to authenticate into the connected system with.
-clearshimpassword <driver dn> <password>	Clears the application password.
-setremoteloaderpassword <driver dn> <password>	Sets the Remote Loader password. The Remote Loader password is used to control access to the Remote Loader instance.
<clearremoteloaderpassword <driver dn>	Clears the Remote Loader password.

Option	Description
-sendcommand <driver dn> <input filename> <output filename>	<p>Processes an XDS command document.</p> <p>Specify the XDS command document as the input file.</p> <p>Examples:</p> <p>NetWare: sys:\files\user.xml</p> <p>Windows: c:\files\user.xml</p> <p>Linux: /files/user.log</p> <p>Specify the output filename to see the results.</p> <p>Examples:</p> <p>NetWare: sys:\files\user.log</p> <p>Windows: c:\files\user.log</p> <p>Linux: /files/user.log</p>
-sendevent <driver dn> <input filename>	<p>Submits a document to the driver's Subscriber channel, bypassing the driver cache. The document is processed ahead of anything that might be in the cache at the time of the submission. It also means that the submission fails if the driver is not running.</p>
-queueevent <driver dn> <input filename>	<p>Submits a document to the driver's Subscriber channel by queuing the document in the driver cache. The document is processed after anything that might be in the cache at the time of the submission. The submission won't fail if the driver isn't running.</p>
-setlogevents <dn> <integer ...>	<p>Sets Novell Audit log events on the driver. The integer is the option of the item to log. See Table A-6 on page 82 for the list of the integers to enter.</p>
-clearlogevents <dn>	<p>Clears all Novell Audit log events that are set on the driver.</p>
-setdriverset <driver set dn>	<p>Associates a driver set with the server.</p>
-cleardriverset	<p>Clears the driver set association from the server.</p>
-getversion	<p>Shows the version of Identity Manager that is installed.</p>
-initdriver object <dn>	<p>Performs an internal initialization of data on a new Driver object. This is only for testing purposes.</p>
-setnamedpassword <driver dn> <name> <password> [description]	<p>Sets named passwords on the driver object. You specify the name, the password, and the description of the named password.</p>
-clearnamedpassword <driver dn> <name>	<p>Clears a specified named password.</p>
-startjob <job dn>	<p>Starts the specified job.</p>

Option	Description
-abortjob <job dn>	Stops the specified job.
-getjobrunningstate <job dn>	Returns the specified job's running state.
-getjobenabledstate <job dn>	Returns the specified job's enabled state.
-getjobnextruntime <job dn>	Returns the specified job's next run time.
-updatejob <job dn>	Updates the specified job.
-clearallnamedpasswords <driver dn>	Clears all named passwords set on a specific driver.

If a command is executed successfully, it returns a zero. If the command returns anything other than zero, it is an error. For example, 0 means success, and -641 means invalid operation. -641 is an eDirectory error code. [Table A-9 on page 87](#) contains other values for specific command line options.

Table A-9 *Command Line Option Values*


Command Line Option	Values
-getstate	0- stopped 1- starting 2- running 3- shutting down 11- get schema Anything else that is returned is an error.
-getstartoption	0- disabled 1- manual 2- auto Anything else that is returned is an error.
-getcachelimit	0- unlimited Anything else that is returned is an error.
-getjobrunningstate	0- stopped 1- running Anything else that is returned is an error.
-getjobenabledstate	0- disabled 1- enabled 2- configuration error Anything else that is returned is an error.

Command Line Option	Values
-getjobnextruntime	Returns the next scheduled time for the job in eDirectory time format (number of seconds since 00:00:00 Jan 1, 1970 UTC).

Properties of the eDirectory Driver

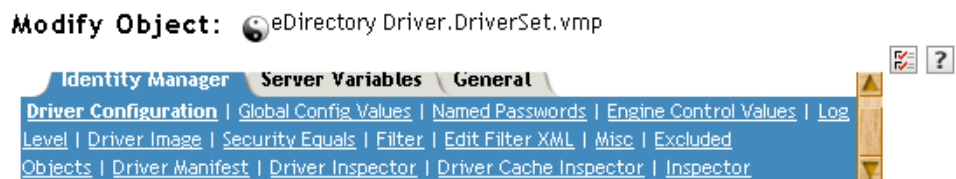
B

This section is a reference for all of the fields on the driver as displayed in iManager and Designer. Sometimes fields are displayed differently in iManager than in Designer.

The information is presented from the viewpoint of iManager. If a field is different in Designer for Identity Manager, it is marked with a Designer  icon.

The following figure illustrates property pages in iManager:

Figure B-1 Property Pages in iManager



- ◆ [Section B.1, “Driver Configuration,” on page 89](#)
- ◆ [Section B.2, “Global Configuration Values,” on page 95](#)
- ◆ [Section B.3, “Named Passwords,” on page 96](#)
- ◆ [Section B.4, “Engine Control Values,” on page 97](#)
- ◆ [Section B.5, “Log Level,” on page 99](#)
- ◆ [Section B.6, “Driver Image,” on page 100](#)
- ◆ [Section B.7, “Security Equals,” on page 100](#)
- ◆ [Section B.8, “Filter,” on page 100](#)
- ◆ [Section B.9, “Edit Filter XML,” on page 101](#)
- ◆ [Section B.10, “Misc,” on page 101](#)
- ◆ [Section B.11, “Excluded Users,” on page 102](#)
- ◆ [Section B.12, “Driver Manifest,” on page 102](#)
- ◆ [Section B.13, “Driver Cache Inspector,” on page 103](#)
- ◆ [Section B.14, “Driver Inspector,” on page 103](#)
- ◆ [Section B.15, “Server Variables,” on page 104](#)

B.1 Driver Configuration

In Designer:

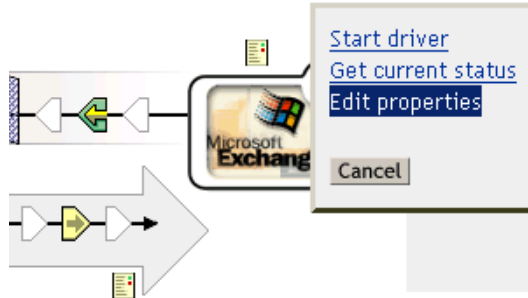
- 1 Open a project in the Modeler, then right-click the driver line.
- 2 Click *Properties > Driver Configuration*.

In iManager:

- 1 Click *Identity Manager > Identity Manager Overview*, then click *Search* to search for the driver set that is associated with the driver.

- 2 Select the eDirectory driver.
- 3 Click the driver's status indicator, in the upper right corner of the driver icon, then select *Edit Properties*.

Figure B-2 The Edit Properties Option



- 4 Click *Driver Configuration*.

To configure the eDirectory driver, set parameters on the following:

- ◆ [Section B.1.1, “Driver Module,” on page 90](#)
- ◆ [Section B.1.2, “Driver Object Password,” on page 91](#)
- ◆ [Section B.1.3, “Authentication,” on page 91](#)
- ◆ [Section B.1.4, “Startup Option,” on page 92](#)
- ◆ [Section B.1.5, “Driver Parameters,” on page 93](#)
- ◆ [Section B.1.6, “ECMAScript,” on page 95](#)



B.1.1 Driver Module

The driver module changes the driver from running locally to running remotely or the reverse.

In Designer:

- 1 Open a project in the Modeler.
- 2 Right-click the driver line, then select *Properties > Driver Configuration*.
- 3 Select the *Driver Module* tab.

Option	Description
<i>Java</i>	Used to specify the name of the Java class that is instantiated for the shim component of the driver. This class can be located in the <code>classes</code> directory as a class file, or in the <code>lib</code> directory as a <code>.jar</code> file. If this option is selected, the driver is running locally.
<i>Native</i>	Used to specify the name of the <code>.dll</code> file that is instantiated for the application shim component of the driver. If this option is selected, the driver is running locally.

Option	Description
<i>Connect to Remote Loader</i>	Used when the driver is connecting remotely to the connected system.
 <i>Remote Loader Client Configuration for Documentation</i>	 Includes information on the Remote Loader client configuration when Designer generates documentation for the Delimited Text driver.

In iManager:

- 1 Click *Identity Manager > Identity Manager Overview*, then click *Search* to search for the driver set that is associated with the driver.
- 2 Browse to the driver, then click the upper right corner of the driver icon.
- 3 Click *Edit Properties > Driver Configuration > Driver Module*.

B.1.2 Driver Object Password

In Designer:

- 1 Open a project in the Modeler.
- 2 Right-click the driver line, then click *Properties > Driver Configuration*.
- 3 Click *Driver Module > Connect to Remote Loader > Set Password*.

Option	Description
<i>Driver Object Password</i>	Use this option to set a password for the Driver object. If you are using the Remote Loader, you must enter a password on this page. Otherwise, the remote driver does not run. The Remote Loader uses this password to authenticate itself to the remote driver shim.

In iManager:











- 1 Click *Identity Manager > Identity Manager Overview*, then click *Search* to search for the driver set that is associated with the driver.
- 2 Browse to the driver, then click the upper right corner of the driver icon.
- 3 Click *Edit Properties > Driver Configuration > Driver Object Password > Set Password*.

B.1.3 Authentication

The authentication section stores the information required to authenticate to the connected system.

In Designer:

- 1 Open a project in the Modeler.
- 2 Right-click the driver line, then select *Properties > Driver Configuration*.
- 3 Click *Authentication*.

Option	Description
 Authentication information for server <i>Authentication ID</i> <i>Authentication Context</i> or  <i>Connection Information</i> <i>Remote Loader Connection Parameters</i> or  <i>Host name</i>  <i>Port</i>  <i>KMO</i>  <i>Other parameters</i>	Displays or specifies the server that the driver is associated with. Specify a user application ID. This ID is used to pass Identity Vault subscription information to the application. Example: <code>Administrator</code> Specify the IP address or name of the server that the application shim should communicate with. Used only if the driver is connecting to the application through the Remote Loader. The parameter to enter is <code>hostname=xxx.xxx.xxx.xxx port=xxxx</code> <code>kmo=certificatename</code> , when the host name is the IP address of the application server running the Remote Loader server and the port is the port the Remote Loader is listening on. The default port for the Remote Loader is 8090. The <code>kmo</code> entry is optional. It is used only when an SSL connection exists between the Remote Loader and the Metadirectory engine. Example: <code>hostname=10.0.0.1 port=8090</code> <code>kmo=IDMCertificate</code> Specify the maximum event cache file size (in KB). If it is set to zero, the file size is unlimited.  Click <i>Unlimited</i> to set the file size to unlimited in Designer. Specify the password for the user object listed in the <i>Authentication ID</i> field. Used only if the driver is connecting to the application through the Remote Loader. The password is used to control access to the Remote Loader instance. It must be the same password specified during the configuration of the Remote Loader on the connected system.
 <i>Cache limit (KB)</i> <i>Application Password</i> or  <i>Set Password</i> <i>Remote Loader Password</i> or  <i>Set Password</i>	

In iManager:


- 1 Click *Identity Manager > Identity Manager Overview*, then click *Search* to search for the driver set that is associated with the driver.
- 2 Browse to the driver, then click the upper right corner of the driver icon.
- 3 Click *Edit Properties > Driver Configuration > Authentication*.

B.1.4 Startup Option

The Startup Option enables you to set the driver state when the Identity Manager server is started.

In Designer:

- 1 Open a project in the Modeler.
- 2 Right-click the driver line, then select *Properties > Driver Configuration*.
- 3 Click *Startup Option*.

Option	Description
<i>Auto start</i>	The driver starts every time the Identity Manager server is started.
<i>Manual</i>	The driver does not start when the Identity Manager server is started. The driver must be started through Designer or iManager.
<i>Disabled</i>	The driver has a cache file that stores all of the events. When the driver is set to <i>Disabled</i> , this file is deleted and no new events are stored in the file until the driver state is changed to <i>Manual</i> or <i>Auto Start</i> .
 <i>Do not automatically synchronize the driver</i>	This option applies only if the driver is deployed and was previously disabled. If this is not selected, the driver re-synchronizes the next time it is started.

In iManager:

- 1 Click *Identity Manager > Identity Manager Overview*, then click *Search* to search for the driver set that is associated with the driver.
- 2 Browse to the driver, then click the upper right corner of the driver icon.
- 3 Click *Edit Properties > Driver Configuration > Startup Option*.

B.1.5 Driver Parameters

In Designer:

- 1 Open a project in the Modeler.
- 2 Right-click the driver line, then select *Properties > Driver Configuration*.
- 3 Click *Driver Parameters*.


Modify parameters. For information on settings, see [Table B-1 on page 94](#).

If no fields display on the Driver options, Subscriber options, or Publisher options tabs, see [Section 9.2, “Creating Functionality on the Driver Parameters Page,” on page 69](#).

In iManager:

- 1 Click *Identity Manager > Identity Manager Overview*, then click *Search* to search for the driver set that is associated with the driver.
- 2 Browse to the driver, then click the upper right corner of the driver icon.
- 3 Click *Edit Properties > Driver Configuration > Driver Parameters*.

Table B-1 Settings: Driver Parameters

Parameter	Description
 Driver parameters for server	Displays or specifies the server name or IP address of the server whose driver parameters you want to modify.
Edit XML button	Opens an editor so that you can edit the driver's configuration file.
Driver Options	
<i>SSL type</i>	Specifies whether to use a Key Material Object (KMO) for SSL or use a Java keystore file. For more information, click the <i>Information</i> icon.
Subscriber Options	
<i>Address or host name of remote publisher</i>	Specifies the IP address or DNS name of the server hosting the remote eDir-to-eDir driver that the local subscriber connects to.
<i>TCP port of remote publisher</i>	If the remote publisher options specify a TCP port, then this must be set to <i>specify</i> and the value from the remote Publisher channel entered into the <i>Port number</i> field. (These two fields must match what is set in the remote Publisher channel's options, which have corresponding fields).
<i>Port number</i>	Specifies the port number that the remote publisher is configured to run on. Displays only if you select <i>specify</i> in the <i>TCP port of remote publisher</i> field.
<i>Advanced options</i>	Displays additional fields when you select <i>show</i> .
<i>Socket local bind</i>	The <i>local bind</i> fields specify which IP address the Subscriber channel's socket will be bound to. This is generally only useful if the server has more than one IP address and it is important to bind to a particular address because of firewall settings.
Local bind address for subscriber socket	The <i>local bind</i> fields specify which IP address the Subscriber channel's socket will be bound to. This is generally only useful if the server has more than one IP address and it is important to bind to a particular address because of firewall settings.
Receive timeout in minutes	In order to detect a lost TCP/IP connection, the eDir-to-eDir driver periodically sends small packets. This value determines how long since entering a receive-wait condition the Subscriber channel waits until sending a "keep-alive" packet to determine if the TCP/IP connection has been lost. Generally, do not change this value except under instruction from Novell. The default value for the Subscriber channel is one minute.
Publisher Options	
<i>Periodic heartbeat documents</i>	Turns the Publisher channel heartbeat on or off. (Heartbeat is a method that can be used to periodically cause something to happen in a policy on the Publisher channel).
<i>Heartbeat interval (in minutes)</i>	If the heartbeat is on, this setting specifies how often the local Publisher channel sends a heartbeat document to the engine.

Parameter	Description
<i>Local bind address for publisher socket</i>	Specifies which IP address the Subscriber channel's socket will be bound to. This is generally only useful if the server has more than one IP address and it is important to bind to a particular address because of firewall settings. This setting applies to the local publisher's "server" socket on which the local publisher listens for connections from the remote Subscriber channel.
<i>Receive timeout in minutes</i>	In order to detect a lost TCP/IP connection, the eDir-to-eDir driver periodically sends small packets. This value determines how long since entering a receive-wait condition the Publisher channel waits until sending a "keep-alive" packet to determine if the TCP/IP connection has been lost. Generally, do not change this value except under instruction from Novell. The default value for the Publisher channel is ten minutes.

B.1.6 ECMAScript

Enables you to add ECMAScript resource files. The resources extend the driver's functionality when Identity Manager starts the driver.

B.2 Global Configuration Values

Global configuration values (GCVs) enable you to specify settings for the Identity Manager features such as password synchronization and driver heartbeat, as well as settings that are specific to the function of an individual driver configuration. Some GCVs are provided with the drivers, but you can also add your own.

IMPORTANT: Password synchronization settings are GCVs, but it's best to edit them in the graphical interface provided on the Server Variables page for the driver, instead of the GCV page. The Server Variables page that shows Password Synchronization settings is accessible as a tab as with other driver parameters, or by clicking *Password Management > Password Synchronization*, searching for the driver, and clicking the driver name. The page contains online help for each Password Synchronization setting.

In iManager:

- 1 Click *Identity Manager > Identity Manager Overview*, then click *Search* to search for the driver set that is associated with the driver.
- 2 Browse to the driver, then click the upper right corner of the driver icon.
- 3 Click *Edit Properties > Global Config Values*.

In Designer:

- 1 Open a project in the Modeler.
- 2 Right-click the driver line, then select *Properties > Global Config Values*.

Table B-2 Global Configuration Values > Password Configuration

Option	Description
<i>Application accepts passwords from Identity Manager</i>	If True, allows passwords to flow from the Identity Manager data store to the connected system.
<i>Identity Manager accepts passwords from application</i>	If True, allows passwords to flow from the connected system to Identity Manager.
<i>Publish passwords to NDS password</i>	Use the password from the connected system to set the non-reversible NDS® password in eDirectory.
<i>Publish passwords to Distribution Password</i>	Use the password from the connected system to set the NMAST™ Distribution Password used for Identity Manager password synchronization.
<i>Require password policy validation before publishing passwords</i>	If True, applies NMAST password policies during publish password operations. The password is not written to the data store if it does not comply.
<i>Reset user's external system password to the Identity Manager password on failure</i>	If True, on a publish Distribution Password failure, attempt to reset the password in the connected system by using the Distribution Password from the Identity Manager data store.
<i>Notify the user of password synchronization failure via e-mail</i>	If True, notify the user by e-mail of any password synchronization failures.
<i>Connected System or Driver Name</i>	The name of the connected system, application, or Identity Manager driver. This value is used by the e-mail notification templates.

B.3 Named Passwords

Identity Manager enables you to store multiple passwords securely for a particular driver. This functionality is referred to as Named Passwords. Each different password is accessed by a key, or name.

You can also use the Named Passwords feature to store other pieces of information securely, such as a user name. To configure Named Passwords, see [Section 7.9, “Storing Driver Passwords Securely with Named Passwords,” on page 46](#).

In iManager:

- 1 Click *Identity Manager > Identity Manager Overview*, then click *Search* to search for the driver set that is associated with the driver.
- 2 Browse to the driver, then click the upper right corner of the driver icon.
- 3 Click *Edit Properties > Named Passwords*.

In Designer:

- 1 Open a project in the Modeler.
- 2 Right-click the driver line, then select *Properties > Named Passwords*.

B.4 Engine Control Values

The engine control values are a means through which certain default behaviors of the Metadirectory engine can be changed. The values can only be accessed if a server is associated with the Driver Set object.

In iManager:

- 1 Click *Identity Manager > Identity Manager Overview*, then click *Search* to search for the driver set that is associated with the driver.
- 2 Browse to the driver, then click the upper right corner of the driver icon.
- 3 Click *Edit Properties > Engine Control Values*.

In Designer:

- 1 In the Modeler, right-click a driver line.
- 2 Select *Properties > Engine Control Values*.
- 3 Click the tooltip icon to the right of the *Engine Controls for Server* field. If a server is associated with the Identity Vault, the Engine Control Values display in the large pane.

Table B-3 *Engine Control Values*

Option	Description
<i>Subscriber channel retry interval in seconds</i>	The Subscriber channel retry interval controls how frequently the Metadirectory engine retries the processing of a cached transaction after the application shim's Subscriber object returns a retry status.
<i>Qualified form for DN-syntax attribute values</i>	The qualified specification for DN-syntax attribute values controls whether values for DN-syntax attribute values are presented in unqualified slash form or qualified slash form. A True setting means the values are presented in qualified form.
<i>Qualified form from rename events</i>	The qualified form for rename events controls whether the new-name portion of rename events coming from the Identity Vault are presented to the Subscriber channel with type qualifiers. For example, CN=. A True setting means the names are presented in qualified form.
<i>Maximum eDirectory replication wait time in seconds</i>	This setting controls the maximum time that the Metadirectory engine waits for a particular change to replicate between the local replica and a remote replica. This only affects operations where the Metadirectory engine is required to contact a remote eDirectory server in the same tree to perform an operation and might need to wait until some change has replicated to or from the remote server before the operation can be completed (for example, object moves when the Identity Manager server does not hold the master replica of the moved object; file system rights operations for Users created from a template.)

Option	Description
<i>Use non-compliant backwards-compatible mode for XSLT</i>	<p>This control sets the XSLT processor used by the Metadirectory engine to a backwards-compatible mode. The backward-compatible mode causes the XSLT processor to use one or more behaviors that are not XPath 1.0 and XSLT 1.0 standards-compliant. This is done in the interest of backward compatibility with existing DirXML[®] style sheets that depend on the non-standard behaviors.</p> <p>For example, the behavior of the XPath “!=” operator when one operand is a node-set and the other operand is other than a node-set is incorrect in DirXML releases up to and including Identity Manager 2.0. This behavior has been corrected; however, the corrected behavior is disabled by default through this control in favor of backward compatibility with existing DirXML style sheets.</p>
<i>Maximum application objects to migrate at once</i>	<p>This control is used to limit the number of application objects that the Metadirectory engine requests from an application during a single query that is performed as part of a Migrate Objects from Application operation.</p> <p>If java.lang.OutOfMemoryError errors are encountered during a Migrate from Application operation, this number should be set lower than the default. The default is 50.</p>
<i>Set creatorsName on objects created in Identity Vault</i>	<p>NOTE: This control does not limit the number of application objects that can be migrated; it merely limits the batch size.</p> <p>This control is used by the Identity Manager engine to determine if the creatorsName attribute should be set to the DN of this driver on all objects created in the Identity Vault by this driver.</p> <p>Setting the creatorsName attribute allows for easily identifying objects created by this driver, but also carries a performance penalty. If not set, the creatorsName attribute defaults to the DN of the NCP[™] Server object that is hosting the driver.</p>
<i>Write pending associations</i>	<p>This control determines whether the Identity Manager engine writes a pending association on an object during Subscriber channel processing.</p> <p>Writing a pending association confers little or no benefit but does incur a performance penalty. Nevertheless, the option exists to turn it on for backward compatibility.</p>
<i>Use password event values</i>	<p>This control determines the source of the value reported for the nspmDistributionPassword attribute for Subscriber channel Add and Modify events.</p> <p>Setting the control to False means that the current value of the nspmDistributionPassword is obtained and reported as the value of the attribute event. This means that only the current password value is available. This is the default behavior.</p> <p>Setting the control to True means that the value recorded with the eDirectory event is decrypted and is reported as the value of the attribute event. This means that both the old password value (if it exists) and the replacement password value at the time of the event are available. This is useful for synchronizing passwords to certain applications that require the old password to enable setting a new password.</p>

Option	Description
<i>Enable password synchronization status reporting</i>	This control determines whether the Identity Manager engine reports the status of Subscriber channel password change events. Reporting the status of Subscriber channel password change events allows applications such as the Identity Manager User Application to monitor the synchronization progress of a password change that should be synchronized to the connected application.

B.5 Log Level

Each driver set and each driver has a log level field where you can define the level of errors that should be tracked. The level you indicate here determines which messages are available to the logs. By default, the log level is set to track error messages. (This also includes fatal messages.) To track additional message types, change the log level.


Novell® recommends that you use Novell Audit instead of setting the log levels. See the *Identity Manager 3.5.1 Logging and Reporting*.

In iManager:

- 1 Click *Identity Manager > Identity Manager Overview*, then click *Search* to search for the driver set that is associated with the driver.
- 2 Browse to the driver, then click the upper right corner of the driver icon.
- 3 Click *Edit Properties > Log Level*.

In Designer:

- 1 Open a project in the Modeler.
- 2 Right-click the driver line, then select *Properties > Driver Log Level*.

Option	Description
<i>Use log settings from the DriverSet</i>	If this is selected, the driver logs events as the options are set on the Driver Set object.
<i>Log errors</i>	Logs just errors.
<i>Log errors and warnings</i>	Logs errors and warnings.
<i>Log specific events</i>	Logs the events that are selected. Click the  icon to see a list of the events.
<i>Only update the last log time</i>	Updates the last log time.
<i>Logging off</i>	Turns logging off for the driver.
<i>Turn off logging to DriverSet, Subscriber and Publisher logs</i>	If selected, turns all logging off for this driver on the Driver Set object, Subscriber channel, and the Publisher channel.
<i>Maximum number of entries in the log (50-500)</i>	Number of entries in the log. The default value is 50.

B.6 Driver Image

Allows you to change the image associated with the driver. You can browse and select a different image from the default image.

The image associated with a driver is used by the Identity Manager Overview plug-in when showing the graphical representation of your Identity Manager configuration. Although storing an image is optional, it makes the overview display more intuitive.

NOTE: The driver image is maintained when a driver configuration is exported.

In iManager:

- 1 Click *Identity Manager > Identity Manager Overview*, then click *Search* to search for the driver set that is associated with the driver.
- 2 Browse to the driver, then click the upper right corner of the driver icon.
- 3 Click *Edit Properties > Driver Image*.

In Designer:

- 1 Open a project in the Modeler.
- 2 Right-click the driver line, then select *Properties > iManager Icon*.

B.7 Security Equals

Use the Security page to view or change the list of objects that the driver is explicitly security equivalent to. This object effectively has all rights of the listed objects.

If you add or delete an object in the list, the system automatically adds or deletes this object in that object's "Security Equal to Me" property. You don't need to add the [Public] trustee or the parent containers of this object to the list, because this object is already implicitly security equivalent to them.

In iManager:

- 1 Click *Identity Manager > Identity Manager Overview*, then click *Search* to search for the driver set that is associated with the driver.
- 2 Browse to the driver, then click the upper right corner of the driver icon.
- 3 Click *Edit Properties > Security Equals*.

Designer does not list the users the driver is security equals to.

B.8 Filter

Launches the Filter editor. You can edit the Filter from this tab.

In iManager:

- 1 Click *Identity Manager > Identity Manager Overview*, then click *Search* to search for the driver set that is associated with the driver.
- 2 Browse to the driver, then click the upper right corner of the driver icon.

- 3 Click *Edit Properties > Filter*.

In Designer:

- 1 In an open project, click the *Outline* tab (Outline view).
- 2 Select the driver you want to manage the filter for, then click the plus sign to the left.
- 3 Double-click the *Filter* icon to launch the Filter Editor.

B.9 Edit Filter XML

Allows you to edit the filter directly in XML instead of using the Filter Editor.

In iManager:

- 1 Click *Identity Manager > Identity Manager Overview*, then click *Search* to search for the driver set that is associated with the driver.
- 2 Browse to the driver, then click the upper right corner of the driver icon.
- 3 Click *Edit Properties > Filter*.

In Designer:

- 1 In an open project, click the *Outline* tab (Outline view).
- 2 Select the driver you want to manage the filter for, then click the plus sign to the left.
- 3 Double-click the *Filter* icon to launch the Filter Editor, then click *XML Source* at the bottom of the Filter Editor.

B.10 Misc

Allows you to add a trace level to your driver. With the trace level set, DSTrace displays the Identity Manager events as the Metadirectory engine processes the events. The trace level affects only the driver it is set for. Use the trace level for troubleshooting issues with the driver when the driver is deployed. DSTrace displays the output of the specified trace level.


In iManager:

- 1 Click *Identity Manager > Identity Manager Overview*, then click *Search* to search for the driver set that is associated with the driver.
- 2 Browse to the driver, then click the upper right corner of the driver icon.
- 3 Click *Edit Properties > Misc*.

In Designer:

- 1 Open a project in the Modeler.
- 2 Right-click the driver line, then select *Properties > Trace*.

Option	Description
<i>Trace level</i>	Increases the amount of information displayed in DSTrace. Trace level 1 shows errors, but not the cause of the errors. If you want to see password synchronization information, set the trace level to 5.

Option	Description
<i>Trace file</i>	When a value is set in this field, all Java information for the driver is written to the file. The value for this field is the path for that file. As long as the file is specified, Java information is written to this file. If you do not need to debug Java, leave this field blank.
<i>Trace file size limit</i>	Allows you to set a limit for the Java trace file. If you set the file size to Unlimited, the file grows in size until there is no disk space left.
<i>Trace name</i>	Driver trace messages are prepended with the value entered in this field.
 <i>Use setting from Driver Set</i>	This option is only available in Designer. It allows the driver to use the same setting that is set on the Driver Set object.

B.11 Excluded Users

Use this page to create a list of users or resources that are not replicated to the application. Novell recommends that you add all objects that represent an administrative role to this list (for example, the Admin object).

In iManager:

- 1 Click *Identity Manager > Identity Manager Overview*, then click *Search* to search for the driver set that is associated with the driver.
- 2 Browse to the driver, then click the upper right corner of the driver icon.
- 3 Click *Edit Properties > Excluded Users*.

Designer does not list the excluded users.

B.12 Driver Manifest

The driver manifest is like a resumé for the driver. It states what the driver supports, and includes a few configuration settings. The driver manifest is created by default when the Driver object is imported. A network administrator usually does not need to edit the driver manifest.

In iManager:

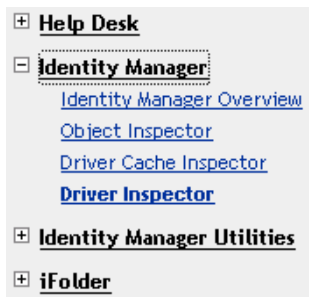
- 1 Click *Identity Manager > Identity Manager Overview*, then click *Search* to search for the driver set that is associated with the driver.
- 2 Browse to the driver, then click the upper right corner of the driver icon.
- 3 Click *Edit Properties > Driver Manifest*.

In Designer:

- 1 Open a project in the Modeler.
- 2 Right-click the driver line, then select *Properties > Driver Manifest*.

B.13 Driver Cache Inspector

Figure B-3 The Link to the Driver Cache Inspector



The Driver Cache Inspector page uses a table format to display information about the cache file that stores events while the driver is stopped.

- ◆ **Driver:** A link to run the *Driver Overview* on the driver that is associated with this cache file.
- ◆ **Driver Set:** A link to run the *Driver Set Overview* on the driver set that holds the driver.
- ◆ **Driver's cache on:** Lists the server object that contains this instance of the cache file.
- ◆ **Start/Stop Driver icons:** Displays the current state of the driver and allows you to start or stop the driver.
- ◆ **Edit icon:** Enables you to edit the properties of the currently selected Server object.
- ◆ **Delete:** Deletes the selected items from the cache file.
- ◆ **Refresh:** Enables you to re-read the cache file and refresh the displayed information.
- ◆ **Show:** Limits the number of items to be displayed. The options are:
 - ◆ 25 per page
 - ◆ 50 per page
 - ◆ 100 per page
 - ◆ Other: Enables you to specify a desired number.
- ◆ **Actions:** Enables you to perform actions on the entries in the cache file. Click *Actions* to expand the menu, which includes:
 - ◆ **Expand All:** Expands all of the entries displayed in the cache file.
 - ◆ **Collapse All:** Collapses all of the entries displayed in the cache file.
 - ◆ **Go To:** Enables you to access a specified entry in the cache file. Specify the entry number, then click *OK*.
 - ◆ **Cache Summary:** Summarizes all events stored in the cache file.

B.14 Driver Inspector

The Driver Inspector page displays information about objects associated with the driver. Designer does not have this option.

- ◆ **Driver:** A link to run the *Driver Overview* on the driver that is being inspected.
- ◆ **Driver Set:** A link to run the *Driver Set Overview* of the driver set that holds the driver.

- ◆ **Delete:** Deletes the associations of the selected objects.
- ◆ **Refresh:** Enables you to re-read all of the objects associated with the driver and refresh the displayed information.
- ◆ **Actions:** Enables you to perform actions on the objects associated with the driver. Click *Actions* to expand the menu, which includes:
 - ◆ **Show All Associations:** Displays all objects associated with the driver.
 - ◆ **Filter for Disabled Associations:** Displays all the driver's associated objects that have a Disabled state.
 - ◆ **Filter for Manual Associations:** Displays all the driver's associated objects that have a Manual state.
 - ◆ **Filter for Migrate Associations:** Displays all the driver's associated objects that have a Migrate state.
 - ◆ **Filter for Pending Associations:** Displays all the driver's associated objects that have a Pending state.
 - ◆ **Filter for Processed Associations:** Displays all the driver's associated objects that have a Processed state.
 - ◆ **Filter for Undefined Associations:** Displays all the driver's associated objects that have an Undefined state.
 - ◆ **Association Summary:** Displays the state of all objects associated with the driver.
- ◆ **Object DN:** Displays the DN of the associated objects.
- ◆ **State:** Displays the association state of the object.
- ◆ **Object ID:** Displays the value of the association.

B.15 Server Variables

This page lets you enable and disable Password Synchronization and the associated options for the selected driver.

When setting up Password Synchronization, consider both the settings on this page for an individual driver and the Universal Password Configuration options in your password policies.

This page lets you control which password Identity Manager updates directly, either the Universal Password for an Identity Vault, or the Distribution Password used for password synchronization by Identity Manager.

However, Novell Modular Authentication Service (NMAS) controls whether the various passwords inside the Identity Vault are synchronized with each other. Password Policies are enforced by NMAS, and they include settings for synchronizing Universal Password, NDS Password, Distribution Password, and Simple Password.

To change these settings in iManager:

- 1 In iManager, select *Passwords > Password Policies*.
- 2 Select a password policy, then click *Edit*.
- 3 Select *Universal Password*.

This option is available from a drop-down list or a tab, depending on your version of iManager and your browser.

4 Select *Configuration Options*, make changes, then click *OK*.

NOTE: Enabling or disabling options on this page corresponds to values of True or False for certain global configuration values (GCVs) used for password synchronization in the driver parameters. Novell recommends that you edit them here in the graphical interface, instead of on the GCVs page. This interface helps ensure that you don't set conflicting values for the password synchronization GCVs.

Option	Description
<i>Identity Manager accepts password (Publisher Channel)</i>	<p>If this option is enabled, Identity Manager allows passwords to flow from the connected system driver into the Identity Vault data store.</p> <p>Disabling this option means that no <password> elements are allowed to flow to Identity Manager. They are stripped out of the XML by a password synchronization policy on the Publisher channel.</p> <p>If this option is enabled, and the option below it for Distribution Password is disabled, a <password> value coming from the connected system is written directly to the Universal Password in the Identity Vault if it is enabled for the user. If the user's password policy does not enable Universal Password, the password is written to the NDS Password.</p>
<i>Use Distribution Password for password synchronization</i>	<p>To use this setting, you must have a version of eDirectory that supports Universal Password, regardless of whether you have enabled Universal Password in your password policies.</p> <p>If this option is enabled, a password value coming from the connected system is written to the Distribution Password. The Distribution Password is reversible, which means that it can be retrieved from the Identity Vault data store for password synchronization. It is used by Identity Manager for bidirectional password synchronization with connected systems. For Identity Manager to distribute passwords to connected systems, this option must be enabled.</p> <p>NMAS and Password policies control whether the Distribution Password is synchronized with other passwords in the Identity Vault. By default, the Distribution Password is the same as the Universal Password in the Identity Vault.</p> <p>If the password in the Identity Vault is to be independent of Password Synchronization, so that Identity Manager is a conduit only for synchronizing passwords among connected systems, change this default setting. In the Universal Password Configuration Options in a Password policy, disable <i>Synchronize Universal Password with Distribution Password</i>. This use of Identity Manager Password Synchronization is also referred to as "tunneling."</p>

Option	Description
<i>Accept password only if it complies with user's Password Policy</i>	<p>To use this setting, users must have a Password policy assigned that has Universal Password enabled, and Advanced Password Rules enabled and configured.</p> <p>If this option is chosen, Identity Manager does not write a password from this connected system to the Distribution Password in the Identity Manager data store or publish it to connected systems unless the password complies with the user's Password policy.</p> <p>By using the notification option that is also on this page, you can inform users when a password is not set because it is not compliant.</p>
<i>If password does not comply, ignore Password Policy on the connected system by resetting user's password to the Distribution Password</i>	<p>This option lets you enforce Password policies on the connected system by replacing a password that does not comply. If you select this option, and a user's password on the connected system does not comply with the user's Password policy, Identity Manager resets the password on the connected system by using the Distribution Password from the Identity Vault data store.</p> <p>Keep in mind that if you do not select this option, user passwords can become out-of-sync on connected systems.</p> <p>By using the notification option that is also on this page, you can inform users when a password is not set or reset. Notification is especially helpful for this option. If the user changes to a password that is allowed by the connected system but rejected by Identity Manager because of the Password policy, the user won't know that the password has been reset until the user receives a notification or tries to log in to the connected system with the old password.</p>
<i>Always accept password; ignore Password Policies</i>	<p>NOTE: Consider the connected system's password policies when deciding whether to use this option. Some connected systems might not allow the reset because they don't allow you to repeat passwords.</p> <p>If you select this option, Identity Manager does not enforce the user's Password policy for this connected system. Identity Manager writes the password from this connected system to the Distribution Password in the Identity Vault data store, and distributes it to other connected systems, even if the password does not comply with the user's Password policy.</p>

Option	Description
<i>Application accepts passwords (Subscriber Channel)</i>	<p>If you select this option, the driver sends passwords from the Identity Vault data store to this connected system. This also means that if a user changes the password on a different connected system that is publishing passwords to the Distribution Password in the Identity Vault data store, the password is changed on this connected system.</p> <p>By default, the Distribution Password is the same as the Universal Password in the Identity Vault, so changes to the Universal Password made in the Identity Vault are also sent to the connected system.</p> <p>If you want the password in the Identity Vault to be independent of Password Synchronization, so that Identity Manager is a conduit only for synchronizing passwords among connected systems, you can change this default setting. In the Universal Password Configuration Options in a password policy, disable <i>Synchronize Universal Password with Distribution Password</i>. This use of Password Synchronization is also referred to as “tunneling.”</p>
<i>Notify the user of password synchronization failure via-email</i>	<p>If you select this option, e-mail is sent to the user if a password is not synchronized, set, or reset. The e-mail that is sent to the user is based on an e-mail template. This template is provided by the Password Synchronization application. However, for the template to work, you must customize it and specify an e-mail server to send the notification messages.</p>
<p>NOTE: To set up e-mail notification, select <i>Passwords > Edit EMail Templates</i>.</p>	