

Novell Identity Manager Driver for LDAP

3.5.1

www.novell.com

IMPLEMENTATION GUIDE

October 15, 2007

N

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 Novell trademarks, see [Novell Trademark and Service List \(http://www.novell.com/company/legal/trademarks/tmlist.html\)](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 Synchronizing Data	11
1.2.2 Two Publication Methods	11
1.2.3 How the LDAP Driver Works	12
1.3 Driver Features	14
1.3.1 Local and Remote Platforms	14
1.3.2 Role-Based Entitlements	14
1.3.3 Password Synchronization	14
1.3.4 Data Synchronization	15
2 Installing the LDAP Driver	17
2.1 Planning Considerations	17
2.1.1 Prerequisites	17
2.1.2 Where to Install the LDAP Driver	17
2.1.3 Information to Gather	18
2.1.4 Assumptions about the LDAP Data Source	18
2.2 Upgrading to Identity Manager 3.5.1	19
2.3 Installing and Configuring Password Synchronization	19
2.3.1 Manually Adding Publisher Settings	21
2.4 Installing the Driver Separately	22
3 Upgrading the LDAP Driver	23
3.1 Upgrading the Driver by Using Designer	23
3.2 Upgrading the Driver by Using iManager	26
4 Importing an Example LDAP Configuration File	27
4.1 Using Designer to Import	27
4.2 Using iManager to Import	29
5 Configuring the LDAP Driver	31
5.1 Preparing the LDAP Server	31
5.1.1 Creating an LDAP User Object with Authentication Rights	31
5.1.2 Enabling the Change Log	32
5.2 Migrating and Resynchronizing Data	33
5.3 Controlling Data Flow from the LDAP Directory to an Identity Vault	34
5.3.1 LDAP Driver Settings	35
5.3.2 LDAP Subscriber Settings	35
5.3.3 LDAP Publisher Settings: Changelog and LDAP-Search Methods	36
5.3.4 LDAP Publisher Settings: Only the Changelog Method	37
5.3.5 LDAP Publisher Settings: Only the LDAP-Search Method	39
5.4 Synchronizing Data	41

5.4.1	Determining Which Objects Are Synchronized	41
5.4.2	Defining Schema Mapping	41
5.4.3	Defining Object Placement in Netscape Directory Server	42
5.4.4	Working with eDirectory Groups and Netscape	44
5.5	Configuring SSL Connections	44
5.5.1	Step 1: Generating a Server Certificate	44
5.5.2	Step 2: Sending the Certificate Request	45
5.5.3	Step 3: Installing the Certificate	45
5.5.4	Step 4: Activating SSL in Netscape Directory Server 4.12	46
5.5.5	Step 5: Exporting the Trusted Root from the Directory Tree	46
5.5.6	Step 6: Importing the Trusted Root Certificate	47
5.5.7	Step 7: Adjusting Driver Settings	48
6	Activating the LDAP Driver	49
7	Managing the LDAP Driver	51
7.1	Starting, Stopping, or Restarting the LDAP Driver	51
7.2	Migrating and Resynchronizing Data	51
7.3	Using the DirXML Command Line Utility	52
7.4	Viewing Driver Version Information	52
7.4.1	Viewing a Hierarchical Display of Version Information	52
7.4.2	Viewing the Version Information as a Text File	54
7.4.3	Saving Versioning Information	55
7.5	Reassociating a Driver Set Object with a Server Object	56
7.6	Changing the Driver Configuration	57
7.7	Storing Driver Passwords Securely with Named Passwords	57
7.7.1	Using Designer to Configure Named Passwords	58
7.7.2	Using iManager to Configure Named Passwords	58
7.7.3	Using Named Passwords in Driver Policies	60
7.7.4	Using the DirXML Command Line Utility to Configure Named Passwords	61
7.8	Adding a Driver Heartbeat	64
8	Synchronizing Objects	67
8.1	What Is Synchronization?	67
8.2	When Does Synchronization Occur?	67
8.3	How Does the Metadirectory Engine Decide Which Object to Synchronize?	68
8.4	How Synchronization Works	69
8.4.1	Scenario One	69
8.4.2	Scenario Two	71
8.4.3	Scenario Three	72
9	Troubleshooting	75
9.1	Troubleshooting Driver Processes	75
9.1.1	Viewing Driver Processes	75
9.2	Migrating Users into an Identity Vault	81
9.3	OutOfMemoryError	81
9.4	LDAP v3 Compatibility	82
9.5	Frequently Asked Questions	82
10	Backing Up the LDAP Driver	83
10.1	Exporting the Driver in Designer	83

10.2	Exporting the Driver in iManager	83
11	Security: Best Practices	85
A	The DirXML Command Line Utility	87
A.1	Interactive Mode	87
A.2	Command Line Mode	96
B	Properties of the LDAP Driver	101
B.1	Identity Manager: Driver Configuration	101
B.1.1	Driver Module	102
B.1.2	Driver Object Password	103
B.1.3	Authentication	103
B.1.4	Startup Option	104
B.2	Identity Manager: Global Configuration Values	105
B.3	Identity Manager: Named Passwords	106
B.4	Identity Manager: Engine Control Values	107
B.5	Identity Manager: Log Level	109
B.6	Identity Manager: Driver Image	110
B.7	Identity Manager: Security Equals	110
B.8	Identity Manager: Filter	110
B.9	Identity Manager: Edit Filter XML	111
B.10	Identity Manager: Misc.	111
B.11	Identity Manager: Excluded Users	112
B.12	Identity Manager: Driver Manifest	112
B.13	Driver Cache Inspector	113
B.14	Driver Inspector	113
B.15	Server Variables	114
C	Documentation Updates	117
C.1	October 15, 2007	117

About This Guide

This guide explains how to install, configure, and manage the Identity Manager Driver for LDAP.

- ◆ Chapter 1, “Overview,” on page 11
- ◆ Chapter 2, “Installing the LDAP Driver,” on page 17
- ◆ Chapter 3, “Upgrading the LDAP Driver,” on page 23
- ◆ Chapter 4, “Importing an Example LDAP Configuration File,” on page 27
- ◆ Chapter 5, “Configuring the LDAP Driver,” on page 31
- ◆ Chapter 6, “Activating the LDAP Driver,” on page 49
- ◆ Chapter 7, “Managing the LDAP Driver,” on page 51
- ◆ Chapter 8, “Synchronizing Objects,” on page 67
- ◆ Chapter 9, “Troubleshooting,” on page 75
- ◆ Chapter 10, “Backing Up the LDAP Driver,” on page 83
- ◆ Chapter 11, “Security: Best Practices,” on page 85

Audience

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

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 LDAP* 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 LDAP 3.5.1 (LDAP driver) synchronizes data between the Identity Vault and LDAP-compliant directories. The driver supports the Subscriber and Publisher channels, uses filters to control objects and attributes, and uses policies to control data.

- ◆ [Section 1.1, “What’s New?,” on page 11](#)
- ◆ [Section 1.2, “Driver Concepts,” on page 11](#)
- ◆ [Section 1.3, “Driver Features,” on page 14](#)

1.1 What’s New?

The LDAP Driver 3.5.1 for Identity Manager 3.5.1 has the ability to synchronize passwords on the Publisher channel when using Sun* Java* System Directory and a provided plug-in. See [Section 2.3, “Installing and Configuring Password Synchronization,” on page 19](#).

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, “Synchronizing Data,” on page 11](#)
- ◆ [Section 1.2.2, “Two Publication Methods,” on page 11](#)
- ◆ [Section 1.2.3, “How the LDAP Driver Works,” on page 12](#)

1.2.1 Synchronizing Data

The Identity Manager Driver for LDAP synchronizes data between an Identity Vault and LDAP-compliant directories. The driver can run anywhere that a Metadirectory server or Identity Manager Remote Loader is running. See [Section 1.3.1, “Local and Remote Platforms,” on page 14](#).

The driver uses the Lightweight Directory Access Protocol to bidirectionally synchronize changes between an Identity Vault and the connected LDAP-compliant directory.

Because of this flexible model for communicating, the driver can synchronize with LDAP-compliant directories running on platforms (for example, HP-UX*, OS/400*, and OS/390*) that are not supported by an Identity Vault.

1.2.2 Two Publication Methods

The driver can use either of two publication methods to recognize data changes and communicate them to an Identity Vault through Identity Manager:

- ◆ The changelog method

This method is preferred when a change log is available. Change logs are found on the following:

- ◆ Critical Path* InJoin* Directory

- ◆ IBM* SecureWay Directory
- ◆ IBM Tivoli* Directory
- ◆ iPlanet* Directory Server
- ◆ Netscape* Directory Server
- ◆ Oracle* Internet Directory
- ◆ Sun Java System Directory

See [Section 5.3.3, “LDAP Publisher Settings: Changelog and LDAP-Search Methods,”](#) on page 36 and [Section 5.3.4, “LDAP Publisher Settings: Only the Changelog Method,”](#) on page 37.

- ◆ The LDAP-search method

Some servers don't use the changelog mechanism. The LDAP-search method enables the LDAP driver to publish data about the LDAP server to an Identity Vault.

Additional software and changes to the LDAP-compliant directory are not required.

See [Section 5.3.5, “LDAP Publisher Settings: Only the LDAP-Search Method,”](#) on page 39.

1.2.3 How the LDAP Driver Works

Channels, filters, and policies control data flow.

- ◆ [“Publisher and Subscriber Channels”](#) on page 12
- ◆ [“Filters”](#) on page 12
- ◆ [“Policies”](#) on page 13

Publisher and Subscriber Channels

The LDAP driver supports Publisher and Subscriber channels:

- ◆ The Publisher channel reads information from the LDAP directory change log or an LDAP search and submits that information to an Identity Vault via the Metadirectory engine.

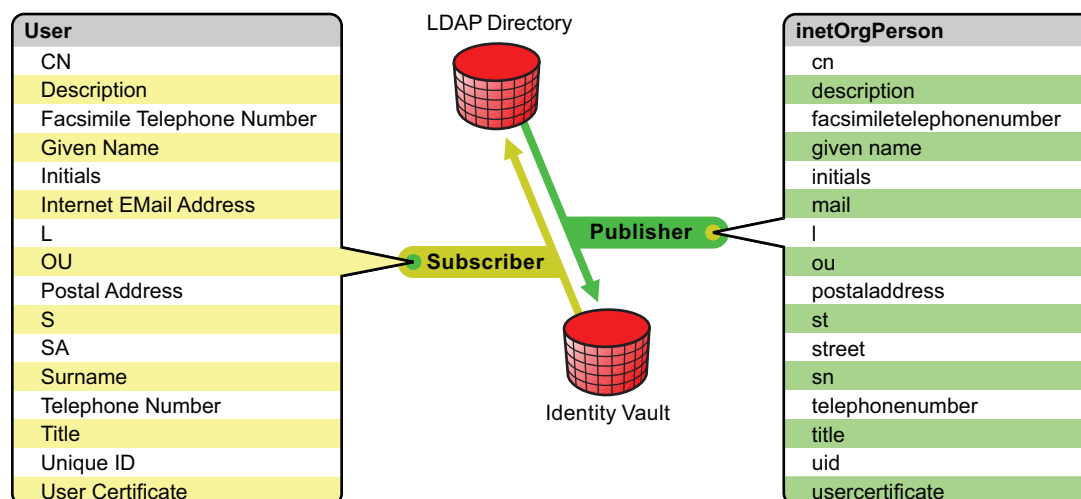
By default, the Publisher channel checks the log every 20 seconds, processing up to 1000 entries at a time, starting with the first unprocessed entry.

- ◆ The Subscriber channel watches for additions and modifications to Identity Vault objects and issues LDAP commands that make changes to the LDAP directory.

Filters

Identity Manager uses filters to control which objects and attributes are shared. The default filter configurations for the LDAP driver allow objects and attributes to be shared, as illustrated in the following figure:

Figure 1-1 LDAP Driver Filters



Policies

Policies are used to control data synchronization between the driver and an Identity Vault. The LDAP driver comes with two preconfiguration options to set up policies:

- ◆ The Flat option implements a flat structure for users in both directories.

With this configuration, when user objects are created in one directory, they are placed in the root of the container you specified during driver setup for the other directory. (The container name doesn't need to be the same in both the Identity Vault and the LDAP directory). When existing objects are updated, their context is preserved.

- ◆ The Mirror option matches the hierarchical structure in the directories.

With this configuration, when new user objects are created in one directory, they are placed in the matching hierarchical level of the mirror container in the other directory. When existing objects are updated, their context is preserved.

Except for the Placement policy and the fact that the Flat configuration doesn't synchronize Organizational Unit objects, the policies set up for these options are identical.

The following table provides information on default policies. These policies and the individual rules they contain can be customized through Novell® iManager as explained in [Chapter 5, “Configuring the LDAP Driver,”](#) on page 31.

Table 1-1 Default Policies

Policy	Description
Mapping	Maps the Identity Vault User object and selected properties to an LDAP inetOrgPerson. Maps the Identity Vault Organizational Unit to an LDAP organizationalUnit. By default, more than a dozen standard properties are mapped.

Policy	Description
Publisher Create	Specifies that in order for a User to be created in an Identity Vault, the cn, sn, and mail attributes must be defined. In order for an Organization Unit to be created, the OU attribute must be defined.
Publisher Placement	<p>With the Simple placement option, new User objects created in the LDAP directory are placed in the container in an Identity Vault that you specify when importing the driver configuration. The User object is named with the value of cn.</p> <p>With the Mirror placement option, new User objects created in the LDAP directory are placed in the Identity Vault container that mirrors the object's LDAP container.</p>
Matching	Specifies that a user object in an Identity Vault is the same object as an inetOrgPerson in the LDAP directory when the e-mail attributes match.
Subscriber Create	Specifies that in order for a user to be created in the LDAP directory, the CN, Surname, and Internet Email Address attributes must be defined. In order for an Organization Unit to be created, the OU attribute must be defined.
Subscriber Placement	<p>If you choose the Flat placement option during the import of the driver configuration, new User objects created in an Identity Vault are based on the value you specified during import.</p> <p>If you choose Mirrored placement during the import of the driver configuration, new User objects created in an Identity Vault are placed in the LDAP directory container that mirrors the object's Identity Vault container.</p>

1.3 Driver Features

- ◆ [Section 1.3.1, “Local and Remote Platforms,” on page 14](#)
- ◆ [Section 1.3.2, “Role-Based Entitlements,” on page 14](#)
- ◆ [Section 1.3.3, “Password Synchronization,” on page 14](#)
- ◆ [Section 1.3.4, “Data Synchronization,” on page 15](#)

1.3.1 Local and Remote Platforms

The LDAP driver runs in any Identity Manager 3.5.1 installation or Remote Loader installation. See [“Prerequisites to Installation”](#) in the *Identity Manager 3.5.1 Installation Guide*.

1.3.2 Role-Based Entitlements

The provided sample configuration for the LDAP driver supports “account” and “group membership” entitlements.

1.3.3 Password Synchronization

Passwords can be synchronized on the Subscriber channel either by issuing a `modify-password` command, or by mapping the `nspmDistributionPassword` attribute used with the Universal Password

feature to the userpassword attribute on the LDAP server. Passwords can be synchronized on the Publisher channel only with the Sun Java System Directory. See [Section 2.3, “Installing and Configuring Password Synchronization,”](#) on page 19.

1.3.4 Data Synchronization

See [Section 1.2.1, “Synchronizing Data,”](#) on page 11.

Installing the LDAP Driver

2

- ♦ [Section 2.1, “Planning Considerations,” on page 17](#)
- ♦ [Section 2.2, “Upgrading to Identity Manager 3.5.1,” on page 19](#)
- ♦ [Section 2.3, “Installing and Configuring Password Synchronization,” on page 19](#)
- ♦ [Section 2.4, “Installing the Driver Separately,” on page 22](#)

2.1 Planning Considerations

The LDAP Driver for Identity Manager works with most LDAP v3 compatible LDAP servers. The driver is written to the LDAP specification (IETF RFCs 4510-4519). For information on compatibility issues, see [Section 9.4, “LDAP v3 Compatibility,” on page 82](#).

- ♦ [Section 2.1.1, “Prerequisites,” on page 17](#)
- ♦ [Section 2.1.2, “Where to Install the LDAP Driver,” on page 17](#)
- ♦ [Section 2.1.3, “Information to Gather,” on page 18](#)
- ♦ [Section 2.1.4, “Assumptions about the LDAP Data Source,” on page 18](#)

2.1.1 Prerequisites

The LDAP driver requires the following:

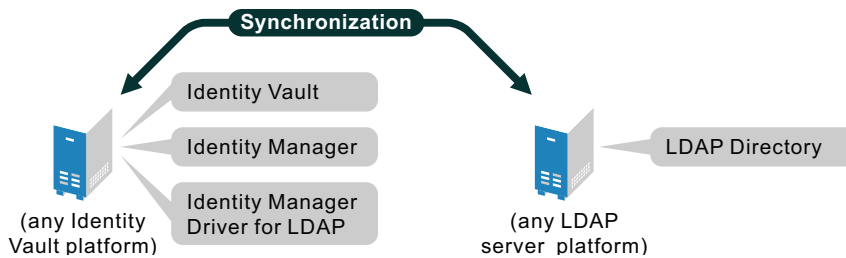
- Novell® Identity Manager is already installed.
- The system requirements of Identity Manager have been met.
- If you are using the changelog method, one of the following LDAP directories must exist:
 - ♦ Critical Path InJoin Directory 3.1
 - ♦ IBM SecureWay Directory 3.2, 4.1.1, or 5.1
 - ♦ IBM Tivoli Directory 6.0 or later
 - ♦ iPlanet Directory Server 5.0 or greater
 - ♦ Netscape Directory Server 4.x or later
 - ♦ Oracle Internet Directory 2.1.1 (Oracle 9:) or later
 - ♦ Sun Java System Directory 5.2 or later
 - ♦ Sun ONE* 5.2

2.1.2 Where to Install the LDAP Driver

You can install the LDAP driver locally or remotely.

An installation on the same computer where an Identity Vault and the Metadirectory engine are installed is referred to as a local configuration. The following figure illustrates a local configuration:

Figure 2-1 A Local Configuration



If platform or policy constraints make a local configuration difficult, you can install the LDAP driver on the computer hosting the target application. This installation is referred to as a remote configuration.

Although a remote configuration is possible, it provides little additional flexibility because of the following:

- ◆ The driver can run on any Identity Vault platform.
- ◆ The driver communicates with the LDAP server on any platform across the wire via the LDAP protocol.

2.1.3 Information to Gather

During installation and setup, you are prompted for information such as the following:

- ◆ Whether to use the Flat or Mirror option for synchronizing hierarchical structure. See [“Policies” on page 13](#).
- ◆ The Identity Vault and LDAP directory containers that you want to hold synchronized objects.
- ◆ The Identity Vault User object to assign as a security equivalent for the driver and the objects to exclude from synchronization.
- ◆ The LDAP object and password used to provide driver access to the LDAP directory.

For information on settings, see [Table 4-1 on page 27](#).

2.1.4 Assumptions about the LDAP Data Source

If you are using the Publisher channel to send data to an Identity Vault about changes in the LDAP directory, you must understand the two methods that the driver uses to publish data:

- ◆ The changelog method

The change log is a mechanism in an LDAP directory. The change log can provide LDAP event information for the driver. This method is preferred when a change log is available.

See [Section 5.3.3, “LDAP Publisher Settings: Changelog and LDAP-Search Methods,” on page 36](#) and [Section 5.3.4, “LDAP Publisher Settings: Only the Changelog Method,” on page 37](#).

- ◆ The LDAP-search method

This method enables the LDAP driver to publish to an Identity Vault data about the LDAP servers that don't use change logs. See [Section 5.3.5, "LDAP Publisher Settings: Only the LDAP-Search Method,"](#) on page 39.

2.2 Upgrading to Identity Manager 3.5.1

During an Identity Manager 3.5.1 installation, you can install the Driver for LDAP (along with other Identity Manager drivers) at the same time that the Metadirectory engine is installed. See the [Identity Manager 3.5.1 Installation Guide](#). You can upgrade from DirXML[®] 1.1a, Identity Manager 2, or Identity Manager 3 to Identity Manager 3.5.1

2.3 Installing and Configuring Password Synchronization

Password synchronization on the Publisher channel works only with the Sun Java System Directory. You can have password synchronization on the Publisher channel by installing a plug-in and configuring the driver to accept password changes from the plug-in. For example, on Linux* the directory is `linux/setup/utilities/sun_password_plugins` directory, and on Windows* it is `nt\dirxml\utilities\sun_password_plugins`. These directories reside on the Identity Manager DVD or download image.

These directories contain the Novell[®] Identity Manager Password plug-in for Sun Java System Directory for several platforms. The plug-in can be used to synchronize user passwords from Sun Java System Directory to the Novell Identity Manager Identity Vault via the Identity Manager distribution password.

IMPORTANT: Only passwords that are set or modified after the plug-in is installed can be synchronized.

The plug-in is a post-operation plug-in. Sun Java System Directory notifies the plug-in whenever a password is set or changed. The plug-in then encrypts the password by using the Advanced Encryption Standard (AES) and stores the encrypted password on the `novellDistPassword` attribute. The LDAP driver can then synchronize the encrypted password to Novell Identity Manager. The LDAP driver decrypts the password and uses it to set the Identity Manager distribution password.

Several versions of the Novell Identity Manager Password plug-in exist. You can install these versions in your Sun Java System directory on Windows, Linux, Solaris* SPARC*, or AIX*.

NOTE: The plug-in has been tested only on certain versions of the following platforms:

- ◆ Windows XP Professional
 - ◆ Red Hat* 7.2
 - ◆ Solaris 8 64-bit
 - ◆ AIX 5.1
-

1 Locate the correct plug-in binary file.

The plug-in binary files are stored in directories representing the supported platforms.

For example, if your Sun directory runs on AIX, look in the AIX directory. The plug-in filename is `novl-idm-pswd.so` on all platforms except Windows, where the filename is `novl-idm-pswd.dll`.

- 2 Copy the binary plug-in file to the `lib` directory in your Sun Java System Directory installation location.

For example, on Windows the default installation location for Sun Java System Directory is `C:\Program Files\Sun\MPS` and inside that directory is a `lib` directory. Put `novl-idm-pswd.dll` in the `lib` directory.

On other platforms, the default installation location is often `/var/Sun/mps`. You need to locate the Sun Java System Directory installation location on your system, and put the plug-in file inside the `lib` directory.

NOTE: On Solaris SPARC computers, the Sun Java System Directory installation includes two versions of most libraries: a 32-bit version and a 64-bit version. By default, the 32-bit version is found at `/var/Sun/mps/lib`. The 64-bit version is found at `/var/Sun/mps/lib/64`.

Both a 32-bit and a 64-bit version of the plug-in are provided. Copy both versions to their respective locations on your Solaris installation. At runtime, the Sun Java System Directory determines which version is the appropriate version to load.

- 3 Locate and edit the `novl-idm-pswd.ldif` file.

The `.ldif` file contains plug-in configuration information that you apply to the directory. It also contains two schema definitions:

- ♦ One definition is for the `novellDistPassword` attribute that stores the encrypted password.
- ♦ The other definition is for the `novellDistPasswordUser` auxiliary class that is applied to your users to allow the use of the `novellDistPassword` attribute.

As a convenience, the `.ldif` file also contains an instruction to turn on the Retro Changelog Plugin, which most customers want turned on to enable Publisher channel operations with the Identity Manager LDAP driver. If you know that the changelog is already enabled, or if you don't want to enable the changelog, you can remove the Retro Changelog Plugin section from the `.ldif` file.

Most users need to edit only two items in the `.ldif` file:

- ♦ The `nsslapd-pluginPath` attribute
- ♦ The `nsslapd-pluginarg0` attribute

Ensure that the value of `nsslapd-pluginPath` is the path where you installed the plug-in. For example, if you installed the plug-in in the `/var/Sun/mps/lib` directory, the value should be `/var/Sun/mps/lib/novl-idm-pswd.so`. Set the value of `nsslapd-pluginarg0` to a password that will be used to generate an AES key used to encrypt user passwords. This must match the password used in [Step 6 on page 21](#).

NOTE: Solaris users should set the value of `nsslapd-pluginPath` to the path of the 32-bit version of the plug-in, even if the operating system is 64-bit. (See [Step 2 on page 20](#).) At runtime, the directory determines whether to load the 32-bit or the 64-bit version of the plug-in.

- 4 Apply the `novl-idm-pswd.ldif` file to the Sun directory.

To complete this step, you need to know the configuration administrator's DN and password. Typically, the DN will be `"uid=admin,ou=Administrators,ou=TopologyManagement,o=NetscapeRoot"`. However, the password will vary. You also need to know the LDAP port used by your Sun directory.

The `ldapmodify` command line utility that was installed with your Sun Java System Directories can be used to apply the `.ldif` file. Use a command similar to the following:

```
ldapmodify -h localhost -p 389 -D
"uid=admin,ou=Administrators,ou=TopologyManagement,o=NetscapeRoot" -w password -f novl-idm-pswd.ldif
```

- 5 Restart Sun Java System Directory so that your changes take affect and the plug-in starts.

For troubleshooting, note any errors that might appear on the console.

- 6 Install and configure the LDAP driver for Identity Manager.

If you are installing from Identity Manager version 3.0.5 or later, and are using the supplied example configuration template for the LDAP driver, a section at the end of the Publisher Settings enables you to configure the driver to synchronize passwords found in a Sun directory.

IMPORTANT: Use the same encryption password (used to generate an AES key) for both the LDAP driver configuration and the Novell Identity Manager Password plug-in on Sun Java System Directory.

If you are upgrading an existing driver and can't use the sample configuration template provided with Identity Manager 3.0.5 or later, you can add the appropriate Publisher settings yourself. See [Section 2.3.1, "Manually Adding Publisher Settings," on page 21](#).

2.3.1 Manually Adding Publisher Settings

- 1 In iManager, select *Edit XML*.
- 2 Copy the following XML and paste it at the end of your Publisher settings, but still within the definitions element.

```
<group>
  <definition display-name="Use Sun Password Plugin"
    name="useSunPluginGroup" type="enum">
    <description>Specify Yes if you have installed and
      configured the Novell Identity Manager Password Plugin on
      Sun Java System Directory and want to use it to
      synchronize to the Identity Manager distribution
      password.</description>
    <enum-choice display-name="Yes">yes</enum-choice>
    <enum-choice display-name="No">no</enum-choice>
    <value>yes</value>
  </definition>
  <subordinates active-value="yes">
    <definition display-name="Password Publishing Encryption
      Password" name="pub-password-encryption-key"
      type="string">
    <description>Enter the same password configured on the
      Novell Identity Manager Password Plugin on the Sun
      Java System Directory. This password will be used to
      generate a key that will decrypt the passwords.
    </description>
    <value>enter encrypt password</value>
    </definition>
  </subordinates>
</group>
```

2.4 Installing the Driver Separately

Install the Identity Manager Driver for LDAP on a Windows NT* 2003 server, or Windows NT 2000 with Support Pack 2.

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 LDAP driver. See “[Installing Identity Manager](#)” in the *Identity Manager 3.5.1 Installation Guide*.

Typically, an Identity Manager installation installs all drivers, including the LDAP driver, at the same time that the Metadirectory engine is installed. If the LDAP 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.

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 LDAP Driver

3

If you are upgrading from Identity Manager 3.5.0 to Identity Manager 3.5.1, skip this section. The information does not apply.

If you have been using a version earlier than Identity Manager 3.5, follow the instructions in this section instead of the instructions in [Chapter 2, “Installing the LDAP Driver,” on page 17](#).

Identity Manager 3.5 contained a new architecture for how policies reference one another. To take advantage of this architecture, the driver configuration file provided for LDAP must be upgraded. For more information on this 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 23](#)
- [Section 3.2, “Upgrading the Driver by Using iManager,” on page 26](#)

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.

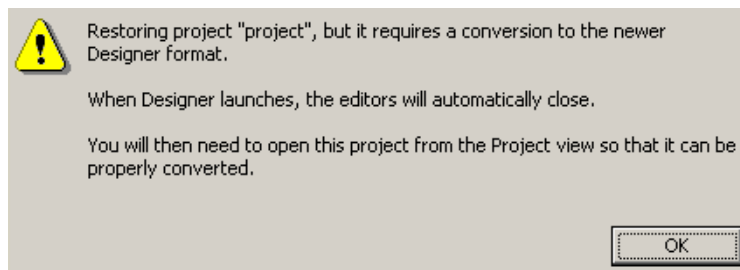
- 2 Back up the driver.

See [Chapter 10, “Backing Up the LDAP Driver,” on page 83](#) for instructions on how to back up the driver.

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

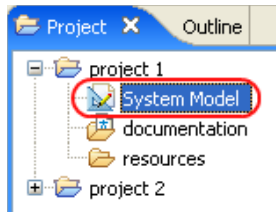
If you had a project open in Designer when you upgraded Designer, proceed to [Step 4](#). If you didn't have a project open in Designer when you upgraded Designer, skip to [Step 5](#).

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

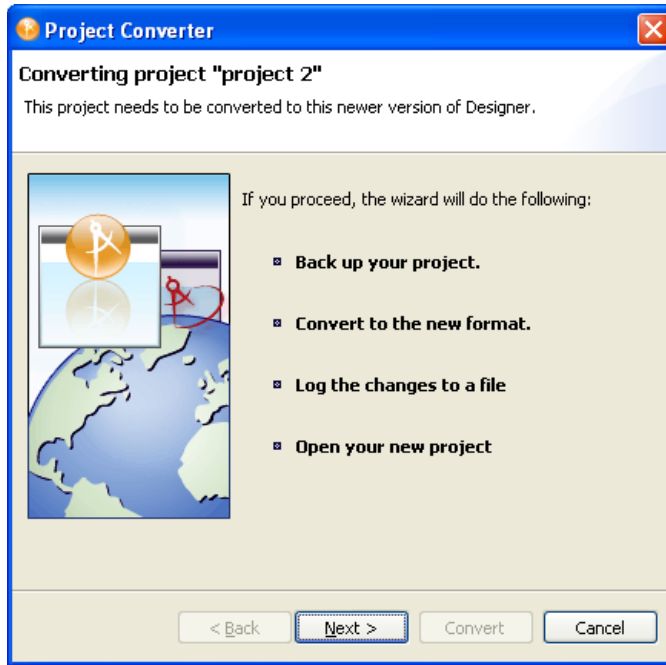


Designer closes the project to perform the upgrade.

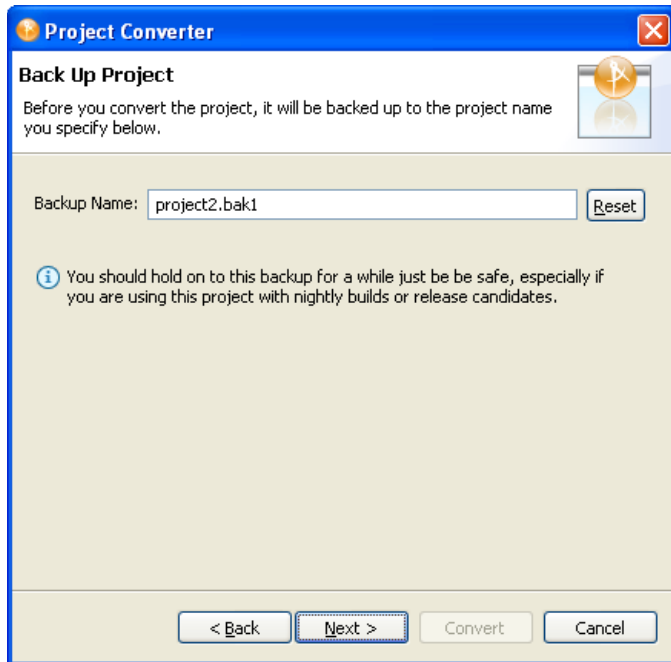
- 5 To open and convert the project, double-click *System Model* in the Project view.



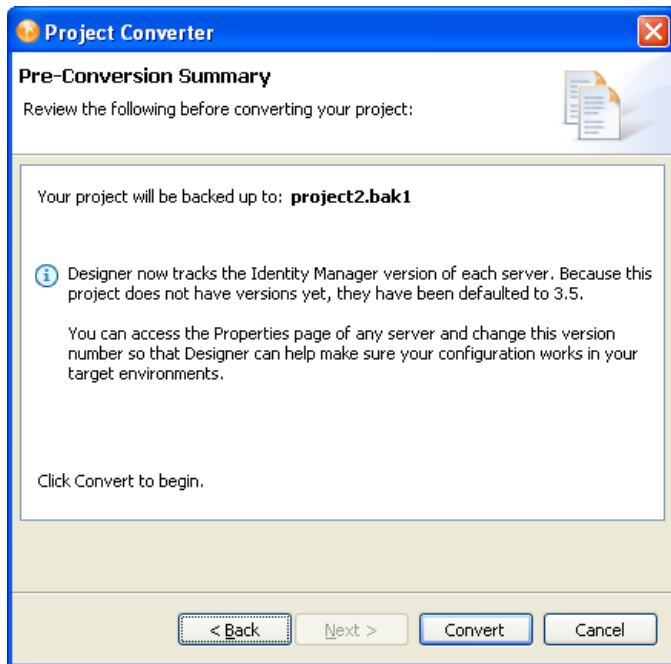
6 Read the tasks listed in the Project Converter message, then click *Next*.



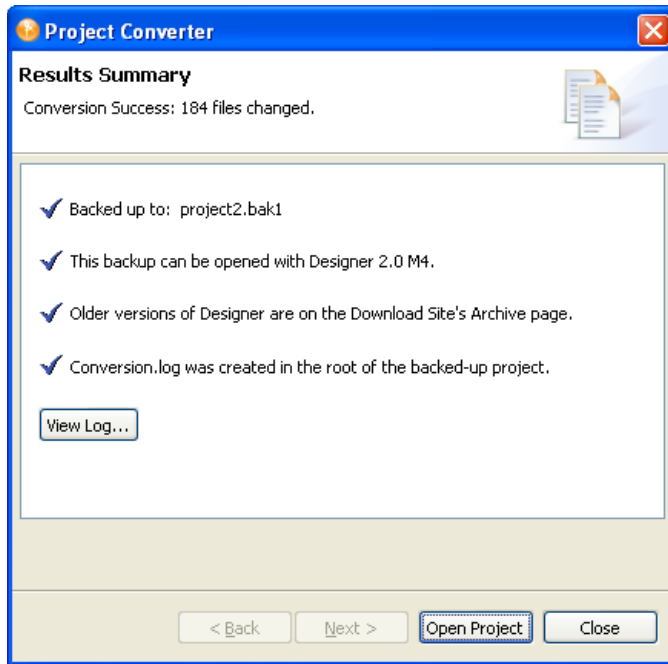
7 Specify the name of the backup project name, then click *Next*.



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

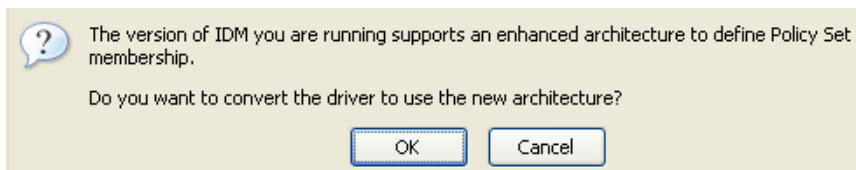
- 1 Make sure that you have updated your driver with all the patches for the version you are currently running.

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 LDAP Driver,” on page 83](#).

- 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*.



Importing an Example LDAP 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 27](#)
- ♦ [Section 4.2, “Using iManager to Import,” on page 29](#)

4.1 Using Designer to Import

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

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 *LDAP*, then click *Run*.
- 4 Click *Yes* in the Perform Prompt Validation window.
- 5 Configure the driver by filling in the fields.
Specify information specific to your environment. See [Table 4-1 on page 27](#).
- 6 After specifying parameters, click *OK* to import the driver.
- 7 Customize and test the driver.
- 8 Deploy the driver into the Identity Vault.
See “[Deploying and Exporting](#)” in the *Designer 2.1 for Identity Manager 3.5.1* guide.

Table 4-1 Settings for the LDAP Driver

Field	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>Placement Type</i>	<p>With the Simple placement option, new User objects created in the LDAP directory are placed in the container in an Identity Vault that you specify when importing the driver configuration. The user object is named with the value of cn.</p> <p>With the Mirror placement option, new User objects created in the LDAP directory are placed in the Identity Vault container that mirrors the object's LDAP container.</p>

Field	Description
<i>eDirectory Container</i>	<p>The container in an Identity Vault where new users should be created.</p> <p>If this container doesn't exist, you must create it before you start the driver.</p> <p>For the LDAPMirrorSample.xml configuration, this directory is the starting point for the driver's Placement policy. Subordinate containers should be named the same as the subordinate containers in the LDAP mirror container.</p> <p>For the Flat configuration, this container houses all User objects.</p>
<i>LDAP Container</i>	<p>The container in the LDAP directory where new users should be created.</p> <p>If this container doesn't exist, you must create it before you start the driver.</p> <p>For the Flat configuration, this directory is the starting point for the driver's Placement policy.</p> <p>For the LDAPSimplePlacementSample.xml configuration, this container houses all User objects.</p>
<i>LDAP Server</i>	The hostname or IP address and port of the LDAP server.
<i>LDAP Authentication DN</i>	Specify the LDAP DN of the administrator account created for the LDAP driver.
<i>LDAP Authentication Password</i>	<p>The password for the LDAP driver administrator account. You confirm the password by re-entering it in the next field.</p> <p>This is the required password for the authenticated user.</p> <p>If the LDAP driver uses Directory Manager exclusively, the default authenticated user works well. However, if this user is used for any other purpose, you should probably change the default after you get the driver running. See "Creating an LDAP User Object with Authentication Rights" on page 31.</p>
<i>SSL</i>	Encrypts LDAP protocol communications.
<i>Configure Data Flow</i>	<ul style="list-style-type: none"> ◆ Bidirectional: Both LDAP and the Identity Vault are authoritative sources of the data synchronized between them. ◆ LDAP to eDirectory: LDAP is the authoritative source. ◆ eDirectory to LDAP: The Identity Vault is the authoritative source.
<i>Install Driver as Remote/Local</i>	Configure the driver for use with the Remote Loader service by selecting <i>Remote</i> , or select <i>Local</i> to configure the driver for local use.
<i>Remote Host Name and Port</i>	Specify the host name or IP address and port number where the Remote Loader Service has been installed and is running for this driver. The default port is 8090.
<i>Driver Password</i>	The Remote Loader uses the Driver object password to authenticate itself to the Metadirectory server. The Driver object password must be the same password that is specified as the Driver object password on the Identity Manager Remote Loader.
<i>Remote Password</i>	<p>This password is used only in the Remote Loader configuration. It allows the Remote Loader to authenticate to the Metadirectory engine.</p> <p>The Remote Loader password is used to control access to the Remote Loader instance. The Remote Loader password must be the same password that is specified as the Remote Loader password on the Identity Manager Remote Loader.</p>

Field	Description
<i>Password Failure Notification User</i>	Sends an e-mail notification to a specified user when a password fails.
<i>Enable Entitlements</i>	Choose Yes or No. Because this is a design decision, you should understand entitlements before choosing to use it. For information about entitlements, see “ Creating and Using Entitlements ” in the <i>Novell Identity Manager 3.5.1 Administration Guide</i> .

4.2 Using iManager to Import



Identity Manager provides an example configuration file. 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 Drivers*.
- 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 *LDAP*, then click *Next*.
- 4 Configure the driver by filling in the configuration parameters.
For information on the settings, see [Table 4-1 on page 27](#).
- 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 2 on page 29](#). 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 LDAP Driver

5

The LDAP driver includes an example configuration file that you can use as a starting point for your deployment. However, most Identity Manager deployments require you to modify the example.

- ◆ [Section 5.3, “Controlling Data Flow from the LDAP Directory to an Identity Vault,” on page 34](#)
- ◆ [Section 5.4, “Synchronizing Data,” on page 41](#)
- ◆ [Section 5.5, “Configuring SSL Connections,” on page 44](#)

NOTE: When you customize data synchronization, you must work within the supported standards and conventions for the operating systems and accounts being synchronized. Data containing characters that are valid in one environment, but invalid in another, causes errors.

5.1 Preparing the LDAP Server

If you use the driver only to synchronize data from an Identity Vault to the LDAP server (on a Subscriber channel), most LDAP servers and applications work without any additional configuration.

You always create a User object that has the necessary rights so that the driver can authenticate to the LDAP server.

However, if the changes made to entries on the LDAP server must synchronize back to an Identity Vault (on a Publisher channel), and if you plan to use the changelog method, you need to perform at least one other configuration task on the LDAP server before running the driver. Verify that the change log mechanism of the LDAP server is enabled. For information on the changelog method, see [Section 1.2.2, “Two Publication Methods,” on page 11](#).

IMPORTANT: If the LDAP server doesn't have a changelog mechanism, use the LDAP-search method. Otherwise, the driver won't be able to publish events for that server.

- ◆ [Section 5.1.1, “Creating an LDAP User Object with Authentication Rights,” on page 31](#)
- ◆ [Section 5.1.2, “Enabling the Change Log,” on page 32](#)

5.1.1 Creating an LDAP User Object with Authentication Rights

When you use the changelog publication method, the driver attempts to prevent loopback situations where an event that occurs on the Subscriber channel is sent back to the Metadirectory engine on the Publisher channel. However, the LDAP-search method relies on the Metadirectory engine to prevent loopback.

With the changelog method, one way that the driver prevents loopback from happening is to look in the change log to see which user made the change. If the user that made the change is the same user that the driver uses to authenticate with, the Publisher assumes that the change was made by the driver's Subscriber channel.

NOTE: If you use Critical Path InJoin Server, the change log implementation on that server is somewhat limited because it doesn't provide the DN of the object that initiated the change. Therefore, the creator/modifier DN can't be used to determine whether the change came from an Identity Vault or not.

In that case, all changes found in the change log are sent by the Publisher to the Metadirectory engine, and the Optimize/Modify discards unnecessary or repetitive changes.

To stop the Publisher channel from discarding legitimate changes, make sure the User object that the driver uses to authenticate with is not used for any other purpose.

For example, suppose you are using the Netscape Directory Server and have configured the driver to use the administrator account CN=Directory Manager. If you want to manually make a change in the Netscape Directory Server and have that change synchronize, you can't log in and make the change with CN=Directory Manager. You must use another account.

To avoid this problem:

- 1 Create a user account that the driver uses exclusively.
- 2 Assign that user account rights to see the change log and to make any changes that you want the driver to be able to make

For example, at the VMP company, you create a user account for the driver called uid=ldriver,ou=Directory Administrators,o=lansing.vmp.com. You then assign the appropriate rights to the user account by applying the following LDIF to the server by using the LDAPModify tool or the Novell[®] Import Conversion Export utility.

```
# give the new user rights to read and search the changelog
dn: cn=changelog
changetype: modify
add: aci
aci: (targetattr = "*") (version 3.0; acl "LDAP DirXML Driver";
allow (compare,read,search) userdn = "ldap:///
uid=ldriver,ou=Directory Administrators,o=lansing.vmp.com"; )
-

# give the new user rights to change anything in the
o=lansing.vmp.com container
dn: o=lansing.vmp.com
changetype: modify
add: aci
aci: (targetattr = "*") (version 3.0; acl "LDAP DirXML Driver";
allow (all) userdn = "ldap:///uid=ldriver,ou=Directory
Administrators,o=lansing.vmp.com"; )
-
```

5.1.2 Enabling the Change Log

The change log is the part of the LDAP server that enables the driver to recognize changes that require publication from the LDAP directory to an Identity Vault. The LDAP directories supported by this driver support the changelog mechanism.

Critical Path InJoin and Oracle Internet Directory have the change log enabled by default. Unless the change log has been turned off, you don't need to perform any additional steps to enable it.

IBM SecureWay, Netscape Directory Server, and iPlanet Directory Server require you to enable the change log after installation. For information on enabling the change log, refer to the documentation supporting your LDAP directory.

TIP: The iPlanet change log requires you to enable the Retro Changelog Plug-in.

5.2 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 LDAP 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.

NOTE: When migrating data from an Identity Vault into the LDAP directory, you might need to change your LDAP server settings to allow migration of large numbers of objects. See [Section 9.2, “Migrating Users into an Identity Vault,” on page 81](#).

- ♦ **Migrate Data into the Identity Vault:** Allows you to define the criteria that Identity Manager uses to migrate objects from an LDAP 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, select *Identity Manager > Identity Manager Overview*.
- 2 Locate the driver set that contains the Identity Manager Driver for LDAP, then double-click the driver icon.
- 3 Click the appropriate migration button.

5.3 Controlling Data Flow from the LDAP Directory to an Identity Vault

Figure 5-1 Settings in the Example Configuration File

Driver Parameters

VITAL2.vel

[Edit XML](#)

Driver Settings

LDAP Directory Type ⓘ	LDAPv3 ▾
Enforce Matching Parenthesis in Schema Elements ⓘ	No ▾
Additional Allowable Schema Name Characters ⓘ	<input type="text"/>
Use SSL ⓘ	No ▾

Subscriber Settings

LDAP Server Supports Binary Attribute Option ⓘ	Yes ▾
--	-------

Publisher Settings

Polling Interval in Seconds ⓘ	<input type="text" value="20"/>
Temporary File Directory ⓘ	<input type="text"/>
Heartbeat interval in minutes ⓘ	<input type="text"/>
Publication Method ⓘ	Changelog ▾
Changelog Entries to Process on Startup ⓘ	Previously unprocessed ▾
Maximum Batch Size for Changelog Processing ⓘ	<input type="text" value="1000"/>
Preferred LDAP ObjectClass Names ⓘ	<input type="text"/>
Prevent Loopback ⓘ	Yes ▾
Use Sun Password Plugin ⓘ	No ▾

Adjusting the driver’s operating parameters allows you to tune driver behavior to align with your network environment. For example, you might find the default Publisher channel polling interval to be shorter than your synchronization needs require. Making the interval longer could improve network performance while still maintaining appropriate synchronization.

If the LDAP server has a change log, we recommend that you use the changelog publication method. If a change log is unavailable, you can use the LDAP-search publication method. The changelog method is the preferred method. See [Section 1.2.2, “Two Publication Methods,” on page 11](#).

- ◆ [Section 5.3.1, “LDAP Driver Settings,” on page 35](#)
- ◆ [Section 5.3.2, “LDAP Subscriber Settings,” on page 35](#)
- ◆ [Section 5.3.3, “LDAP Publisher Settings: Changelog and LDAP-Search Methods,” on page 36](#)
- ◆ [Section 5.3.4, “LDAP Publisher Settings: Only the Changelog Method,” on page 37](#)
- ◆ [Section 5.3.5, “LDAP Publisher Settings: Only the LDAP-Search Method,” on page 39](#)

5.3.1 LDAP Driver Settings

Figure 5-2 LDAP Driver Settings

LDAP Directory Type ⓘ	LDAPv3 ▾
Enforce Matching Parenthesis in Schema Elements ⓘ	No ▾
Additional Allowable Schema Name Characters ⓘ	-
Use SSL ⓘ	Yes ▾
Keystore Path for SSL Certs ⓘ	c:\mykeystore
Use Mutual Authentication ⓘ	No ▾

- 1 In iManager, select *Identity Manager > Identity Manager Overview*, then search for the driver set.
- 2 In the driver set, click the LDAP driver icon.
- 3 In the driver view, click the LDAP driver icon again.
- 4 Scroll to *Driver Parameters*.
- 5 In the *Driver Settings* section, select the desired option.
For information on a setting, click the Information icon ⓘ.

5.3.2 LDAP Subscriber Settings

Figure 5-3 The LDAP Subscriber Setting

LDAP Server Supports Binary Attribute Option ⓘ	Yes ▾
--	-------

You aren't prompted for this setting when you import the sample configuration file. However, you can change the setting after importing the file. In the *Subscriber Settings* section, select the desired option.

The default setting is *Yes*. Most LDAP servers support the use of the binary attribute option as defined in RFC 2251 section 4.1.5.1.

If you don't know whether the LDAP server that this driver connects to supports the binary attribute option, select *Yes*.

5.3.3 LDAP Publisher Settings: Changelog and LDAP-Search Methods

Figure 5-4 LDAP Common Publisher Settings

Publisher Settings	
Polling Interval in Seconds ⓘ	<input type="text" value="20"/>
Temporary File Directory ⓘ	<input type="text"/>
Heartbeat interval in minutes ⓘ	<input type="text"/>

Some settings apply to both the changelog and LDAP-search publication methods. Some settings apply only to the changelog publication method. Other settings apply only to the LDAP-search publication method.

- ◆ “Polling Interval in Seconds” on page 36
- ◆ “Temporary File Directory” on page 36
- ◆ “Heartbeat Interval in Minutes” on page 36

Polling Interval in Seconds

The interval at which the driver checks the LDAP server’s changelog or LDAP-search method. When new changes are found, they are applied to the Identity Vault.

The recommended polling interval is 120 seconds.

Temporary File Directory

Set the value to a directory on the local file system (the one where the driver is running) where temporary state files can be written. If you don’t specify a path, the driver uses the default driver path.

Table 5-1 Temporary File Directories

Platform or Environment	Default Directory
eDirectory™	The DIB file directory
Remote Loader	The root Remote Loader directory

These files help do the following:

- ◆ Maintain driver consistency even when the driver is shut down
- ◆ Prevent memory shortages when the data being searched is extensive

Heartbeat Interval in Minutes

To turn on a heartbeat, type a value. To turn off the heartbeat, leave this field empty.

For information on the driver heartbeat, see “[Adding Driver Heartbeat](#)” in the *Novell Identity Manager 3.5.1 Administration Guide*.

5.3.4 LDAP Publisher Settings: Only the Changelog Method

Figure 5-5 Changelog Settings on the LDAP Publisher Channel

Publisher Settings	
Polling Interval in Seconds ⓘ	<input type="text" value="20"/>
Temporary File Directory ⓘ	<input type="text"/>
Heartbeat interval in minutes ⓘ	<input type="text"/>
Publication Method ⓘ	Changelog ▾
Changelog Entries to Process on Startup ⓘ	Previously unprocessed ▾
Maximum Batch Size for Changelog Processing ⓘ	<input type="text" value="1000"/>
Preferred LDAP ObjectClass Names ⓘ	<input type="text"/>
Prevent Loopback ⓘ	Yes ▾
Use Sun Password Plugin ⓘ	Yes ▾
Sun Plugin Encryption Password ⓘ	<input type="password"/>
Reenter Sun Plugin Encryption Password	<input type="password"/>
<input type="checkbox"/> Remove existing password	

- ◆ “[Changelog Entries to Process on Startup](#)” on page 37
- ◆ “[Maximum Batch Size for Changelog Processing](#)” on page 38
- ◆ “[Preferred LDAP ObjectClass Names](#)” on page 38
- ◆ “[Prevent Loopback](#)” on page 38

Changelog Entries to Process on Startup

This parameter specifies which entries to process on startup.

- ◆ **All:** The Publisher attempts to process all of the changes found in the change log. The Publisher continues until all changes have been processed. It processes new changes according to the poll rate.
- ◆ **None:** When the driver starts running, the Publisher doesn’t process any previously existing entries. It processes new changes according to the poll rate.
- ◆ **Previously Unprocessed:** This setting is the default. If this is the first time the driver has been run, it behaves like the *All* option, processing all new changes.

If the driver has been run before, this setting causes the Publisher to process only changes that are new since the last time the driver was running. Thereafter, it processes new changes according to the poll rate.

When using the changelog method, the driver looks for a batch size and a Prevent Loopback setting.

Maximum Batch Size for Changelog Processing

When the Publisher channel processes new entries from the LDAP change log, the Publisher asks for the entries in batches of this size. If there are fewer than this number of change log entries, all of them are processed immediately. If there are more than this number, they are processed in consecutive batches of this size.

Preferred LDAP ObjectClass Names

The *Preferred LDAP ObjectClass Name* setting is an optional driver parameter that lets you specify preferred object classes on the Publisher channel.

Identity Manager requires that objects be identified by using a single object class. However, many LDAP servers and applications can list multiple object classes for a single object. By default, when the Identity Manager Driver for LDAP finds an object on the LDAP server or application that has been added, deleted, or modified, it sends the event to the Metadirectory engine and identifies it by using the object class that has the most levels of inheritance in the schema definition.

For example, a user object in LDAP is identified with the object classes of inetorgperson, organizationalperson, person, and top. Inetorgperson has the most levels of inheritance in the schema (inheriting from organizationalperson, which inherits from person, which inherits from top). By default, the driver uses inetorgperson as the object class it reports to the Metadirectory engine.

If you want to change the default behavior of the driver, you can add the optional driver Publisher parameter named preferredObjectClasses. The value of this parameter can be either one LDAP object class or a list of LDAP object classes separated by spaces.

When this parameter is present, the Identity Manager Driver for LDAP examines each object being presented on the Publisher channel to see if it contains one of the object classes in the list. It looks for them in the order they appear in the preferredObjectClasses parameter. If it finds that one of the listed object classes matches one of the values of the objectclass attribute on the LDAP object, it uses that object class as the one it reports to the Metadirectory engine. If none of the object classes match, it resorts to its default behavior for reporting the primary object class.

Prevent Loopback

The Prevent Loopback parameter is used only with the changelog publication method. The LDAP-search method doesn't prevent loopback, other than the loopback prevention built into the Metadirectory engine.

The default behavior for the Publisher channel is to avoid sending changes that the Subscriber channel makes. The Publisher channel detects Subscriber channel changes by looking in the LDAP change log at the creatorsName or modifiersName attribute to see whether the authenticated entry that made the change is the same entry that the driver uses to authenticate to the LDAP server. If the entry is the same, the Publisher channel assumes that this change was made by the driver's Subscriber channel and doesn't synchronize the change.

As an example scenario, you might not have a Subscriber channel configured for this driver but you want to be able to use the same DN and password as other processes use to make changes.

If you are certain that you want to allow this type of loopback to occur, edit the driver parameter:

- 1 In iManager, select *Identity Manager Management > Identity Manager Overview*.
- 2 Find the driver in its driver set.

- 3 Click the driver to open the Driver Overview page, then click the driver again to open the *Modify Object* page.
- 4 Scroll to the *Publisher Settings* section, then set *Prevent Loopback* to *No*.
- 5 Click *OK*, click *Apply*, then restart the driver for this parameter to function.

5.3.5 LDAP Publisher Settings: Only the LDAP-Search Method

Figure 5-6 LDAP-Search Settings on the LDAP Publisher Channel

Publisher Settings	
Polling Interval in Seconds ⓘ	20
Temporary File Directory ⓘ	
Heartbeat interval in minutes ⓘ	
Publication Method ⓘ	LDAP Search ▼
Search Base DN ⓘ	o=mycompany
Search Scope ⓘ	Subtree ▼
Class Processing Order ⓘ	others groupofuniquenames
Search Results to Synchronize on First Startup ⓘ	Synchronize only subsequent changes ▼

Traditionally, the LDAP driver has been able to detect changes in an LDAP server only by reading its change log. However, some servers don't use the changelog mechanism, which is actually not part of the LDAP standard. Where change logs don't exist, the LDAP driver has previously been unable to publish data about these LDAP servers to an Identity Vault.

However, the LDAP-search publication method doesn't require a change log. This method detects changes by using standard LDAP searches and then comparing the results from one search interval to the next interval.

You can use the LDAP-search publication method as an alternative to the traditional changelog publication method. The Identity Manager Driver for LDAP supports either method. However, the changelog method has performance advantages and is the preferred method when a change log is available.

WARNING: The LDAP-Search method works by comparing the current state of the LDAP server with previous states, and sending updates to the Identity Vault that reflect the changes. When an entry with a specific DN exists in a previous state, but not the current state, the driver has no way to know whether that entry was deleted or whether it was renamed or moved. Therefore, it sends a Delete event to the Identity Vault for the previous DN, and if it was renamed or moved, then a new Add event is generated.

This process usually works well if the LDAP server is the authoritative source for all of the entry attributes. However, if other sources (such as other drivers) also provide information for the entry in the Identity Vault, then deleting an entry that has only been moved or renamed would be undesirable because it could result in data loss. In this case, you might need to create policy that would veto Delete events on the publisher channel, or re-evaluate whether moves or renames should be done at all in the LDAP directory.

If no change log is available, set the following parameters:

- ◆ “Search Base DN” on page 40
- ◆ “Search Scope (1-Subtree, 2-One Level, 3-Base)” on page 40
- ◆ “Class Processing Order” on page 40
- ◆ “Search Results to Synchronize on First Startup” on page 40

Search Base DN

A required parameter when you use the Publisher channel if no change log is available. Set the parameter to the LDAP distinguished name (DN) of the container where the polling searches should begin (for example, ou=people,o=company).

To use a change log, leave this parameter blank.

Search Scope (1-Subtree, 2-One Level, 3-Base)

Indicates the depth of the polling searches. This parameter defaults to search the entire subtree that the Search Base DN points to.

Set this parameter when no change log is available.

Class Processing Order

An optional parameter that the Publisher channel uses to order certain events when referential attributes are an issue. The value of the parameter is a list of class names from the LDAP server, separated by spaces. For example, to make sure that new users are created before they are added to groups, make sure that `interorgperson` comes before `groupofuniquenames`.

The Identity Manager Driver for LDAP defines a special class name, “others,” to mean all classes other than those explicitly listed.

The default value for this parameter is “other groupofuniquenames.”

Use this parameter when no change log is available.

Search Results to Synchronize on First Startup

The first time that the LDAP driver starts, the driver performs the defined LDAP search. The *Search Results to Synchronize on First Startup* setting defines whether the initial search results are synchronized, or only subsequent changes are synchronized.

The *Search Results to Synchronize on First Startup* option appears only if the *Publication Method* parameter is set to *LDAP-Search*. You aren’t prompted for this setting when you import the configuration file. However, you can change the setting after importing the file.

- 1 In iManager, select *Identity Manager > Identity Manager Overview*, then search for the driver set.
- 2 In the driver set, click the LDAP driver icon.
- 3 In the driver view, click the LDAP driver icon again.
- 4 Scroll to *Driver Parameters*.
- 5 In the *Publisher Settings* section, select the desired option.

The default setting is *Synchronize only subsequent changes*.

6 Click *OK*.

5.4 Synchronizing Data

- ♦ [Section 5.4.1, “Determining Which Objects Are Synchronized,” on page 41](#)
- ♦ [Section 5.4.2, “Defining Schema Mapping,” on page 41](#)
- ♦ [Section 5.4.3, “Defining Object Placement in Netscape Directory Server,” on page 42](#)
- ♦ [Section 5.4.4, “Working with eDirectory Groups and Netscape,” on page 44](#)

5.4.1 Determining Which Objects Are Synchronized

Identity Manager uses filters on the Publisher and Subscriber channels to control which objects are synchronized and to define the authoritative data source for these objects.

The default filters are illustrated in [“Filters” on page 12](#). Use the following procedure to make changes to the default.

- 1 In iManager, select *Identity Manager > Identity Manager Overview*.
- 2 Locate the driver in its driver set.
- 3 Click the driver to open the *Identity Manager Driver Overview* page.
- 4 Click the Publisher or Subscriber filter icon and make the appropriate changes.

The Publisher filter must include the Identity Vault mandatory attributes. The Subscriber filter must include the LDAP server required attributes.

For every object and attribute selected in the filter, the Mapping policy must have a corresponding entry unless the class or attribute names are the same in both directories. Before mapping an attribute, verify that a corresponding attribute actually exists in the target directory.

- 5 Click *OK*.

5.4.2 Defining Schema Mapping

Different LDAP servers have different schemas. When the driver is first started, it queries the server for the specific schema.

You must be familiar with the characteristics of directory attributes and the LDAP server attributes. The driver handles all LDAP attribute types (cis, ces, tel, dn, int, bin). It also handles the eDirectory Facsimile Telephone Number.

When mapping attributes, follow these guidelines:

- ♦ Verify that every class and attribute specified in the Subscriber and Publisher policies is mapped in the Mapping policy unless the class or attribute names are the same in both directories.
- ♦ Before mapping a directory attribute to an LDAP server attribute, verify that an LDAP server attribute actually exists. For example, the Full Name attribute is defined for a User object on an Identity Vault but fullname doesn't exist in an inetOrgPerson object on Netscape.

- ◆ Always map attributes to attributes of the same type. For example, map strings attributes to strings attributes, octet attributes to binary attributes, or telnumber attributes to telnumber attributes.
- ◆ Map multivalued attributes to multivalued attributes.

The driver doesn't provide data conversion between different attribute types or conversions from multivalued to single-valued attributes. The driver also doesn't understand structured attributes except for Facsimile Telephone Number and Postal Address.

Identity Manager is flexible about the syntax that it accepts coming in from the Publisher:

- ◆ **Accepting Non-Structured/Non-Octet Syntax:** Identity Manager accepts any non-structured/non-octet syntax for any other non-structured/non-octet syntax as long as the actual data can be coerced to the appropriate type. That is, if the Identity Vault is looking for a numeric value, the actual data should be a number.
- ◆ **Coercing the Data to Octet:** When Identity Manager is expecting octet data and gets another non-octet/non-structured type, Identity Manager coerces the data to octet by serializing the string value to UTF-8.
- ◆ **Coercing the Data to a String:** When Identity Manager is passed octet data and another non-structured type is expected, Identity Manager coerces the data to a string by decoding the Base64 data. Identity Manager next tries to interpret the result as a UTF-8 encoded string (or the platform's default character encoding if it is not a valid UTF-8 string) and then applies the same rules as Accepting Non-Structured/Non-Octet Syntax.
- ◆ **FaxNumber:** For faxNumber, if a non-structured type is passed in, Accepting Non-Structured/Non-Octet Syntax and Coercing the Data to a String are applied to the data to get the phone number portion of the fax number. The other fields are defaulted.
- ◆ **State:** State. For state, False, No, F, N (in either upper or lowercase), 0 and "" (empty string) are interpreted as False, and any other value is interpreted as True.

To configure the Schema Mapping policy:

- 1 In iManager, click *Identity Manager > Identity Manager Overview*.
- 2 Locate the driver in its driver set.
- 3 Click the driver to open the Identity Manager Driver Overview page.
- 4 Click the schema mapping icon on the Publisher or Subscriber channel.
- 5 Edit the policy as appropriate for your setup.

5.4.3 Defining Object Placement in Netscape Directory Server

We recommend following the Netscape naming rules for objects in Netscape Directory Server. A brief explanation of naming rules is included here for your convenience.

The directory contains entries that represent people. These person entries must have names. In other words, you must decide what the relative distinguished name (RDN) is for each person entry. The DN must be a unique, easily recognizable, permanent value. We recommend that you use the uid attribute to specify a unique value associated with the person. An example DN for a person entry is:

```
uid=jsmith,o=novell
```

The directory also contains entries that represent many things other than people (for example, groups, devices, servers, network information, or other data). We recommend that you use the cn attribute in the RDN. Therefore, if you are naming a group entry, name it as follows:

```
cn=administrators,ou=groups,o=novell
```

The directory also contains branch points or containers. You need to decide what attributes to use to identify the branch points. Because attribute names have a meaning, use the attribute name with the type of entry it is representing. The Netscape recommended attributes are defined as follows:

Table 5-2 Netscape Recommended Attributes

Attribute Name	Definition
c	Country name
o	Organization name
ou	Organizational Unit
st	State
l	Locality
dc	Domain Component

A Subscriber Placement policy specifies the naming attribute for a classname. The following example is for the User classname. The <placement> statement specifies that uid is used as the naming attribute.

```
<placement-rule>
  <match-class class-name="User"/>
  <match-path prefix="\Novell-Tree\Novell\Users"/>
  <placement>uid=<copy-name/>,ou=People,o=Netscape</
placement>
</placement-rule>
```

The following Subscriber Placement specifies that ou is used as the naming attribute for class-name Organizational Unit.

```
<placement-rule>
  <match-class class-name="Organizational Unit"/>
  <match-path prefix="\Novell-Tree\Novell\Users"/>
  <placement>ou=<copy-name/>,ou=People,o=Netscape</placement>
</placement-rule>
```

To configure a placement policy:

- 1 In iManager, click *Identity Manager > Identity Manager Overview*.
- 2 Locate the driver in its driver set.
- 3 Open the Identity Manager Driver Overview page by clicking the driver.
- 4 Click the Publisher or Subscriber Placement policy icon, then make the appropriate changes.
- 5 Click *Close*.

5.4.4 Working with eDirectory Groups and Netscape

Because group attributes are different in an Identity Vault and Netscape Directory Server, some special processing is required by the driver. On the Publisher channel, special processing takes place when the driver sees the attribute *uniquemember* in the classname *groupofuniquenames*.

The driver also sets the attribute Equivalent To Me in the eDirectory Group. The attribute Equivalent To Me must be included in the Publisher filter. The attribute Equivalent To Me need not be in the Schema Mapping policy because the eDirectory attribute name is used. There is no equivalent attribute name in Netscape Directory Server. No special processing is required on the Subscriber channel.

5.5 Configuring SSL Connections

The driver uses the LDAP protocol to communicate with the LDAP server. Most LDAP servers allow non-encrypted (clear-text) connections. Additionally, when configured correctly, some LDAP servers allow SSL-encrypted connections. SSL connections encrypt all traffic on the TCP/IP socket by using a public/private key pair. The actual LDAP protocol doesn't change, but the communication channel performs the encryption.

The procedure for enabling SSL connections differs slightly from one LDAP server to another. This document covers the process for enabling SSL connections when using Netscape Directory Server 4.12.

- ♦ [“Step 1: Generating a Server Certificate” on page 44](#)
- ♦ [“Step 2: Sending the Certificate Request” on page 45](#)
- ♦ [“Step 3: Installing the Certificate” on page 45](#)
- ♦ [“Step 4: Activating SSL in Netscape Directory Server 4.12” on page 46](#)
- ♦ [“Step 5: Exporting the Trusted Root from the Directory Tree” on page 46](#)
- ♦ [“Step 6: Importing the Trusted Root Certificate” on page 47](#)
- ♦ [“Step 7: Adjusting Driver Settings” on page 48](#)

If you are using another LDAP server, the procedure is similar.

5.5.1 Step 1: Generating a Server Certificate

You first need to install a server certificate. The LDAP server itself can generate a certificate, but the certificate must then be signed by a CA that is trusted by the server. One way to get the certificate signed is to use the CA that comes with an Identity Vault.

To generate a certificate request:

- 1 In the navigation tree in Netscape Console, select the server that the driver will communicate with.
- 2 Click *Open Server*.
- 3 Click *Tasks > Certificate Setup Wizard*.
- 4 Provide information to request a certificate.

Depending on the certificates or tokens that might already be installed on the host system, you might see some or all of the following fields:

Select a Token (Cryptographic Device): Select *Internal (Software)*.

Is the Server Certificate Already Requested and Ready to Install? Select *No*.

If a trust database doesn't already exist for this host, one is generated for you.

A trust database is a key pair and certificate database installed on the local host. When you use an internal token, the trust database is the database into which you install the key and certificate.

5 Type and confirm the password.

The password must contain at least eight characters, and at least one of them must be numeric. This password helps secure access to the new key database you're creating.

6 Continue providing information as prompted, then click *Next*.

7 After a trust database is created, click *Next*.

8 Type the requested information, then click *Next*.

9 Type the password for the token you selected earlier, then click *Next*.

The Certificate Setup Wizard generates a certificate request for your server. When you see the page, you can send the certificate request to the certification authority.

10 Continue with [Step 2: Sending the Certificate Request](#).

5.5.2 Step 2: Sending the Certificate Request

1 Copy the server certificate request into Notepad or another text editor.

2 Save the file as `csr.txt`.

Your certificate request e-mail should look like the following:

```
-----BEGIN NEW CERTIFICATE REQUEST-----
```

```
.
```

```
.
```

```
.
```

```
-----END NEW CERTIFICATE REQUEST-----
```

3 In iManager, select *Novell Certificate Server > Issue Certificate*.

4 In the *Filename* field, browse to `csr.txt`, then click *Next*.

5 Select *Organizational Certificate Authority*.

6 Specify SSL as the key type, then click *Next*.

7 Specify the certificate parameters, click *Next*, then click *Finish*.

8 Save the certificate in Base64 format as `cert.b64` to a local disk or diskette.

9 Continue with [Step 3: Installing the Certificate](#).

5.5.3 Step 3: Installing the Certificate

1 In the navigation tree in Netscape Console, select the server that the driver will be connecting to.

2 Click *Open*.

- 3 Click *Tasks > Certificate Setup Wizard*.
- 4 Start the wizard and indicate that you are ready to install the certificate.
- 5 When prompted, provide the following information:
Select a Token (Cryptographic Device): Select *Internal (Software)*.
Is the Server Certificate Already Requested and Ready to Install? Select *Yes*.
- 6 Click *Next*.
- 7 In the *Install Certificate For* field, select *This Server*.
- 8 In the *Password* field, type the password you used to set up the trust database, then click *Next*.
- 9 In the *Certificate Is Located in This File* field, type the absolute path to the certificate (for example, A: \CERT.B64).
- 10 After the certificate is generated, click *Add*.
- 11 After the certificate is successfully installed, click *Done*.
- 12 Continue with [Step 4: Activating SSL in Netscape Directory Server 4.12](#).

5.5.4 Step 4: Activating SSL in Netscape Directory Server 4.12

After you install the certificate, complete the following to activate SSL:

- 1 In the navigation tree in Netscape Console, select the server you want to use SSL encryption with.
- 2 Click *Open > Configuration > Encryption*.
- 3 Enter the following information:
Enable SSL: Select this option.
Cipher Family: Select *RSA*.
Token to Use: Select *Internal (Software)*.
Certificate to Use: Select *Server-Cert*.
Client Authentication: Because the driver doesn't support client authentication, select *Allow Client Authentication*.
- 4 Click *Save*.
- 5 Click *Tasks*, then restart the server for the changes to take effect.
- 6 Continue with [Step 5: Exporting the Trusted Root from the Directory Tree](#).

5.5.5 Step 5: Exporting the Trusted Root from the Directory Tree

- 1 In iManager, select *eDirectory Administration > Modify Object*.
- 2 Browse to the Certificate Authority (CA) object, then click *OK*.
- 3 Select *Certificates* from the drop-down list.
- 4 Click *Export*.
- 5 Click *No* at the prompt that displays *Do you want to export the private key with the certificate?*

- 6 Click *Next*.
- 7 In the *Filename* field, type in a filename (for example, `PublicKeyCert`), then select *Base64* as the format.
- 8 Click *Export*.
- 9 Continue with [Step 6: Importing the Trusted Root Certificate](#).

5.5.6 Step 6: Importing the Trusted Root Certificate

You need to import the trusted root certificate into the LDAP server's trust database and the client's certificate store.

- ♦ [“Importing into the LDAP Server's Trust Database” on page 47](#)
- ♦ [“Importing into the Client's Certificate Store” on page 47](#)

Importing into the LDAP Server's Trust Database

You need to import the trusted root certificate into the LDAP server's trust database. Because the server certificate was signed by the Identity Vault's CA, the trust database needs to be configured to trust the Identity Vault CA.

- 1 In the Netscape Console, click *Tasks > Certificate Setup Wizard > Next*.
- 2 In *Select a Token*, accept the default for *Internal (Software)*.
- 3 In *Is the Server Certificate Already Requested and Ready to Install*, select *Yes*.
- 4 Click *Next* twice.
- 5 In *Install Certificate For* dialog box, select *Trusted Certificate Authority*.
- 6 Click *Next*.
- 7 Select *The Certificate Is Located in This File*, then type the full path to the `.b64` file containing the trusted root certificate.
- 8 Click *Next*.
- 9 Verify the information on the screen, then click *Add*.
- 10 Click *Done*.
- 11 Continue with [Importing into the Client's Certificate Store](#).

Importing into the Client's Certificate Store

You need to import the trusted root certificate into a certificate store (also called a keystore) that the driver can use.

- 1 Use the `KeyTool` class found in `rt.jar`.

For example, if your public key certificate is saved as `PublicKeyCert.b64` on a diskette and you want to import it into a new certificate store file named `.keystore` in the current directory, type the following at the command line:

```
java sun.security.tools.KeyTool -import -alias TrustedRoot -file  
a:\PublicKeyCert.b64
```

```
-keystore .keystore -storepass keystorepass
```

- 2 When you are asked to trust this certificate, select *Yes*, then click *Enter*.
- 3 Copy the `.keystore` file to any directory on the same file system that has the Identity Vault files.
- 4 In iManager, select *Identity Manager > Identity Manager Overview*.
- 5 Search for drivers.
- 6 Click the LDAP Driver object, then click it again in the *Identity Manager Driver Overview* page.
- 7 In the *Keystore Path* parameter, enter the complete path to the `.keystore` file.
- 8 Continue with [Step 7: Adjusting Driver Settings](#).

5.5.7 Step 7: Adjusting Driver Settings

The following table lists the driver's settings and its default values in the sample configurations.

Table 5-3 *Driver Settings and Default Values*

Parameter	Sample Configuration Value	Description
Use SSL for LDAP Connections	no	The value for this parameter should be either <i>Yes</i> or <i>No</i> . It indicates whether or not SSL connections should be used when communicating with the LDAP server. To use SSL, you must also correctly configure the LDAP server. For more information, refer to "Configuring SSL Connections" on page 44 ,
SSL Port	636	This parameter is ignored unless Use SSL for LDAP Connections is set to <i>Yes</i> . It indicates which port the LDAP server uses for secure connections.
Keystore Path (for SSL Certs)	[blank]	When Use SSL for LDAP Connections is set to <i>Yes</i> , this parameter value should be the complete path to the keystore file that contains the trusted root certificate of the Certificate Authority (CA) that signed the server certificate. For more information about creating the keystore file, refer to "Importing into the Client's Certificate Store" on page 47 .

Activating the LDAP Driver

6

Activate the driver within 90 days of installation. Otherwise, the driver won't work.

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

Managing the LDAP Driver

7

- ♦ [Section 7.1, “Starting, Stopping, or Restarting the LDAP Driver,” on page 51](#)
- ♦ [Section 7.2, “Migrating and Resynchronizing Data,” on page 51](#)
- ♦ [Section 7.3, “Using the DirXML Command Line Utility,” on page 52](#)
- ♦ [Section 7.4, “Viewing Driver Version Information,” on page 52](#)
- ♦ [Section 7.5, “Reassociating a Driver Set Object with a Server Object,” on page 56](#)
- ♦ [Section 7.6, “Changing the Driver Configuration,” on page 57](#)
- ♦ [Section 7.7, “Storing Driver Passwords Securely with Named Passwords,” on page 57](#)
- ♦ [Section 7.8, “Adding a Driver Heartbeat,” on page 64](#)

7.1 Starting, Stopping, or Restarting the LDAP Driver

In Designer:

- 1 Open a project in the Modeler, then right-click the driver line.
- 2 Select *Live > Start Driver, Stop Driver, or Restart 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 51](#).

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 LDAP 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.

NOTE: When migrating data from an Identity Vault into the LDAP directory, you might need to change your LDAP server settings to allow migration of large numbers of objects. See [Section 9.2, “Migrating Users into an Identity Vault,” on page 81](#).

- ♦ **Migrate Data into the Identity Vault:** Allows you to define the criteria that Identity Manager uses to migrate objects from an LDAP 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 67](#).

7.3 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 87](#) for information about the DirXML Command Line utility. For daily tasks, use iManager or Designer.

7.4 Viewing Driver Version Information

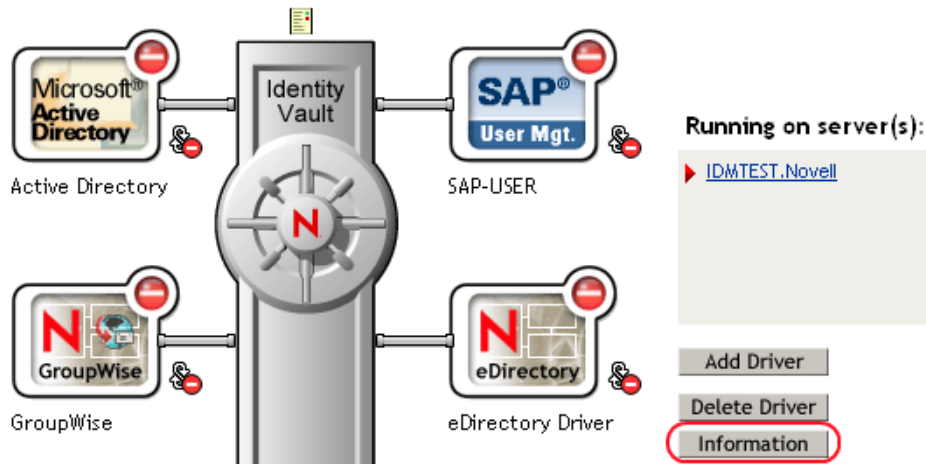
To view information on versions of Identity Manager and versions of drivers, use the Versioning Discovery tool. This feature exists only in iManager.

- ♦ [Section 7.4.1, “Viewing a Hierarchical Display of Version Information,” on page 52](#)
- ♦ [Section 7.4.2, “Viewing the Version Information as a Text File,” on page 54](#)
- ♦ [Section 7.4.3, “Saving Versioning Information,” on page 55](#)

7.4.1 Viewing a Hierarchical Display of Version Information

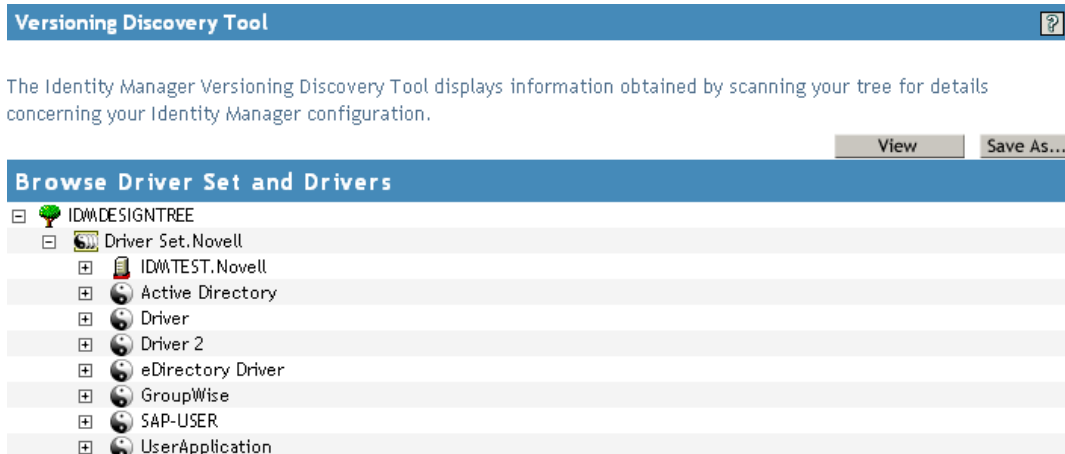
- 1 To find your Driver Set object in iManager, click *Identity Manager > Identity Manager Overview*, then click *Search*.
- 2 In the Identity Manager Overview, click *Information*.

Driver Set: Driver Set.Novell  [Activation required by: November 1, 2006](#) 



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.



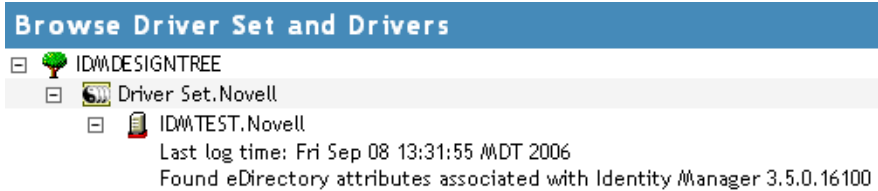
The unexpanded hierarchical view displays the following:

- ♦ 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

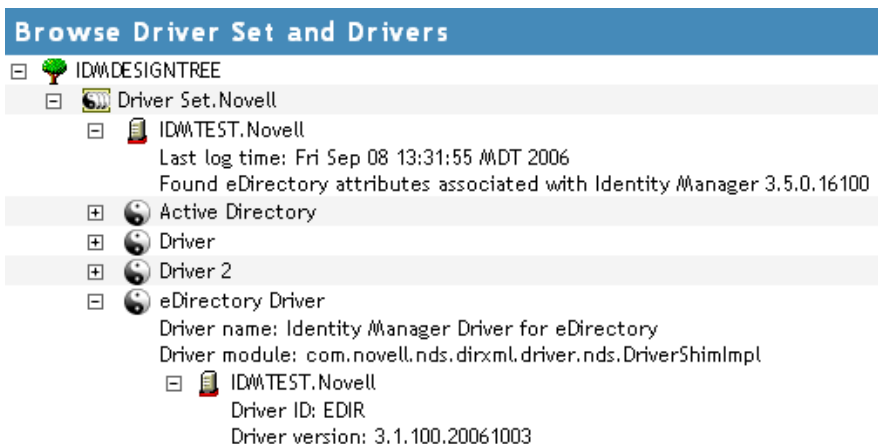
- 4 View version information related to servers by expanding the server icon.



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:

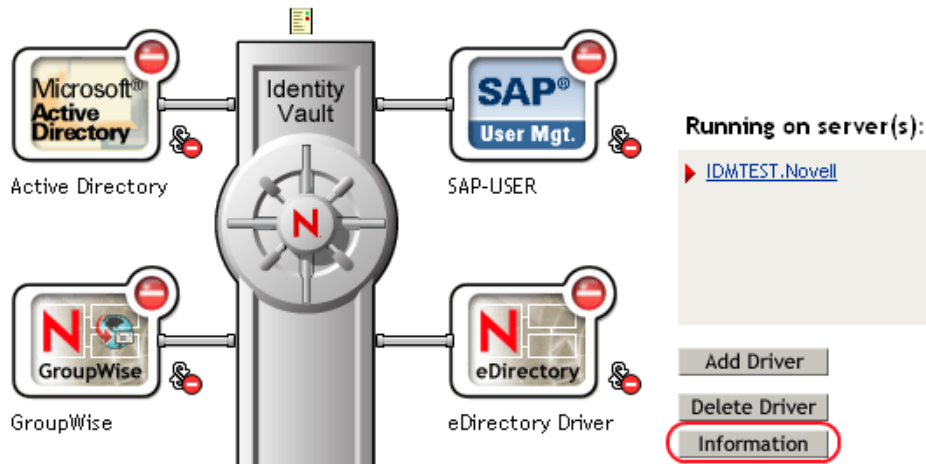
- ◆ The driver ID
- ◆ The version of the instance of the driver running on that server

7.4.2 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.

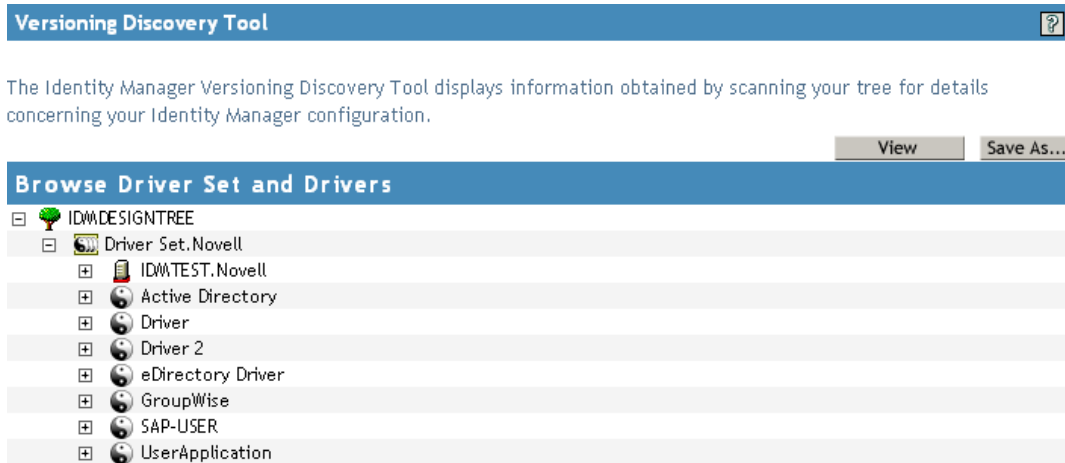
- 1** To find your Driver Set object in iManager, click *Identity Manager > Identity Manager Overview*, then click *Search*.
- 2** In the Identity Manager Overview, click *Information*.

Driver Set: Driver Set.Novell  Activation required by: November 1, 2006 



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*.



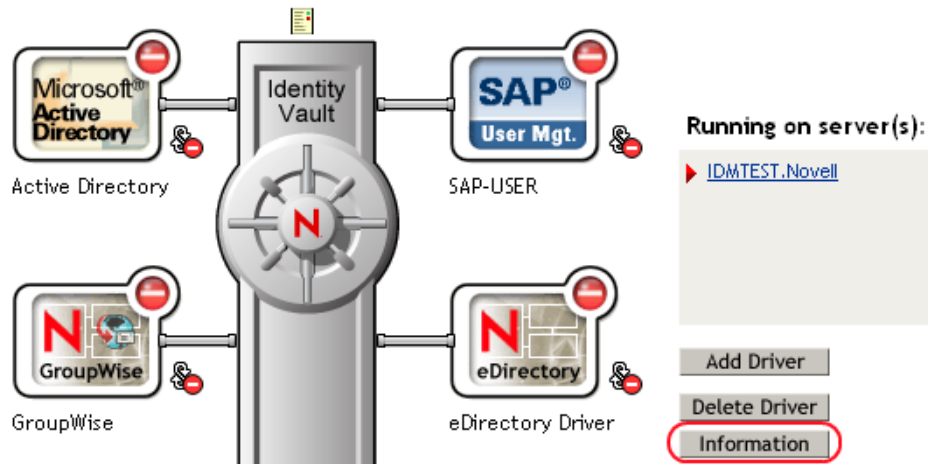
The information is displayed as a text file in the Report Viewer window.

7.4.3 Saving Versioning Information

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

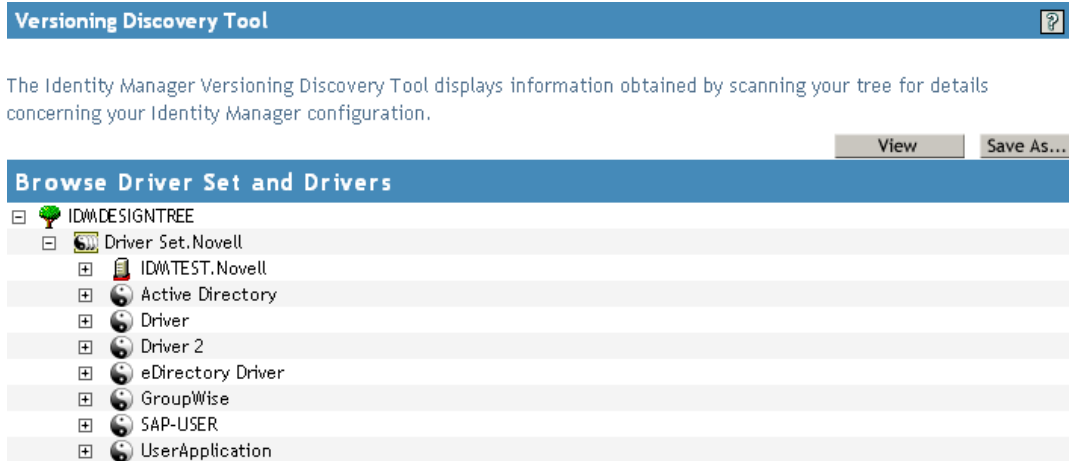
- 1 To find the Driver Set object in iManager, click *Identity Manager > Identity Manager Overview*, then click *Search*.
- 2 In the Identity Manager Overview, click *Information*.

Driver Set: Driver Set.Novell  Activation required by: November 1, 2006 



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.

7.5 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 object 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 on 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*.

7.6 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 list of all of the configuration fields, see [Appendix B, “Properties of the LDAP Driver,” on page 101](#).

7.7 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.

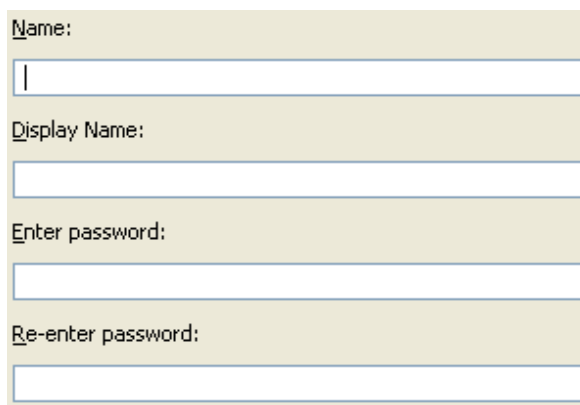
To use a named password in a driver policy, you refer to it by the name of the password, instead of using the actual password, and the Metadirectory engine sends the password to the driver. The

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 7.7.1, “Using Designer to Configure Named Passwords,” on page 58](#)
- ♦ [Section 7.7.2, “Using iManager to Configure Named Passwords,” on page 58](#)
- ♦ [Section 7.7.3, “Using Named Passwords in Driver Policies,” on page 60](#)
- ♦ [Section 7.7.4, “Using the DirXML Command Line Utility to Configure Named Passwords,” on page 61](#)

7.7.1 Using Designer to Configure Named Passwords

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



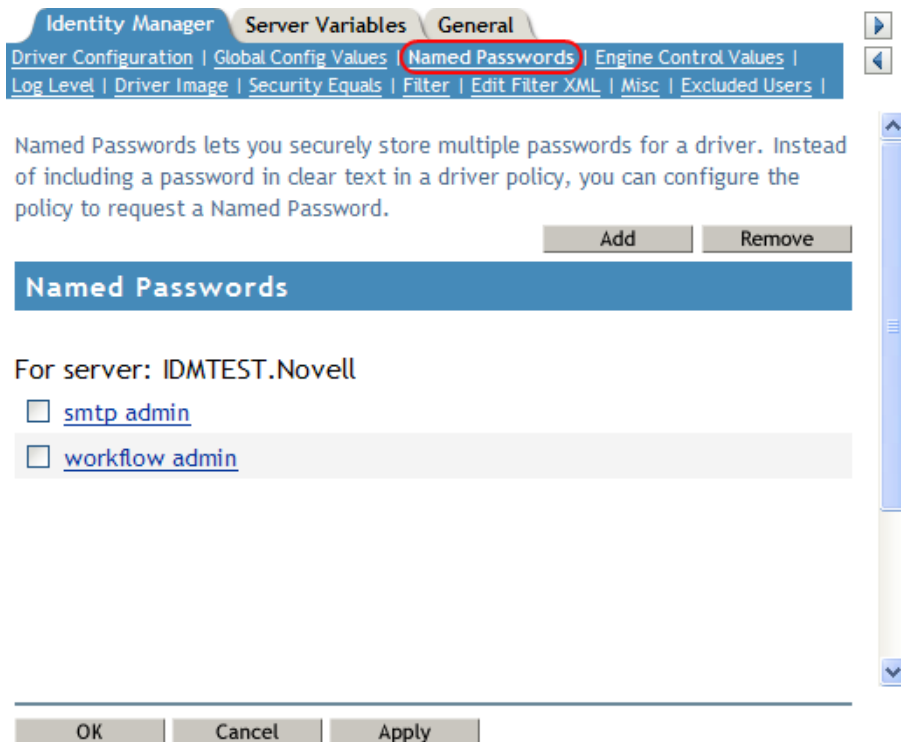
The screenshot shows a configuration dialog box with the following fields:

- Name:** An empty text input field.
- Display Name:** An empty text input field.
- Enter password:** An empty text input field.
- Re-enter password:** An empty text input field.

- 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.

7.7.2 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:

OK Cancel

- 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.

7.7.3 Using Named Passwords in Driver Policies

- ♦ “Making a Call to a Named Password” on page 60
- ♦ “Referencing a Named Password” on page 60

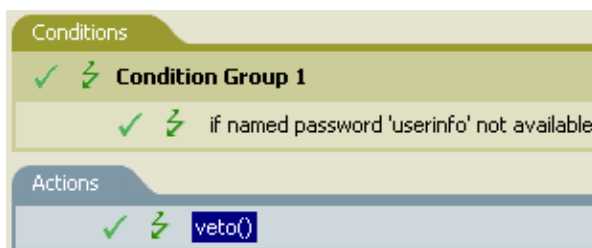
Making a Call to a Named Password

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/>
```

7.7.4 Using the DirXML Command Line Utility to Configure Named Passwords

- ♦ “Creating a Named Password in the DirXML Command Line Utility” on page 61
- ♦ “Removing a Named Password by Using the DirXML Command Line Utility” on page 62

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 87.

- 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, you can 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 87](#).

- 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.

7.8 Adding a Driver Heartbeat

The driver heartbeat is a feature of the Identity Manager drivers that ship with Identity Manager 2 and later. Using it 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 67](#)
- ♦ [Section 8.2, “When Does Synchronization Occur?” on page 67](#)
- ♦ [Section 8.3, “How Does the Metadirectory Engine Decide Which Object to Synchronize?” on page 68](#)
- ♦ [Section 8.4, “How Synchronization Works,” on page 69](#)

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 68.

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 70](#), [Table 8-2 on page 72](#), and [Table 8-3 on page 73](#).

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 sides 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 69](#)
- ♦ [Section 8.4.2, “Scenario Two,” on page 71](#)
- ♦ [Section 8.4.3, “Scenario Three,” on page 72](#)

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

Class Name: User
Attribute Name: Facsimile Telephone Num

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 8-1 on page 70 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 72 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 73 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

- ♦ [Section 9.1, “Troubleshooting Driver Processes,” on page 75](#)
- ♦ [Section 9.2, “Migrating Users into an Identity Vault,” on page 81](#)
- ♦ [Section 9.3, “OutOfMemoryError,” on page 81](#)
- ♦ [Section 9.4, “LDAP v3 Compatibility,” on page 82](#)
- ♦ [Section 9.5, “Frequently Asked Questions,” on page 82](#)

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 75](#)
- ♦ [“Adding Trace Levels in iManager” on page 77](#)
- ♦ [“Capturing Driver Processes to a File” on page 78](#)

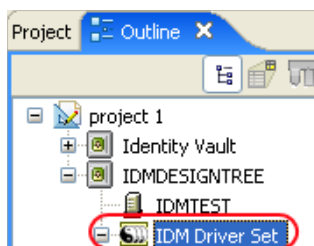
Adding Trace Levels in Designer

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

- ♦ [“Driver Set” on page 75](#)
- ♦ [“Driver” on page 76](#)

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.

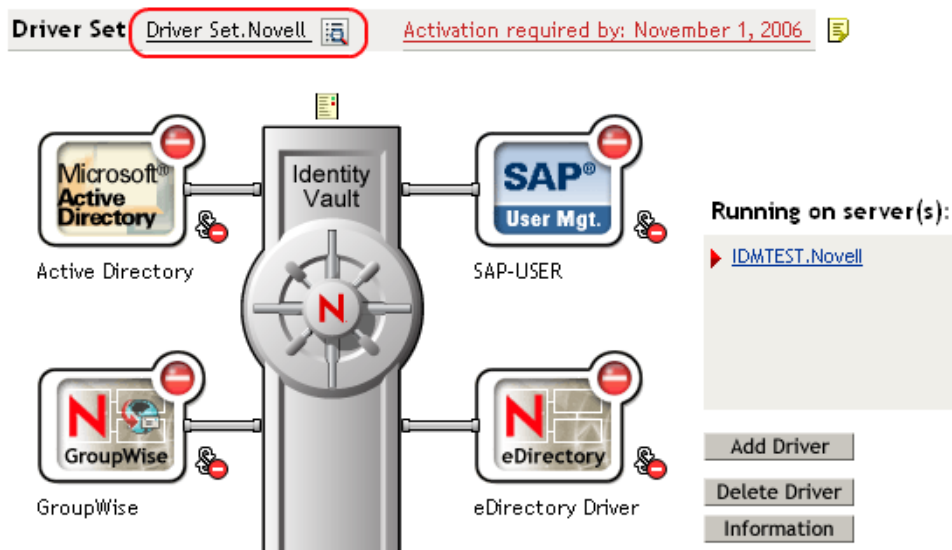
Adding Trace Levels in iManager

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

- ♦ “Driver Set” on page 77
- ♦ “Driver” on page 77

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 78
- ◆ “Windows” on page 78
- ◆ “UNIX” on page 79
- ◆ “iMonitor” on page 79
- ◆ “Remote Loader” on page 80

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`s 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.
- ◆ `ndsimonitor` 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 Migrating Users into an Identity Vault

Some LDAP servers have settings that limit the number of entries that an LDAP query can return. For example, iPlanet Directory Server 5.1 has a default limit of 2000 objects.

When migrating user data from LDAP into an Identity Vault, the driver makes an LDAP query to the server and returns the objects that match the criteria (such as `objectclass=User`).

A limit on the number of entries that can be returned on an LDAP query can cause a migration to stop before it is complete, even though the Identity Manager driver continues to run normally otherwise.

To fix this, change the limit. For example, do the following in iPlanet:

- 1 Go to the *Configuration* tab, then select *Database* settings.
- 2 Raise the look-through limit on the LDBM plug-in tab from default of 5000 to an appropriate number.
This is the number of records the query is allowed to look at while fulfilling the query.
- 3 Go to the *Configuration* tab, select *Directory Server Settings*, select the *Performance* tab, then raise the size limit according to the number of user accounts you need to migrate.
This is the actual number of records that the query is allowed to return.
After these settings have been adjusted, the migration should complete correctly.

9.3 OutOfMemoryError

If you use the LDAP-Search method and the driver shuts down with a `java.lang.OutOfMemoryError`:

- 1 Try setting or increasing the `DHOST_JVM_INITIAL_HEAP` and `DHOST_JVM_MAX_HEAP` environment variables.
- 2 Restart the driver.
- 3 Monitor the driver to make sure that the variables provide enough memory.

For more information, see [TID 10062098 \(http://support.novell.com/cgi-bin/search/searchtid.cgi?/10062098.htm\)](http://support.novell.com/cgi-bin/search/searchtid.cgi?/10062098.htm).

9.4 LDAP v3 Compatibility

The LDAP Driver for Identity Manager works with most LDAP v3 compatible LDAP servers. The driver is written to the LDAP specification (IETF RFCs 4510-4519). To increase compatibility with some LDAP servers that don't fully meet the RFC 2251 requirements, we have added workarounds to the LDAP driver.

One compatibility issue that cannot be ignored nor worked around is the RFC 4511 section 4.1.1 requirement that servers allow Message ID values up to 2,147,483,647 (integer values using four bytes).

Oracle* Internet Directory version 2.1.1.0.0 (which is part of Oracle 8i) allows only Message ID values up to 32,767 (integer values using two bytes). Therefore, it can't function properly with the LDAP Driver for Identity Manager.

If you need compatibility with Oracle Internet Directory, Novell® recommends upgrading to version 9.2.0.1.0 (included with Oracle 9i) or later.

9.5 Frequently Asked Questions

Question: Does the LDAP-search method retrieve everything every time, or just retrieve updates since the last poll?

Answer: The LDAP-search method synchronizes updates from one poll to the next.

Question: If I have a choice between using the LDAP-search method or the changelog method, should I use the LDAP Search method?

Answer: Use the changelog method because it has performance advantages and is the preferred method.

Backing Up the LDAP 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*.

- ♦ [Section 10.1, “Exporting the Driver in Designer,” on page 83](#)
- ♦ [Section 10.2, “Exporting the Driver in iManager,” on page 83](#)

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 87](#)
- ♦ [Section A.2, “Command Line Mode,” on page 96](#)

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 88](#) 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 89 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 93 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/ndsstd/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 91 for a description of these options.</p>
14: <i>Cache operations</i>	<p>Five Cache operations are available. See Table A-4 on page 92 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 7.7, "Storing Driver Passwords Securely with Named Passwords," on page 57 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 Yes, all Named Passwords are cleared. If you enter No, you are prompted to specify the password name that you want to clear.</p>
7: <i>List named passwords</i>	<p>Lists all named passwords that are stored on the Driver object. It lists the password name and the password description.</p>
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>	<p>Exits the current menu and takes you back to the Driver options.</p>

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

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. See Table A-6 on page 94 for a list of items to log.</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. See Table A-6 on page 94 for a list of items to log.</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: Get available job definitions	Allows you to select an existing job. <ul style="list-style-type: none">◆ Enter the job number:◆ Do you want to filter the job definitions by containment? Enter Yes or No◆ Enter name of the file for response: Examples: NetWare: sys:\files\user.log Windows: c:\files\user.log Linux: /files/user.log
2: Operations on specific job object	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 96](#) 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	
-user <user name>	Specify the name of a user with administrative rights to the drivers you want to test.
-host <name or IP address>	Specify the IP address of the server where the driver is installed.
-password <user password>	Specify the password of the user specified above.
-port <port number>	Specify a port number, if the default port is not used.
-q <quiet mode>	Displays very little information when a command is executed.
-v <verbose mode>	Displays detailed information when a command is executed.
-s <stdout>	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 94 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 99](#) 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 LDAP Driver


B

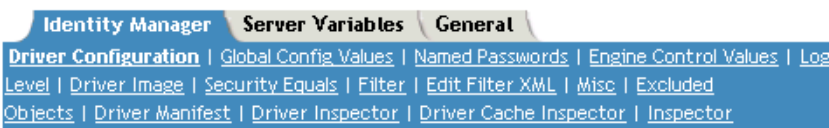
This section is a reference for all fields on the driver’s property pages as displayed in iManager and Designer. Some fields display differently in iManager than in Designer.

The information is organized according to tabs that display in iManager. If a field is different in Designer, it is marked with a Designer  icon.

The following figure illustrates property pages in iManager:

Figure B-1 Property Pages in iManager

Modify Object:  LDAP.vmp



- ◆ [Section B.1, “Identity Manager: Driver Configuration,” on page 101](#)
- ◆ [Section B.2, “Identity Manager: Global Configuration Values,” on page 105](#)
- ◆ [Section B.3, “Identity Manager: Named Passwords,” on page 106](#)
- ◆ [Section B.4, “Identity Manager: Engine Control Values,” on page 107](#)
- ◆ [Section B.5, “Identity Manager: Log Level,” on page 109](#)
- ◆ [Section B.6, “Identity Manager: Driver Image,” on page 110](#)
- ◆ [Section B.7, “Identity Manager: Security Equals,” on page 110](#)
- ◆ [Section B.8, “Identity Manager: Filter,” on page 110](#)
- ◆ [Section B.9, “Identity Manager: Edit Filter XML,” on page 111](#)
- ◆ [Section B.10, “Identity Manager: Misc,” on page 111](#)
- ◆ [Section B.11, “Identity Manager: Excluded Users,” on page 112](#)
- ◆ [Section B.12, “Identity Manager: Driver Manifest,” on page 112](#)
- ◆ [Section B.13, “Driver Cache Inspector,” on page 113](#)
- ◆ [Section B.14, “Driver Inspector,” on page 113](#)
- ◆ [Section B.15, “Server Variables,” on page 114](#)

B.1 Identity Manager: 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 > Driver Configuration*.

In Designer:

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

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

- ◆ [Section B.1.1, “Driver Module,” on page 102](#)
- ◆ [Section B.1.2, “Driver Object Password,” on page 103](#)
- ◆ [Section B.1.3, “Authentication,” on page 103](#)
- ◆ [Section B.1.4, “Startup Option,” on page 104](#)

B.1.1 Driver Module

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



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*.

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.

Table B-1 Settings: Driver Module

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.
<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 on the driver.

B.1.2 Driver Object Password

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*.

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*.

Table B-2 Settings: Driver Object 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.

B.1.3 Authentication

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

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*.

In Designer:

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

Table B-3 Settings: Authentication

Option	Description
Authentication information for server	Displays or specifies the IP address or server name that the driver is associated with

Option	Description
<i>LDAP Authentication DN</i> or  <i>User ID</i>	Specifies the DN of the LDAP account that the driver will use for authentication. Example: Administrator
<i>Authentication Context</i> or  <i>Connection Information</i>	Specify the IP address or name of the server the application shim should communicate with.
<i>Remote Loader Connection Parameters</i> or  <i>Host name</i>  <i>Port</i>  <i>KMO</i>  <i>Other parameters</i>	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>
<i>Driver Cache Limit (kilobytes)</i> or  <i>Cache limit (KB)</i>	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.
<i>Application Password</i> or  <i>Set Password</i>	Specify the password for the user object listed in the <i>Authentication ID</i> field.
<i>Remote Loader Password</i> or  <i>Set Password</i>	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.

B.1.4 Startup Option

The Startup Option allows you to set the driver state when the Identity Manager server 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*.

In Designer:

- 1 Open a project in the Modeler.

- 2 Right-click the driver line, then select *Properties > Driver Configuration*.
- 3 Click *Startup Option*.

Table B-4 Settings: 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.

B.2 Identity Manager: 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 Configuration Values*.

Table B-5 Settings: Password Configuration

Option	Description
<i>Application accepts passwords from Identity Manager</i>	If <i>True</i> , allows passwords to flow from the Identity Manager data store to the connected system.
<i>Identity Manager accepts passwords from application</i>	If <i>True</i> , 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 <i>True</i> , 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 <i>True</i> , 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 <i>True</i> , 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. The e-mail notification templates use this value.

B.3 Identity Manager: 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 configured Named Passwords, see [Section 7.7, “Storing Driver Passwords Securely with Named Passwords,” on page 57](#).

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 Identity Manager: 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-6 Settings: 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 <i>True</i> 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 <i>True</i> 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 backward-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>
<hr/> <p>NOTE: This control does not limit the number of application objects that can be migrated; it merely limits the batch size.</p> <hr/>	
<i>Set creatorsName on objects created in Identity Vault</i>	<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 <i>False</i> 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 <i>True</i> 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 Identity Manager: 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 *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*.

Table B-7 Settings: 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 Identity Manager: 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 Identity Manager: 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 Identity Manager: 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 Identity Manager: 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 Identity Manager: 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*.

Table B-8 *Settings: Misc*

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.
<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 Identity Manager: 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 Identity Manager: 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:

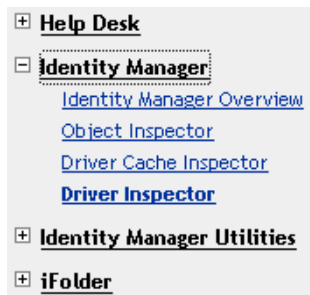
- 1 Click *Identity Manager > Identity Manager Overview*, then click *Search* to search for the driver set that is associated with the driver.
- 2 Browse to the driver, then click the upper right corner of the driver icon.
- 3 Click *Edit Properties > Driver Manifest*.

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-2 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.

- ◆ **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>
<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>

Option	Description
<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>	<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> <hr/> <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><i>Always accept password; ignore Password Policies</i></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>
<p><i>Application accepts passwords (Subscriber Channel)</i></p>	<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>
<p><i>Notify the user of password synchronization failure via email</i></p>	<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> <hr/> <p>NOTE: To set up e-mail notification, select <i>Passwords > Edit EMail Templates</i>.</p>

Documentation Updates

C

This section contains new or updated information on the Identity Manager Driver for LDAP.

The documentation is provided on the Web in two formats: HTML and PDF. The HTML and PDF documentation are both kept up-to-date with the documentation changes listed in this section.

If you need to know whether a copy of the PDF documentation you are using is the most recent, check the date that the PDF file was published. The date is on the title page.

New or updated documentation was published on the following dates:

- ◆ [Section C.1, “October 15, 2007,” on page 117](#)

C.1 October 15, 2007

Table C-1 *Updates as of October 15, 2007*

Location	Change
Section 2.3.1, “Manually Adding Publisher Settings,” on page 21	<p>In the XML code, corrected “Password” name=“pub password-encryption-key” by adding a necessary hyphen: Password” name=“pub-password-encryption-key”.</p> <p>Reformatted the XML code so that it appears correctly on the documentation Web site.</p>