Novell
Identity Manager 3.5.1 Driver for Lotus Notes*

3.5.1

www.novell.com

IMPLEMENTATION GUIDE

September 28, 2007

Legal Notices

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

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

Novell Trademarks

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

Third-Party Materials

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

Contents

4 Customizing the Driver 49

5 Activating the Driver 105

6 Managing the Driver 107

About This Guide

The Identity Manager Driver for Lotus Notes is designed to automatically let you synchronize data in an eDirectory™ tree with data stored in a Domino* Directory or another Notes database. This configurable solution allows you to increase productivity and streamline business processes by integrating Lotus Notes and eDirectory.

The guide contains the following sections:

Audience

This guide is intended for consultants, administrators, and IS personnel who need to install, configure, and maintain the Identity Manager Driver for Lotus Notes.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation, or go to Novell’s Feedback Web site (http://www.novell.com/documentation/feedback.html) and enter your comments there.

Documentation Updates

For the most recent version of this document, see the Drivers Web site (http://www.novell.com/documentation/idm35drivers/index.html).

Additional Documentation

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

Documentation Conventions

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

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

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

Overview

1

The Identity Manager Driver for Lotus Notes lets you synchronize data in a Novell® eDirectory™ tree with data stored in a Domino Directory or another Notes database.

The Identity Manager Driver for Notes is essentially an application programming interface (API) translator that maps object data represented in an XML document between the Identity Vault and the appropriate Lotus Domino Toolkit for Java* object methods.

- Section 1.1, "Changes in Terminology," on page 11
- Section 1.2, "Notes Driver Concepts," on page 11
- Section 1.3, "Key Driver Features," on page 13
- Section 1.4, "Notes Driver Features," on page 14
- Section 1.5, "Driver Components and Configuration," on page 15

1.1 Changes in Terminology

The following terms have changed from earlier releases:

Table 1-1 *Changes in Terminology*

Earlier Terms	New Terms
DirXML®	Identity Manager
DirXML Server	Metadirectory server
DirXML engine	Metadirectory engine
eDirectory	Identity Vault (except when referring to eDirectory attributes or classes)

1.2 Notes Driver Concepts

Identity Manager fundamentals are explained in the "Overview of the Identity Manager Architecture" in the *Novell Identity Manager 3.5.1 Administration Guide*. The Overview discusses the driver architecture in general, and the Guide contains a section on "Managing Identity Manager Drivers."

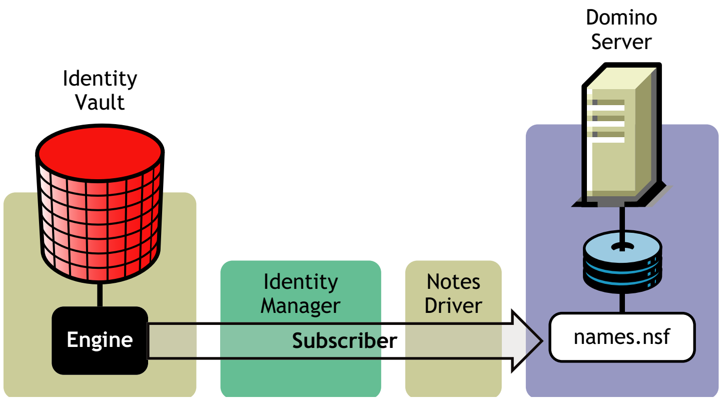
1.2.1 Default Data Flow

A channel is a combination of rules, policies, and filters that are used to synchronize data between two systems. The Subscriber and Publisher channels describe the direction in which the data flows. The Subscriber channel takes the event from Identity Vault (eDirectory) and sends that event to the receiving system (Lotus Notes). The Publisher channel takes the event from Lotus Notes, and sends that event to the Identity Vault. The Subscriber and Publisher channels act independently; actions in one channel are not affected by what happens in the other.

Subscriber Channel

The Subscriber channel is the channel of communication from the Identity Vault to Lotus Notes. The following illustration shows this data flow:

Figure 1-1 *Data Flow Through the Subscriber Channel*

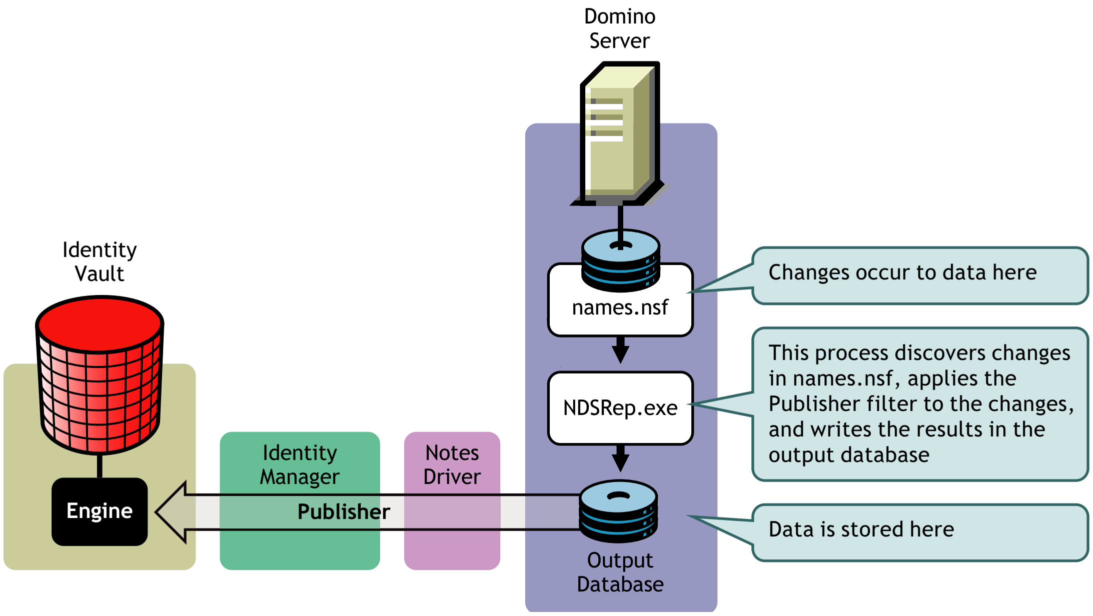


The driver can be configured to work with Notes databases other than the Public Address Book, `names.nsf`.

Publisher Channel

The Publisher channel represents the channel of communication from Lotus Notes to the Identity Vault. The following illustration shows how this data is published:

Figure 1-2 *Data Flow Through the Publisher Channel*


Domino Server
Identity Vault
names.nsf
Changes occur to data here
NDSRep.exe
This process discovers changes in names.nsf, applies the Publisher filter to the changes, and writes the results in the output database
Identity Manager
Notes Driver
Engine
Publisher
Data is stored here
Output Database

1.2.2 Policies

Policies are used to control the synchronization of data between the Identity Vault and the application, database, or directory. Policies transform an event on a channel input into a set of commands on the channel output. The Lotus Notes driver includes the following set of preconfigured policies:

- Schema Mapping: Mappings have been defined for the Notes address book.
- Creation: The default Creation policy logic for the Publisher channel and the Subscriber channel is the same. To create a User object requires a Given name and a Surname. To create a Group object requires Description, Membership, and Owner attributes. You can modify these elements to meet your business policies.
- Matching: The default Matching policy logic for the Publisher channel and the Subscriber channel is the same. An eDirectory User object is considered to be the same object in Notes when Given name and Surname match in both directories. An eDirectory Group object is considered to be the same object in Notes when the CN is the same in both directories. It is recommended that you modify these policies to meet your business policies. Matching policies are typically more specific than this generic Given name/Surname match.
- Placement: The default Placement policy on the Subscriber channel places all User objects from a specified Identity Vault container in a specified Notes Organizational Unit, and all Group objects from a specified Identity Vault container in a specified Organizational Unit in Notes. The same relationship is typically maintained on the Publisher channel. The container names and OU names for this default Placement policy are collected from the user when importing the default driver configuration. It is recommended that you modify or add additional Placement policies and policy rules to meet your business needs.

1.3 Key Driver Features

The sections below contains a list of the key driver features.

- Section 1.3.1, “Local Platforms,” on page 13
- Section 1.3.2, “Remote Platforms,” on page 14
- Section 1.3.3, “Role-Based Entitlements,” on page 14

1.3.1 Local Platforms

The Lotus Notes driver can be installed locally on the following platforms:

- Windows* NT*, 2000, or 2003 with the latest Service Patch
- Novell Open Enterprise Server with the latest Support Pack
- Linux Red Hat* 3.0 and 4.0 for AMD64/EM64T
- SUSE® Linux Enterprise Server 9 and 10 (with the latest Support Pack)
- Solaris* 9 or 10
- AIX* 5.2L, v5.2 and v5.3

1.3.2 Remote Platforms

The Lotus Notes driver can use the Remote Loader service. The Remote Loader service for the Notes driver can be installed on the following platforms:

- Windows NT, 2000, or 2003 with the latest Service Patch
- Novell Open Enterprise Server with the latest Support Pack
- Linux Red Hat 3.0 and 4.0 for AMD64/EM64T
- SUSE Linux Enterprise Server 9 and 10 (with the latest Support Pack)
- Solaris 9 or 10
- AIX 5.2L, v5.2 and v5.3

For more information about installing the Remote Loader services, see "Installing the Remote Loader" in the *Novell Identity Manager 3.5.1 Administration Guide*.

1.3.3 Role-Based Entitlements

The Lotus Notes driver can be configured to use entitlements to manage user accounts and group membership in the Lotus Notes Address Book. When using entitlements, this driver works in conjunction with external services, such as the User Application with workflow provisioning or Role-Based Entitlements, to manage entitlement functionality.

If you want to use the preconfigured entitlements for the Lotus Notes driver and the infrastructure that supports them, you must enable entitlements when you initially create a driver in Designer or in iManager. To do this, answer *Yes* to the Enable Entitlements question as you import the driver file. The preconfigured policies and rules that support the preconfigured entitlements cannot be imported later without re-creating the driver.

The Notes driver also supports customized entitlements, if there are policies created for the driver to consume.

1.4 Notes Driver Features

1.4.1 New Driver Features

The following Lotus Notes driver features are in the Identity Manager Driver 3.5.1 for Lotus Notes.

GVC controlled values:

- Optional sample policy to generate a unique ShortName field value for new Notes users.
- Optional sample policy to generate a unique Internet Email Address for new Notes users.

1.4.2 Identity Manager Features

For information about the new features in Identity Manager, see "What's New in Identity Manager 3.5.1? " in the *Identity Manager 3.5.1 Installation Guide*.

1.5 Driver Components and Configuration

The driver contains the following components:

- **Default Driver Configuration File:** A driver configuration file is a file you can import to set up default rules, style sheets, and driver parameters. The driver configuration file included with this driver is `Notes-IDM3_5-V2.xml`, with its accompanying `.xlf` file (for any language other than English).
- **Driver Files:** `CommonDriverShim.jar` and `NotesDriverShim.jar` are the Java files that direct synchronization between Lotus Notes and the Identity Vault.
- **notesdrvjni:** This shared library provides Java Native Interface (JNI) access from `NotesDriverShim.jar` to Lotus Notes native libraries (Notes C APIs).
- **ndsrep:** Ndsrep is a Lotus Domino server add-in process to enable data synchronization. It keeps track of the time of the last successful synchronization with a Notes database, and checks the Lotus Domino Server for changes based on that time stamp. It then reads the changes from the Notes database, determines the event types they represent, and filters the updates based on objects and attributes specified in the Publisher filter in the driver configuration in the Identity Vault.
- **dsrepcfg.ntf:** A Notes database template required for the initial startup of the Notes driver shim. The Notes driver shim uses this Notes database template to create a configuration database named `dsrepcfg.nsf` used by ndsrep to determine the Publisher filter and other driver publication settings.

Installing and Configuring the Driver

2

This section contains a road map for successfully installing and configuring the driver. There are tasks you must do before you install, tasks you only do on the Lotus Domino server side, tasks you only do on the Novell® eDirectory™ and Identity Manager side, and tasks you do after the installation. The order in which you do these tasks is important. Complete the tasks in the order listed:

- Section 2.1, "Requirements for the Driver," on page 17
- Section 2.2, "Where to Install the Notes Driver," on page 18
- Section 2.3, "Preparing Lotus Notes for Synchronization," on page 20
- Section 2.4, "Installing the Lotus Notes Driver During Identity Manager Installation," on page 21
- Section 2.5, "Importing the Notes Driver Configuration File in iManager," on page 24
- Section 2.6, "Installing the Lotus Notes Driver through Designer," on page 25
- Section 2.7, "Setting Up the Driver," on page 28

2.1 Requirements for the Driver

If you are running the Lotus Notes driver from the Domino server, that computer must be running the following software:

- One of the following with Lotus Notes R5.0.8 or later:
  - Windows NT
  - Windows 2000 Server
  - Windows 2000 Professional
  - Windows 2003
  - Windows XP

  Use the operating system versions required by Lotus Domino.
- One of the following with Lotus Notes v6.x, or v7.x:
  - Windows NT
  - Windows 2000 Server
  - Windows 2000 Professional
  - Windows 2003
  - Linux
  - AIX

    If your Notes system is unable to load tasks on the server, you might need to apply PTF 486444 for AIX 5.2.
  - Solaris

If you are using the driver on Solaris, you should edit the `/etc/system` file on Solaris to include the following line:

```
set msgsys:msginfo_msgtql=1024
```

NOTE: This and other tips are listed in a document published by Sun*, "Domino on Solaris: Common Tuning Tips" (http://www.sun.com/third-party/global/lotus/technical).

Use the operating system versions required by Lotus Domino.

If you are running the Lotus Notes driver from a computer that has only the Lotus Notes client installed, you need:

- Lotus Notes R5.0.8 or later
- Windows NT
- Windows 2000 Server
- Windows 2000 Professional
- Windows 2003
- Windows XP

NOTE: The Lotus Notes driver does not run on Windows 95, Windows 98, or any Macintosh* OS.

2.2 Where to Install the Notes Driver

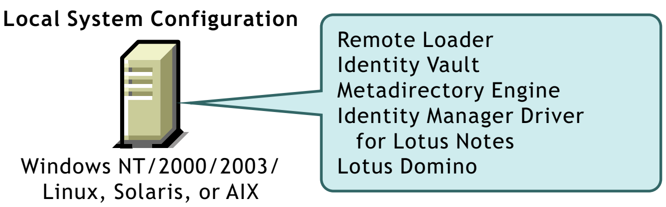
You must decide whether to install the Lotus Notes driver locally or remotely.

- Section 2.2.1, "Installing Locally," on page 18
- Section 2.2.2, "Installing Locally On a Notes Client Workstation," on page 19
- Section 2.2.3, "Installing the Driver Remotely on a Domino Server," on page 19

2.2.1 Installing Locally

A local installation installs the Lotus Notes driver on the same computer where you have installed the Lotus Domino server, the Identity Vault, and Identity Manager. However, if you find there is too much contention for Domino server resources, consider Section 2.2.2, "Installing Locally On a Notes Client Workstation," on page 19.

Figure 2-1 *Use the Remote Loader for Local System Configurations*



It is recommended that you always use the Remote Loader to load the Identity Manager driver for Lotus Notes, even if the driver is installed on the same machine as the Identity Vault and Identity Manager. For Windows installation, after you follow the steps in "Installing Identity Manager on Windows", complete the installation process by following the steps in "Installing the Connected

System Option on Windows ” in the *Identity Manager 3.5.1 Installation Guide*. See Section 2.4.2, “Installing the Lotus Notes Driver on Windows with the Identity Manager Connected System,” on page 22 for the main steps.

For UNIX/Linux installations, see Section 2.4.3, “Installing the Lotus Notes Driver through the GUI Interface On Linux Platforms,” on page 23 or Section 2.4.4, “Using the Command Line to Install the Driver on UNIX/Linux,” on page 23.

2.2.2 Installing Locally On a Notes Client Workstation

A local installation on a Notes client workstation installs the driver on the same computer where you have installed the Lotus Notes client. This option is the recommended installation practice.

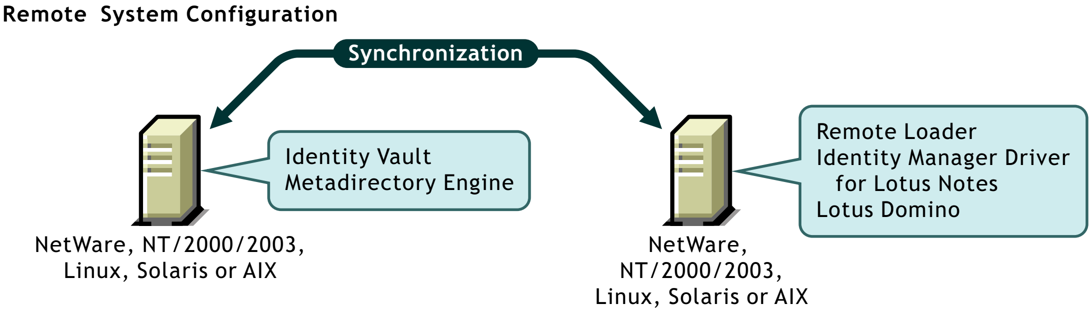
It is recommended that you always use the Remote Loader to load the Identity Manager driver for Lotus Notes. For Windows installation, see Section 2.4.2, “Installing the Lotus Notes Driver on Windows with the Identity Manager Connected System,” on page 22. For UNIX/Linux installations, see Section 2.4.3, “Installing the Lotus Notes Driver through the GUI Interface On Linux Platforms,” on page 23 or Section 2.4.4, “Using the Command Line to Install the Driver on UNIX/Linux,” on page 23.

See also “Installing on a Notes Client Workstation Running Windows” on page 31.

2.2.3 Installing the Driver Remotely on a Domino Server

A remote installation typically installs the driver on a different computer than the one where Identity Manager and the Identity Vault are installed. You must use this option when Domino and the Identity Vault are not on the same server. See Section 2.4.2, “Installing the Lotus Notes Driver on Windows with the Identity Manager Connected System,” on page 22.

Figure 2-2 *When Domino and the Identity Vault Are On Different Servers*



When you install the Notes driver on a Domino server that is running on AIX, Solaris, or Linux, the driver requires the Remote Loader to be installed in the same location. Novell recommends using the Remote Loader with the Notes driver in order to reduce the effects of potential errors caused by a driver policy, which might adversely affect the Domino server’s operation. See Section 2.4.3, “Installing the Lotus Notes Driver through the GUI Interface On Linux Platforms,” on page 23 or Section 2.4.4, “Using the Command Line to Install the Driver on UNIX/Linux,” on page 23.

2.3 Preparing Lotus Notes for Synchronization

Complete the setup tasks in this section to ensure that your Lotus Notes system works with Identity Manager.

- Section 2.3.1, "Collecting Configuration Information," on page 20
- Section 2.3.2, "Creating Lotus Notes Accounts and Groups," on page 20
- Section 2.3.3, "Providing Access to Certifiers and ID Files in the Lotus Notes Infrastructure," on page 20

2.3.1 Collecting Configuration Information

You need to provide a number of system-specific details when you import the driver configuration for Lotus Notes. Some of these details can be collected before you complete the following procedures, and others are defined during the process.

See the list in Table 2-1 on page 26.

2.3.2 Creating Lotus Notes Accounts and Groups

1. Create a Notes User ID to be used exclusively by the driver and give it manager-level ACL access to the target Notes database (usually `names.nsf`), the output database (`ndsrep.nsf`) created by ndsrep, and `certlog.nsf`. If you are synchronizing with the `names.nsf` database, you should select (turn on) all ACL roles (GroupCreator, GroupModifier, NetCreator, NetModifier, PolicyCreator, PolicyModifier, PolicyReader, ServerCreator, ServerModifier, UserCreator, UserModifier).
2. If a Deny Access group doesn't already exist, create this group by using the Lotus Domino Administrator tool.

   This group is used to hold disabled user accounts. You can learn how to set the Deny Access group parameter at Section C.1, "Driver Configuration," on page 161.
3. The installation procedure should take care of Universal ID issues. If you are having problems with Universal IDs, see Section 8.2.1, "Creating Lotus Notes Accounts and Groups," on page 135.

2.3.3 Providing Access to Certifiers and ID Files in the Lotus Notes Infrastructure

The Notes driver user needs access rights to the following:

- Its own user certifier ID file for the driver user in Notes
- The certifier ID files for the Notes certifiers that you want the driver to create users for
- File access to a place where the driver can create new user certifier ID files (optional; depends on whether you want the driver to have this ability)

2.4 Installing the Lotus Notes Driver During Identity Manager Installation

During the installation of Identity Manager, you can select multiple components to install. "Installing Identity Manager on Windows" in the *Identity Manager 3.5.1 Installation Guide* walks you through installing the Metadirectory engine with the Lotus Notes driver and its driver utilities. Use this procedure if you are installing the Lotus Notes driver on the same computer where you are installing the Lotus Domino server, the Identity Vault, and Identity Manager.

It is recommended that you always use the Remote Loader to load the Identity Manager driver for Lotus Notes, even if the driver is installed on the same machine as the Identity Vault and Identity Manager. For Windows installation, after you follow the steps in Section 2.4.1, "Installing the Lotus Notes Driver on Windows with the Identity Manager Metadirectory Server," on page 22, complete the Remote Loader installation process by following the steps in Section 2.4.2, "Installing the Lotus Notes Driver on Windows with the Identity Manager Connected System," on page 22.

A local installation on a Notes client workstation installs the driver on the same computer where you have installed the Lotus Notes. This option is the most recommended installation practice. It is recommended that you use the Remote Loader to load the Identity Manager driver for Lotus Notes. For Windows installation, see Section 2.4.2, "Installing the Lotus Notes Driver on Windows with the Identity Manager Connected System," on page 22. For UNIX/Linux installation, see Section 2.4.3, "Installing the Lotus Notes Driver through the GUI Interface On Linux Platforms," on page 23 or Section 2.4.4, "Using the Command Line to Install the Driver on UNIX/Linux," on page 23.

A remote installation typically installs the Lotus Notes driver on a different computer than the one where Identity Manager and the Identity Vault are installed. You must use this option when Domino and the Identity Vault are not on the same server. If you are installing the Notes driver on a Domino server running on Windows, go to Section 2.4.2, "Installing the Lotus Notes Driver on Windows with the Identity Manager Connected System," on page 22.

When you install the Notes driver on a Domino server that is running on AIX, Solaris, or Linux, the driver requires the Remote Loader to be installed in the same location. Novell recommends using the Remote Loader with the Notes driver in order to reduce the effects of potential errors caused by a driver policy, which may adversely affect the Domino server's operation. See Section 2.4.3, "Installing the Lotus Notes Driver through the GUI Interface On Linux Platforms," on page 23 or Section 2.4.4, "Using the Command Line to Install the Driver on UNIX/Linux," on page 23 and select 2 to install the connected system software (Remote Loader).

2.4.1 Installing the Lotus Notes Driver on Windows with the Identity Manager Metadirectory Server

Follow the steps found in "Installing Identity Manager on Windows" in the *Identity Manager 3.5.1 Installation Guide* to install the Metadirectory engine with the Lotus Notes driver and its driver utilities. Ensure the following steps are a part of the installation process.

1. During the Identity Manager installation process, select the *Novell Identity Manager Metadirectory Server* option, as well as the drivers you want to install, including the Lotus Notes driver.
2. (Optioinal) Select the components to install, including the *Lotus Notes Discovery Tool* option if you need this tool.
3. After you have installed the Identity Manager Metadirectory Server, it is recommended that you complete the Remote Loader installation process by following the steps in Section 2.4.2, "Installing the Lotus Notes Driver on Windows with the Identity Manager Connected System," on page 22.
4. To configure the Lotus Notes driver, see Section 2.5, "Importing the Notes Driver Configuration File in iManager," on page 24 or Section 2.6, "Installing the Lotus Notes Driver through Designer," on page 25.

2.4.2 Installing the Lotus Notes Driver on Windows with the Identity Manager Connected System

Use this installation procedure for a local installation on a Notes client workstation, and for a remote installation when Domino and the Identity Vault are not on the same server.

Follow the steps found in "Installing the Connected System Option on Windows " in the *Identity Manager 3.5.1 Installation Guide* to install the Lotus Notes driver and its driver utilities. Ensure that the following steps are a part of the installation process.

1. During the Identity Manager installation process, first click *Clear All*, then select *Connected System* and *Utilities*. You should also select *Identity Manager Web Components* if you have the iManager utility installed on this server and you want Identity Manager plug-ins for Identity Manager and driver configurations added.
2. Choose a directory to install the Novell Identity Manager Connected System, or keep the default. Click *Next*.
3. On the Components To Install page, the Application Components selection installs components for such application systems as Lotus Notes. Select the utilities you want to install.
4. If you are running iManager on this box, select *Application Components* and click *Next*.
5. (Optional) Select the components to install, including the *Lotus Notes Discovery Tool* option if you need this tool, then click *Next*.
6. To configure the Lotus Notes driver, see Section 2.5, "Importing the Notes Driver Configuration File in iManager," on page 24 or Section 2.6, "Installing the Lotus Notes Driver through Designer," on page 25.

2.4.3 Installing the Lotus Notes Driver through the GUI Interface On Linux Platforms

A remote installation typically installs the Lotus Notes driver on a different computer than the one where Identity Manager and the Identity Vault are installed. You must use this option when Domino and the Identity Vault are not on the same server.

When you install the Notes driver on a Domino server that is running on AIX, Solaris, or Linux, the driver requires the Remote Loader to be installed in the same location. Novell recommends using the Remote Loader with the Notes driver in order to reduce the effects of potential errors caused by a driver policy, which might adversely affect the Domino server's operation.

Follow the steps found in "Installing Identity Manager through the GUI Interface on UNIX/Linux Platforms" in the *Identity Manager 3.5.1 Installation Guide* to install the Lotus Notes driver and its driver utilities. Ensure the following steps are a part of the installation process.

1. During the Identity Manager installation process, select *Custom*. Select *Remote Loader* and *Lotus Notes Driver* on the Choose Install Set page. You should also select *Identity Manager Plug-ins* if you have the iManager utility installed on this server and you want Identity Manager plug-ins for Identity Manager and driver configurations added.
2. By default, Identity Manager driver utilities are not installed on Linux/UNIX installations. You must manually copy the utilities from the Identity Manager installation CD to the Identity Manager server. All utilities are found under the platform's `\setup\utilities` directory.
3. To configure the Lotus Notes driver, see Section 2.5, "Importing the Notes Driver Configuration File in iManager," on page 24 or Section 2.6, "Installing the Lotus Notes Driver through Designer," on page 25.

If you have installed the Identity Manager Metadirectory Server instead of the Remote Loader, it is still recommended that you rerun the installation procedure and install the Remote Loader through the *Connected System* option.

2.4.4 Using the Command Line to Install the Driver on UNIX/ Linux

Follow the steps found in "Using the Console To Install the Connected System Option on UNIX/ Linux" in the *Identity Manager 3.5.1 Installation Guide* to install the Remote Loader and the Lotus Notes driver. Ensure that the following steps are a part of the installation process.

1. During the Identity Manager installation process, select *4-Customize*. Select *Remote Loader* and *Lotus Notes Driver* on the Choose Install Set page. You should also select *Identity Manager Plug-ins* if you have the iManager utility installed on this server and you want Identity Manager plug-ins for Identity Manager and driver configurations added.
2. By default, Identity Manager driver utilities are not installed on Linux/UNIX installations. You must manually copy the utilities from the Identity Manager installation CD to the Identity Manager server. All utilities are found under the *platform's* `\setup\utilities` directory.
3. To configure the Lotus Notes driver, see Section 2.5, "Importing the Notes Driver Configuration File in iManager," on page 24 or Section 2.6, "Installing the Lotus Notes Driver through Designer," on page 25.

If you have installed the Identity Manager Metadirectory Server instead of the Remote Loader, it is still recommended that you rerun the installation procedure and install the Remote Loader through the Connected System option.

2.5 Importing the Notes Driver Configuration File in iManager

The Create Driver Wizard helps you import the basic driver configuration file. This file creates and configures the objects and policies needed to make the driver work properly.

1. In Novell iManager, click *Identity Manager Utilities* > *New Driver*.
2. Select a 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 *Import a Driver Configuration from the Server (.XML file)*, then select *Notes-IDM3_5-V2.xml*.

   The driver configuration files are installed on the Web server when you install Identity Manager. During the import, you are prompted for the driver's parameters and other information.
4. Click *Next* to specify values for the driver's parameters. See Table 2-1 on page 26 for a list of parameters you can set.
5. Click *Next*.

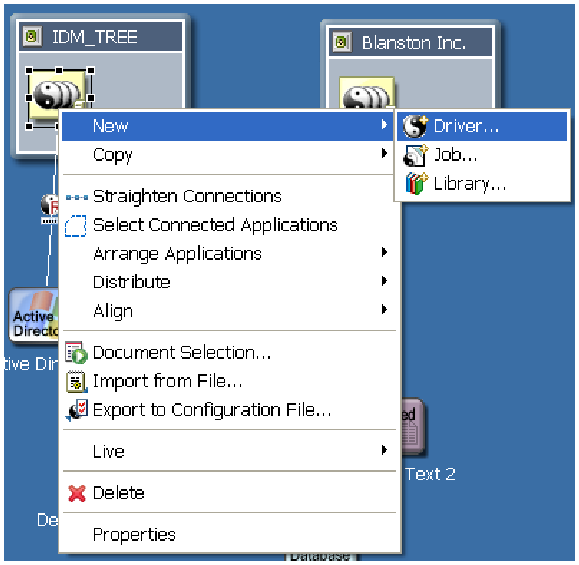
   When the import is finished, you should define security equivalences and exclude administrative roles from replication.

   The driver object must be granted sufficient eDirectory rights to any object it reads or writes. You can do this by granting Security Equivalence to the driver object. The driver must have Read/Write access to users, resources, and distribution lists. Normally, the driver should be given security equal to Admin.
6. Review the driver objects in the *Summary - Current Configuration* page, then click *Finish*.

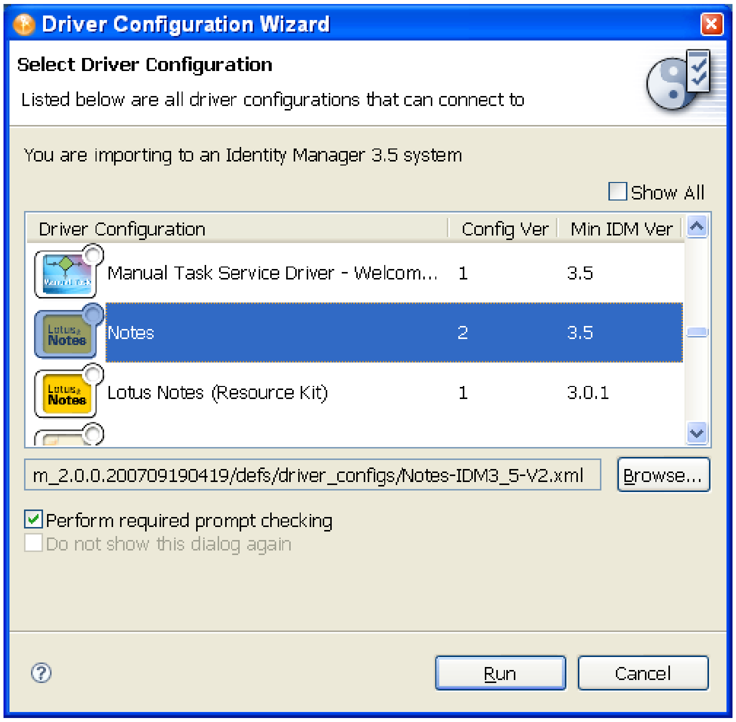
2.6 Installing the Lotus Notes Driver through Designer

Designer has a Driver Configuration Wizard to help you import and configure the Lotus Notes driver configuration file. This file creates and configures the objects and policies needed to make the driver work properly.

1 In Designer, right-click the driver set where you plan to install and configure the Lotus Notes driver, then select *New* > *Driver*.



2 From the Connecting to Application window, select the `Notes-IDM3_5-V2.xml` file. Also select *Perform required prompt checking*, then click *Run*.



Listed below are all driver configurations that can connect to

You are importing to an Identity Manager 3.5 system

☐ Show All

Driver Configuration	Config Ver	Min IDM Ver
Manual Task Service Driver - Welcom...	1	3.5
Notes	2	3.5
Lotus Notes (Resource Kit)	1	3.0.1

m_2.0.0.200709190419/defs/driver_configs/Notes-IDM3_5-V2.xml

☑ Perform required prompt checking

☐ Do not show this dialog again

3 Specify values for the required parameters (see Table 2.1) in the Import Information Requested pages. Then click *Finish*.

***Table 2-1** Setting Up the Lotus Notes Driver Parameters*

Import Prompt	Description
Driver Name	The name of the driver contained in the driver configuration file is “Notes.” Specify the actual name you want to use for the driver.
Enable Entitlements	The Notes driver can be configured to use Entitlements to manage user accounts and group membership in the Lotus Notes Address Book. When using Entitlements, this driver works in conjunction with external services such as the Identity Manager User Application or Role-Based Entitlements to control the conditions under which these features are provisioned or deprovisioned in Lotus Notes. Answer *Yes* only if you plan to use one of these external services to control provisioning to Lotus Notes. Refer to the *Novell Identity Manager 3.5.1 Administration Guide* for details.

Import Prompt	Description
Notes User ID	Specify the Notes User ID this driver will use for Notes Authentication (in fully qualified canonical form, such as cn=Notes Driver/o=Organization). This user ID needs administrative rights to the Input database as well as the Output database. We recommend that this ID be specifically created for the driver and used only by the driver. This prevents the driver from responding to changes made to Notes when this user is used.
Notes User ID File	Specify the full path for the Notes User ID file associated with the Notes User this driver will use for Notes Authentication. The default is `c:\lotus\domino\data\admin.id`.
Notes User Password	Specify the password for the Notes User ID this driver will use when authenticating to Notes (for the above user ID file), then reconfirm the password.
Default Notes Certifier ID File	Specify the full path for the Default Notes Certifier ID file the driver will use as the default certifier. This is usually the root certifier, but can be any certifier with adequate access. The default is `c:\lotus\domino\data\cert.id`.
Notes Fully Qualified Default Certifier Name	Specify the default Fully Qualified (typed) Notes Certifier name as found in the Notes Address Book (`/o=acme`)
Notes Default Certifier Name	Specify the default Notes Certifier name (typeless) as found in the Notes Address Book (`/acme`)
Default Notes Certifier Password	Specify the password for the Default Notes Certifier ID this driver will use when certifying new users, then reconfirm the password. This password is secured using the Named Passwords feature. See Section 4.3, “Using Named Passwords,” on page 52.
Notes Domain	Specify the name of the Notes Domain. The default is *NotesDomain*.
Notes ID Storage Path	Specify the path where the driver should create new user ID files. For example: `/local/notesdata/ids/people` on Linux or `c:\lotus\notes\data\ids\people` on Windows.
Is the Domino Server a North American Server?	Is the Domino server this driver is binding to when certifying new users a North American Domino server? (This affects encryption levels.) Choose *Yes* for 128-bit encryption. The default is *Yes*.
Deny Access Group Name	Specify the Notes Deny Access Group Name. Disabled or deleted Notes users are added to this Notes Group to revoke Notes privileges.
Default Placement Path For Users	Specify the eDirectory path (subtree root) where user changes will be syncrhonized.

Import Prompt	Description
Default Placement Path For Groups	Specify the eDirectory path (subtree root) where group changes will be syncrhonized.
Driver is Remote/Local	Specify if you want this driver to run remotely with the Remote Loader service, or locally. The default is *Remote*. It is recommended that you run the Lotus Notes Driver through the Remote Loader even when running on the same machine as the Identity Manager Engine/Server.
Remote Host Name and Port	Remote driver configuration only. 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.
Driver Password	Remote driver configuration only. The Driver object password is used by the Remote Loader to authenticate itself to the Metadirectory server. It must be the same password that is specified as the Driver object password on the Identity Manager Remote Loader.
Remote Password	Remote driver configuration only. The Remote Loader password is used to control access to the Remote Loader instance. It must be the same password that is specified as the Remote Loader password on the Identity Manager Remote Loader.

2.6.1 After You Import the Driver

1 Check the driver configuration settings to ensure that they are configured correctly for your network system. These settings include:

- Driver parameter setting options
- Subscriber channel parameter setting options
- Publisher channel parameter setting options

You can check these settings in Section C.1.5, “Driver Parameters,” on page 165.

2 Check the driver’s global configuration values (GCVs) to ensure that they are configured correctly for your network system. You can check these settings in Section C.2, “Global Configuration Values,” on page 167.

For example, the Internet Mail Domain parameter defaults to *mycompany.com*, and needs to reflect a correct mail host DNS Name in order to work.

2.7 Setting Up the Driver

Complete these tasks to get the driver installed, configured, and running. (If you are upgrading the driver, see Chapter 3, “Upgrading,” on page 41.)

- Section 2.7.1, “Installing the Driver Shim,” on page 29
- Section 2.5, “Importing the Notes Driver Configuration File in iManager,” on page 24

- Section 2.7.2, “Configuring Publication Synchronization Using Ndsrep,” on page 36
- Section 2.7.3, “Migrating and Resynchronizing Data,” on page 38
- Chapter 5, “Activating the Driver,” on page 105

Most installations require some customization after installation to handle certification. Refer to Chapter 4, “Customizing the Driver,” on page 49 for more information.

2.7.1 Installing the Driver Shim

- “Installing on a Windows Domino Server” on page 29
- “Installing on a Notes Client Workstation Running Windows” on page 31
- “Installing on AIX, Linux, or Solaris” on page 33

Installing on a Windows Domino Server

1 Install the Remote Loader. See Section 2.4.2, “Installing the Lotus Notes Driver on Windows with the Identity Manager Connected System,” on page 22.

To install the driver shim, it is recommended you use the Remote Loader to run the driver, even if the driver is running on the same machine as Identity Manager. See Section 2.4, “Installing the Lotus Notes Driver During Identity Manager Installation,” on page 21.

The necessary files for the driver shim are installed in the `\Novell\RemoteLoader` and `\Novell\RemoteLoader\lib` directories.

2 Manually copy the following files to set up the driver.

Filename	Copy from	Copy to
`ndsrep.exe`	The installed location: `\novell\NDS` or `\Novell\RemoteLoader`	The Domino server executable folder (`\Lotus\Domino`)
`dsrepcfg.ntf`	The installed location: `\novell\NDS` or `\Novell\RemoteLoader`	The Domino server data folder (`Lotus\Domino\Data`)
`Notes.jar`	`\Lotus\Domino`	If running remotely, `\Novell\RemoteLoader\lib` or if running locally, `\Novell\NDS\lib`

3 Make sure that the Domino shared libraries directory (for example, `c:\lotus\domino`) is in the Windows system path, and reboot the computer to make sure this step becomes effective.

Without this directory in the Windows system path, the JVM* might have difficulty locating the Domino shared libraries required by `Notes.jar`, such as `nlsxbe.dll`.

4 If the Domino server requires databases to be signed, use a Notes client or Domino Administrator to sign `dsrepcfg.ntf` with your Domino server’s server ID.

5 After installation, create a driver object as explained in Section 2.5, “Importing the Notes Driver Configuration File in iManager,” on page 24 or Section 2.6, “Installing the Lotus Notes Driver through Designer,” on page 25.

6 Set passwords for the driver and Remote Loader for the initial startup of the Remote Loader.

These passwords must be the same as the Driver Password and Remote Password you specified when importing the driver configuration.

7 Start the driver:

7a In iManager, select *Identity Manager > Identity Manager Overview*.

7b Locate the driver in its driver set by searching the entire tree, or specify a container and click *Search*.

7c Click the driver status indicator in the upper right corner of the driver icon, then click *Start Driver.*

7d Enter the password for the Notes user that you are using for the driver, if you are prompted to do so. This prompt appears only the first time you start the driver, and whether it appears depends on your driver configuration.

When the driver starts for the first time, it does the following:

- Searches for the Domino Server (specified in the driver parameters at import time)
- Opens `dsrepcfg.nsf`. If that file does not exist, the driver automatically creates `dsrepcfg.nsf`, using the `dsrepcfg.ntf` database template that is provided with the driver.
- Writes the Publisher parameters and data to `dsrepcfg.nsf`, specifying an appropriate update database file (usually named `ndsrep.nsf`), so that ndsrep can read them.

IMPORTANT: If multiple `notes.ini` files exist on the machine running the driver, ensure that the NotesDriverShim uses the correct `.ini` by placing its directory in the OS search path. If the driver shim initializes with the wrong `notes.ini` file, the driver shim cannot open `dsrepcfg.ntf`.

If `dsrepcfg.ntf` is not found, or the initial `dsrepcfg.nsf` creation process fails, then the Publisher channel shuts down, and you cannot complete Step 9.

Ensure that the driver shim initializes properly by modifying the Windows system path to find the appropriate `notes.ini` file.

8 At the Domino Console, start the ndsrep task:

```
load ndsrep instance
```

The *instance* must be the driver name, or a unique instance name set up for this driver. If the name of your driver includes spaces, then you must put quotes around the name. After ndsrep is loaded, all `TELL` commands are issued to this instance of ndsrep using the instance name.

A task named `NDSREP-instance` is now displayed in the Notes Task Viewer.

9 After the initial configuration and startup has been validated, update the Domino server's `notes.ini` file so that `ndsrep` is loaded automatically.

For example:

```
ServerTasks=Update,Replica,Router,AMgr,AdminP,ndsrep notesdrv1,
CalConn,Sched,HTTP,IMAP,POP3
```

If the name of your driver includes spaces, then you must put quotes around the name.

After the first successful startup, the Notes driver and ndsrep can be launched in any order that is convenient for your particular configuration.

For more information about ndsrep, see Section 2.7.2, "Configuring Publication Synchronization Using Ndsrep," on page 36.

You can also use Domino Administrator to create a program document that will launch `ndsrep` when the Domino server starts. To do this, see "Automated Method #2" on page 37.

10 Activate the driver, as explained in Chapter 5, "Activating the Driver," on page 105.

Data synchronized by the driver should not be used outside of a test environment if you have not purchased the driver.

11 If you want to synchronize all objects at once, you must initiate the process as explained in Section 2.7.3, "Migrating and Resynchronizing Data," on page 38.

Otherwise, synchronization takes place on an object-by-object basis, the next time a change is made to the individual object.

12 Most installations require some customization after installation to handle certification. Refer to Chapter 4, "Customizing the Driver," on page 49 for more information.

Installing on a Notes Client Workstation Running Windows

1 On a Notes client workstation running Windows, set up a Notes client to connect to the Domino server that hosts the synchronized database.

2 Install the Remote Loader and the Notes driver shim on a Lotus Notes client workstation that is separate from the Domino server. See Section 2.4.2, "Installing the Lotus Notes Driver on Windows with the Identity Manager Connected System," on page 22.

The necessary files for the driver shim are installed in the `\Novell\RemoteLoader` and `\Novell\RemoteLoader\lib` directories.

3 Manually copy the following files to set up the driver.

Filename	Copy from	Copy to
`ndsrep.exe`	The installed location: `\novell\NDS` or `\Novell\RemoteLoader`	The Domino server executable folder (`\Lotus\Domino`)
`dsrepcfg.ntf`	The installed location: `\novell\NDS` or `\Novell\RemoteLoader`	The Domino server data folder (`Lotus\Domino\Data`)
`Notes.jar`	`\Lotus\Notes`	If running remotely, `\Novell\RemoteLoader\lib` or if running locally, `\Novell\NDS\lib`

4 Make sure that the Notes client library `.dll` directory (for example, `c:\lotus\notes`) is in the Windows system path, and reboot the computer to make sure this step becomes effective.

Without this directory in the Windows system path, the JVM might have difficulty locating the Notes client libraries required by `Notes.jar`, such as `nlsxbe.dll`.

5 Make sure the necessary `user.id` and `cert.id` files are available to the Notes Client and `NotesDriverShim.jar` so that the Notes driver can properly authenticate to the Domino server and register new Notes users.

6 After installation, create a driver object as explained in Section 2.5, "Importing the Notes Driver Configuration File in iManager," on page 24 or Section 2.6, "Installing the Lotus Notes Driver through Designer," on page 25.

7 Set passwords for the driver and Remote Loader for the initial startup of the Remote Loader.

These passwords must be the same as the Driver Password and "Remote Password" on page 28 you specified when importing the driver configuration.

8 Start the driver:

8a In iManager, select *Identity Manager > Identity Manager Overview*.

8b Locate the driver in its driver set by searching the entire tree, or specify a container and click *Search*.

8c Click the driver status indicator in the upper-right corner of the driver icon, then click *Start Driver.*

8d Enter the password for the Notes user that you are using for the driver, if you are prompted to do so. This prompt appears only the first time you start the driver, and whether it appears depends on your driver configuration.

When the driver starts for the first time, it does the following:

- Searches for the Domino Server (specified in the driver parameters at import time)
- Opens `dsrepcfg.nsf`. If that file does not exist, the driver creates `dsrepcfg.nsf` automatically, using the `dsrepcfg.ntf` database template that is provided with the driver.
- Writes to `dsrepcfg.nsf` the Publisher parameters and data specifying an appropriate update database file (usually named `ndsrep.nsf`), so that ndsrep can read them.

IMPORTANT: If multiple `notes.ini` files exist on the machine running the driver, ensure that the NotesDriverShim uses the correct `.ini` by placing its directory in the OS search path. If the driver shim initializes with the wrong `notes.ini` file, the driver shim cannot open `dsrepcfg.ntf`.

If `dsrepcfg.ntf` is not found, or the initial `dsrepcfg.nsf` creation process fails, then the Publisher channel shuts down. Ensure that the driver shim initializes properly by modifying the Windows system path to find the appropriate `notes.ini` file.

9 If you choose to run the NotesDriverShim with the Remote Loader on a machine with Windows and a Notes Client, the initial user password (or modified HTTPPassword) is sent across the wire from the NotesDriverShim to the Domino server through a Lotus Notes Client/Domino server communication port.

To secure this connection, use the LotusNotes port communication encryption configured on the Domino server. See Domino server documentation for further information.

10 If the NotesDriverShim runs on the same machine as the Domino server and connects to the Identity Manager services running on a different machine via Remote Loader, you can secure this connection via SSL following the instructions in the IDM Remote Loader documentation. See Setting Up Remote Loaders (http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm/admin/data/bs35pip.html).

11 At the Domino Console, start the ndsrep task:

```
load ndsrep instance
```

The *instance* must be the driver name, or a unique instance name set up for this driver. If the name of your driver includes spaces, then you must put quotes around the name. After ndsrep is loaded, all TELL commands are issued to this instance of ndsrep using the instance name.

A task named NDSREP-instance is now displayed in the Notes Task Viewer.

12 After the initial configuration and startup has been validated, update the Domino server's notes.ini file so that ndsrep is loaded automatically.

For example:

```
ServerTasks=Update,Replica,Router,AMgr,AdminP,ndsrep notesdrv1,
CalConn,Sched,HTTP,IMAP,POP3
```

If the name of your driver includes spaces, then you must put quotes around the name.

After the first successful startup, the Notes driver and ndsrep can be launched in any order that is convenient for your particular configuration.

For more information about ndsrep, see Section 2.7.2, "Configuring Publication Synchronization Using Ndsrep," on page 36.

13 Activate the driver, as explained in Chapter 5, "Activating the Driver," on page 105.

Data synchronized by the driver should not be used outside of a test environment if you have not purchased the driver.

14 If you want to synchronize all objects at once, you must initiate the process as explained in Section 2.7.3, "Migrating and Resynchronizing Data," on page 38.

Otherwise, synchronization takes place on an object-by-object basis, the next time a change is made to the individual object.

15 Most installations require some customization after installation to handle certification. Refer to Chapter 4, "Customizing the Driver," on page 49 for more information.

Installing on AIX, Linux, or Solaris

1 After installing Identity Manager, install the driver shim and the Remote Loader on the connected system where you want to run the driver.

For AIX, Linux, and Solaris, you must use the Remote Loader to run the driver, even if the driver is running on the same machine as Identity Manager.

In the installation, choose Connected System Server, as described in Section 2.4.3, "Installing the Lotus Notes Driver through the GUI Interface On Linux Platforms," on page 23 or Section 2.4.4, "Using the Command Line to Install the Driver on UNIX/Linux," on page 23.

The necessary files for the driver shim are installed in /usr/lib/dirxml.

2 Make sure that the Notes.jar file is linked to the correct directory for your environment.

If your Domino server is prior to V7, enter the following:

```
ls -l /usr/lib/dirxml/classes/Notes.jar
```

The link should be something like the following:

/usr/lib/dirxml/classes/Notes.jar linked to /opt/lotus/notes/latest/linux/Notes.jar

If your Domino server is V7.x running eDirectory 8.8, enter the following:

```
ls -l /opt/novell/eDirectory/lib/dirxml/classes/Notes.jar
```

The link should be something like the following:

`/opt/novell/eDirectory/lib/dirxml/classes/Notes.jar` linked to `/opt/ibm/lotus/notes/latest/linux/jvm/lib/ext/Notes.jar`

3 Make sure you have created a user to run the Remote Loader and the driver, as described in "Creating Lotus Notes Accounts and Groups" on page 20.

   You cannot run Remote Loader for the Notes driver using `root`.

4 Create a driver object as explained in "Importing the Notes Driver Configuration File in iManager" on page 24 or Section 2.6, "Installing the Lotus Notes Driver through Designer," on page 25. Do not start the driver yet.

5 Use a Notes client or Domino Administrator to sign `dsrepcfg.ntf` with your Domino server's server ID.

6 Copy the following files from where they are installed (`/usr/lib/dirxml/rules/notes` or `/opt/novell/eDirectory/lib/dirxml/rules`), to the location where you intend to launch your driver on the Domino server, such as `/local/notesdata`. You might want this location to be in your execution search path. You might also need appropriate rights when copying these files to other directories.

Filename	Description
`rdxml.startnotes`	This script calls the findDomino script, which sets up appropriate Domino operating system environment variables for the Notes driver. Then the `rdxml.startnotes` script launches the Remote Loader with the Notes driver parameters specified in the `rdxml.confignotes` file. If the location where the scripts are placed is not in a current search path, you might need to do one of the following: ◆ Modify `rdxml.startnotes` to include a specific path to the findDomino script. ◆ Create a symbolic link for findDomino in `/usr/bin`.
`rdxml.stopnotes`	This script stops the Remote Loader that is running the Notes driver.
`findDomino`	This script is called from the `rdxml.startnotes` script. When you launch `rdxml.startnotes`, this script sets up operating system environment variables that indicate the location of a UNIX type of installation of Domino.
`rdxml.confignotes` (or wherever your configuration is stored)	This configuration is referenced by `rdxml.startnotes` and `rdxml.stopnotes` scripts. You might need to modify the `rdxml.startnotes` script to fit your environment. For example, if you change the name of the configuration file to a name other than `rdxml.confignotes`, you must revise the last line in the script. You might need to change the configuration ports that are referenced in this file.

These three sample scripts and the sample configuration file are provided to demonstrate how to launch the driver. You can start the Remote Loader for the driver using `rdxml.startnotes`, and stop the Remote Loader for the driver using `rdxml.stopnotes`.

The sample scripts work in a variety of situations. If they do not work in your environment, you might need to edit them appropriately. The sample scripts produce a Remote Loader trace log for the driver that can be used for troubleshooting.

7 Modify the scripts and configuration file to fit to your environment, as described in the table in Step 6.

8 Make sure that the three scripts noted in Step 6 have file access for execution (for example, rwxr-xr-x).

9 Set passwords for the driver and Remote Loader for the initial startup of the Remote Loader.

For example,

```
cd driver_script_directory
      ./rdxml.startnotes -sp driver_password
remote_loader_password
```

These passwords must be the same as the Driver Password and "Remote Password" on page 28 you specified when importing the driver configuration.

10 Use `rdxml.startnotes` to start Remote Loader for the driver.

For example,

```
cd driver_script_directory
      ./rdxml.startnotes
```

The *driver_script_directory* should be the directory where you placed the files in Step 6.

11 Start the driver by using iManager.

11a In iManager, select *Identity Manager* > *Identity Manager Overview*.

11b Locate the driver in its driver set.

11c Click the driver status indicator in the upper right corner of the driver icon, then click *Start Driver*.

When the driver starts the first time, it does the following:

- Searches for the Domino Server (specified in the driver parameters at import time)
- Opens `dsrepcfg.nsf`. If that file does not exist, the driver automatically creates `dsrepcfg.nsf`, using the `dsrepcfg.ntf` database template that is provided with the driver.
- Writes the Publisher parameters and data to `dsrepcfg.nsf,` specifying an appropriate update database file (usually named `ndsrep.nsf`), so that ndsrep can read them.

NOTE: If `dsrepcfg.ntf` is not found, or this initial `dsrepcfg.nsf` creation process fails, then the Publisher channel shuts down, and Step 12 cannot be completed. You must have a valid TCP/IP port configured on the Domino server for this to succeed.

12 At the Domino Console, start the ndsrep task:

```
load ndsrep instance
```

The *instance* must be the driver name, or a unique instance name set up for this driver. If the name of your driver includes spaces, then you must put quotes around the name. After ndsrep is loaded, all TELL commands are issued to this instance of ndsrep using the instance name.

A task named DirXML or a similar name is now displayed in the Notes Task Viewer.

13 After the initial configuration and startup has been validated, update the Domino `notes.ini` file so that ndsrep is loaded automatically.

For example:

```
ServerTasks=Update,Replica,Router,AMgr,AdminP,ndsrep notesdrv1,
CalConn,Sched,HTTP,IMAP,POP3
```

Or you can create a program document (see "Loading Ndsrep" on page 36).

If the name of your driver includes spaces, then you must put quotes around the name.

After the first successful startup, the Notes driver and ndsrep can be launched in any order that is convenient for your particular configuration.

14 Activate the driver, as explained in Chapter 5, "Activating the Driver," on page 105.

Data synchronized by the driver should not be used outside of a test environment if you have not purchased the driver.

15 If you want to synchronize all objects at once, you must initiate the process as explained in Section 2.7.3, "Migrating and Resynchronizing Data," on page 38.

Otherwise, synchronization takes place on an object-by-object basis, the next time a change is made to the individual object.

16 Most installations require some customization after installation to handle certification. Refer to Chapter 4, "Customizing the Driver," on page 49 for more information.

For troubleshooting tips, see "Troubleshooting Installation Problems" on page 136.

2.7.2 Configuring Publication Synchronization Using Ndsrep

Complete the following sections to configure replication using ndsrep:

- "Setting Up Ndsrep" on page 36
- "Loading Ndsrep" on page 36
- "Controlling Ndsrep" on page 37
- "Setting Up Multiple Instances of Ndsrep" on page 38

Setting Up Ndsrep

1 Review the information about ndsrep and starting the driver in the steps in Section 2.7.1, "Installing the Driver Shim," on page 29.

2 Make sure you have copied the necessary files for your platform, as described in Section 2.7.1, "Installing the Driver Shim," on page 29.

3 (Windows only) Add `c:\lotus\domino` (or the appropriate Domino executable folder) to your system path, then reboot the computer.

4 Before trying to load ndsrep, make sure that the Identity Manager Driver for Lotus Notes has been started at least once.

Loading Ndsrep

You always load and run ndsrep at the server console on the Domino server. The ndsrep program creates an output database (by default, `ndsrep.nsf`), detects changes in the Public address book in the Domino server (or other Notes database), and copies these changes to the output database.

You can load ndsrep through two automated methods or though one manual method.

Automated Method #1

1. Load ndsrep in the Domino Server console.
2. Add ndsrep to the ServerTasks = statement in `notes.ini` and restart the Domino server,

   For example:

   ```
   ServerTasks=Update,Replica,Router,AMgr,AdminP,ndsrep notesdrv1,
   CalConn,Sched,HTTP,IMAP,POP3
   ```

Automated Method #2

Create a program document.

Using the Domino Administrator, you can define programs to run on a server using the *Server>Programs* view. Using the Programs view, define the program you want to run, including the server name. Use the following steps to define ndsrep.

1. Using Domino Administrator, select the *Configuration* tab.
2. Expand the *Server* icon in the left pane and select the *Programs* icon under Server.
3. Click the *Add Program* button and fill in the information for ndsrep. For example, if you are trying to set up ndsrep for NotesDrv1 on Domino Server Dom/Serv1/acme, enter the following:
   - Under the Basics section, type in the program name `ndsrep`.
   - In the Command entry, type `NotesDrv1`
   - In the Server to run on entry, type `DomServ1/acme`
   - In the Comments entry, type `NotesDrv1 Publisher polling process`
   - In the Enabled/Disabled entry under the Schedule section, type `At server startup only`
4. Click the *Save & Close* button to save the program definition.

The next time the DomServ1/acme Domino server starts, the nsrep for NotesDrv1 should load automatically.

Manual Method

1. Type the following in the Notes Server Console window:

   ```
   load ndsrep instance
   ```

If the name of your driver includes spaces, then you must put quotes around the name.

Controlling Ndsrep

Use the `TELL` commands described in the table. The following ndsrep `TELL` commands allow for immediate ndsrep actions. These commands are not stored; ndsrep simply executes the action.

TELL Command	Description
`RefreshConfig`	Reads ndsrep configuration information from the configuration store.
`Replicate`	Forces an immediate check for updated notes.

TELL Command	Description
`Resume`	Sets ndsrep to resume processing timer events and replication.
`ShowConfig`	Displays ndsrep configuration settings in the console window.
`ShowFilter`	Displays the first 240 characters of the filter for updated records that ndsrep is using when publishing.
`Suspend`	Suspends activity until the `Resume` command is given.

Setting Up Multiple Instances of Ndsrep

You can run multiple instances of ndsrep to support multiple drivers running against a single Domino server. You must specify the appropriate driver instance name as a parameter when loading ndsrep. By default, this instance name is the name of the driver.

If the name of your driver includes spaces, then you must put quotes around the name.

Consider the following important issues with setting up ndsrep and multiple instances:

- To load ndsrep, you must use the appropriate instance name:

  ```
  load ndsrep instance
  ```

  ndsrep is loaded and referenced using `TELL` commands by the value of *instance*.
- By default, ndsrep stores configuration data for instances in a common Notes database (`dsrepcfg.nsf`).
- When modifying `notes.ini` to auto load multiple instances of ndsrep, simply insert `ndsrep instance` multiple times on the ServerTask line of `notes.ini`.

  For example:

  ```
  ServerTasks=Update,Replica,Router,AMgr,AdminP,
  ndsrep notesdrv1,ndsrep notesdrv2,CalConn,Sched,HTTP,IMAP,POP3
  ```

  You can also create multiple program documents (see "Automated Method #2" on page 37).
- For custom configurations, you can tell ndsrep to utilize a different configuration database. To do so, use the ndsrep configuration parameter and load ndsrep using the `-f filename` parameter as noted in ndsrep configuration database and ndsrep configuration instance in the parameters table in Chapter 4, "Customizing the Driver," on page 49

2.7.3 Migrating and Resynchronizing Data

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

- **Migrate Data from eDirectory:** Allows you to select containers or objects you want to migrate from the Identity Vault to an application. When you migrate an object, the Metadirectory engine applies all of the Matching, Placement, and Create rules, as well as the Subscriber filter, to the object.
- **Migrate Data into eDirectory:** Allows you to define the criteria Identity Manager uses to migrate objects from an application into the Novell Identity Vault. When you migrate an object, the Metadirectory engine applies all of the Matching, Placement, and Create rules, as well as the Publisher filter, to the object. Objects are migrated into the Identity Vault using the order you specify in the Class list.

- **Synchronize:** The Metadirectory engine looks in the Subscriber class filter and processes all objects for those classes. Associated objects will be merged. Unassociated objects are processed as Add events.

To use one of the options explained above:

1. In iManager, select *Identity Manager > Identity Manager Overview*.
2. Locate the driver set containing the Notes driver, then double-click the driver icon.
3. Click the appropriate migration button.

Upgrading

3

3.1 Changes in Policy Architecture

Identity Manager 3.5 and 3.5.1 contain a new policy architecture, which affects how drivers reference policies. Although 3.5.1 driver architecture offers increased functionality in the Identity Manager 3.5.1 environment, the 3.0.x Metadirectory engine cannot run 3.5.1 driver configurations.

However, Identity Manager 3.5 and 3.5.1 can run 3.0x driver configurations. If you have 3.0.x driver configurations associated with both 3.0.x and 3.5.1 Metadirectory engines, do not upgrade the 3.0.x drivers. The 3.0.x driver configurations work in a 3.5.1 environment, but they don't have the increased functionality that Identity Manager 3.5 and above affords. When 3.0.x driver configurations are only associated with a 3.5 or later Metadirectory engine, you should then upgrade the 3.0.x drivers to 3.5.1.

For more information on policy architecture and upgrading drivers to 3.5.1, see "Upgrading Identity Manager Policies" in *Understanding Policies for Identity Manager 3.5.1*.

If you are upgrading from Identity Manager 3.5.0 to Identity Manager 3.5.1, the following information does not apply.

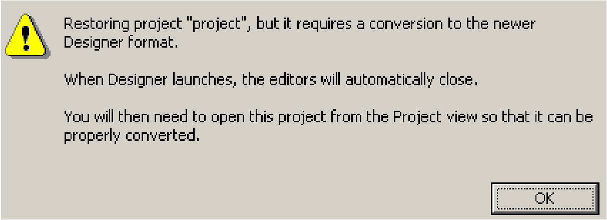
3.2 Upgrading the Driver in Designer

1. Make sure 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 9, "Backing Up the Driver," on page 139 for instruction on how to back up the driver.
3. Install Designer version 2.1 or above, 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, the following warning message is displayed. Read the warning message, then click *OK*.



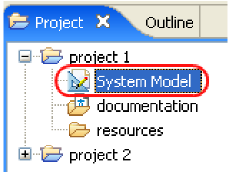
Restoring project "project", but it requires a conversion to the newer Designer format.

When Designer launches, the editors will automatically close.

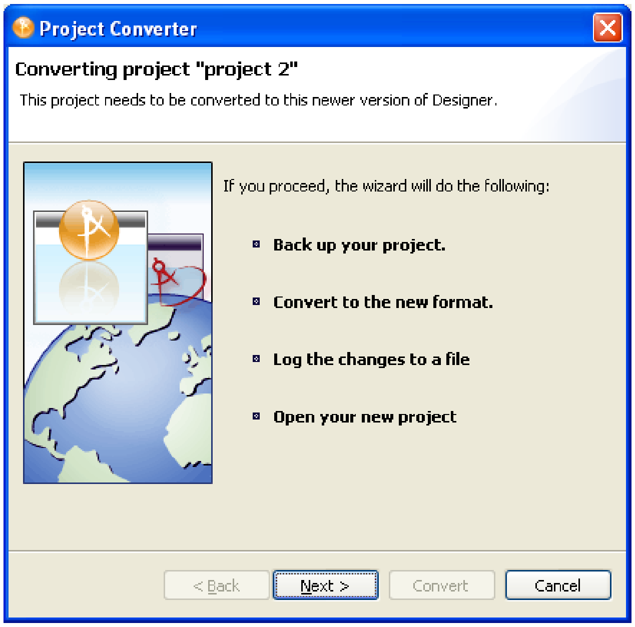
You will then need to open this project from the Project view so that it can be properly converted.

Designer closes the project to preform the upgrade.

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



6 Read the Project Converter message explaining that the project is backed up, converted to the new format, changes logged to a file, and the new project is opened, then click *Next*.



Project Converter

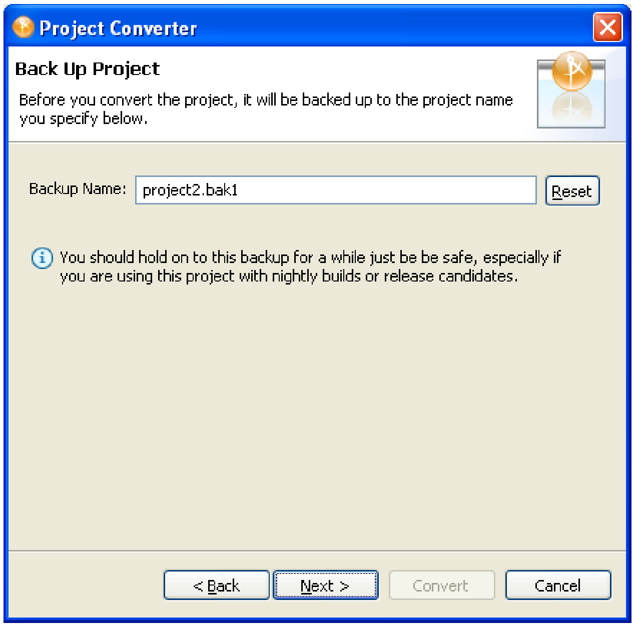
Converting project "project 2"

This project needs to be converted to this newer version of Designer.

If you proceed, the wizard will do the following:

- **Back up your project.**
- **Convert to the new format.**
- **Log the changes to a file**
- **Open your new project**

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



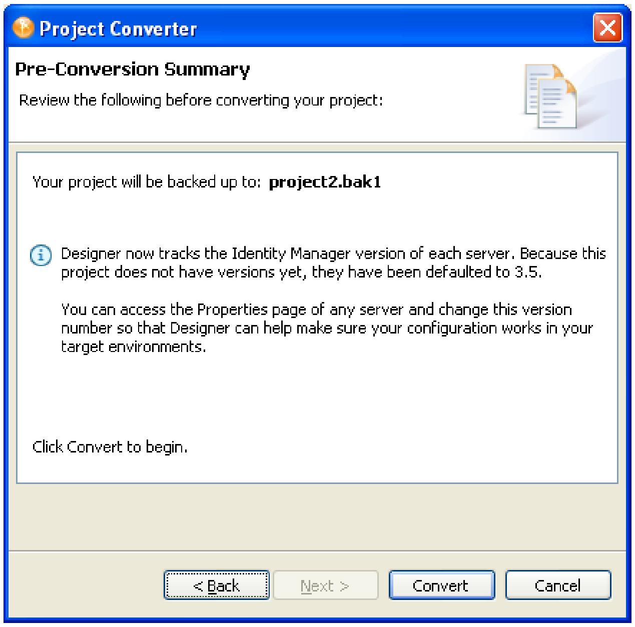
Back Up Project

Before you convert the project, it will be backed up to the project name you specify below.

Backup Name: project2.bak1

You should hold on to this backup for a while just be be safe, especially if you are using this project with nightly builds or release candidates.

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



Pre-Conversion Summary

Review the following before converting your project:

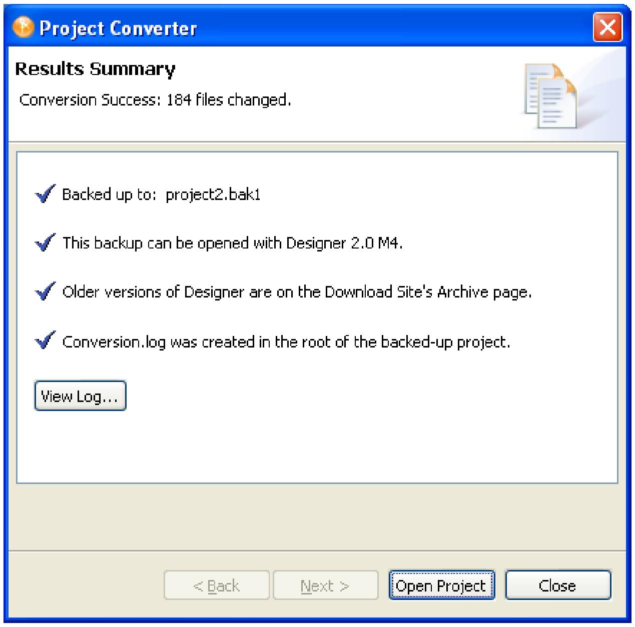
Your project will be backed up to: **project2.bak1**

Designer now tracks the Identity Manager version of each server. Because this project does not have versions yet, they have been defaulted to 3.5.

You can access the Properties page of any server and change this version number so that Designer can help make sure your configuration works in your target environments.

Click Convert to begin.

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



Project Converter

Results Summary

Conversion Success: 184 files changed.

✓ Backed up to: project2.bak1

✓ This backup can be opened with Designer 2.0 M4.

✓ Older versions of Designer are on the Download Site's Archive page.

✓ Conversion.log was created in the root of the backed-up project.

If you want to view the log file that is generated, click *View Log*.

3.3 Upgrading the Driver in iManager

1 Make sure 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.

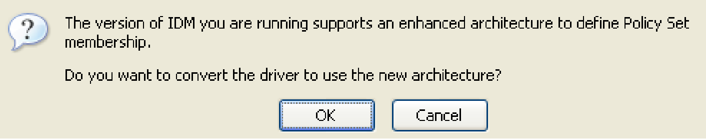
2 Back up the driver. See Chapter 9, "Backing Up the Driver," on page 139 for instruction on how to back up the driver.

3 Verify that Identity Manager 3.6 has be installed and you have the current plug-ins installed, then launch iManager.

4 Click *Identity Manager > Identity Manager Overview*.

5 Click *Search* to find the Driver Set object, then click the driver you want to upgrade.

6 Read the message that is displayed, then click *OK*.



The version of IDM you are running supports an enhanced architecture to define Policy Set membership.

Do you want to convert the driver to use the new architecture?

7 If there is more than one driver to upgrade, repeat Step 2 through Step 6.

3.4 Upgrading on Windows

- Section 3.4.1, "Preparing to Upgrade," on page 45

- Section 3.4.2, “Upgrading the Driver Shim and Configuration from 1.x to Identity Manager 3.5.1,” on page 45
- Section 3.4.3, “Upgrading the Driver Shim and Configuration from 2.x to 3.0,” on page 48

3.4.1 Preparing to Upgrade

The new driver shim is intended to work with your existing driver configuration, but this assumes that your driver shim and configuration have the latest fixes. Make sure you have reviewed all TIDs and product updates for the version of the driver you are using.

3.4.2 Upgrading the Driver Shim and Configuration from 1.*x* to Identity Manager 3.5.1

1 When you install Identity Manager 3.5.1, make sure you select the option to install the utilities along with the Lotus Notes driver. This installs the `movecfg.exe` utility that is necessary for upgrading, noted in Step 5.

You can also download the `movecfg.exe` file from the `Utilities` directory on the `Identity_Manager_3_5_NW_Win.iso,` the `Identity_Manager_3_5_1_Linux.iso`, or the `Identity_Manager_3_5_Unix.iso` image CDs.

2 You can install the upgraded driver shim at the same time you install the Metadirectory engine, or after. To install the driver shim after installation, run the Identity Manager installation program and select the Identity Manager Driver for Notes (called `Notes-IDM3_5-V2.xml)`. Instructions are in Section 2.4.1, “Installing the Lotus Notes Driver on Windows with the Identity Manager Metadirectory Server,” on page 22.

The new driver shim replaces the previous one.

IMPORTANT: Running a new driver with a previous version of the Metadirectory engine is not supported.

3 Convert your existing configuration to 3.5.1 format, using the wizard. See Section 3.2, “Upgrading the Driver in Designer,” on page 41.

4 Unload all instances of ndsrep from the Domino Server Console.

5 Use the `movecfg.exe` utility to upgrade the placement of configuration parameters, as described in Appendix A, “Using the Movecfg.exe Utility,” on page 143.

You can use a batch file such as the example provided in Section A.2, “Example Batch File to Use,” on page 144.

The `movecfg.exe` utility is installed in the `\utilities` directory if you select the option to install Utilities during Identity Manager installation.

For example, on Windows:

```
C:\novell\nds\DirXMLUtilities
```

IMPORTANT: If you have multiple instances of ndsrep, you must run `movecfg.exe` once for each instance, using the -ndsrep parameter.

6 (Windows only) Copy the following files:

- Manually copy `ndsrep.exe` from its installed location (`\novell\NDS`) to the Domino server executable folder (`\Lotus\Domino`).
- Manually copy `dsrepcfg.ntf` from its installed location (`\novell\NDS`) to the Domino server data folder (`\Lotus\Domino\Data`).

  On Linux and Solaris, the package install places it in the `/usr/lib/dirxml/rules/notes` folder and creates a symbolic link for it in the `/local/notesdata` folder.
- Manually copy the `Notes.jar` file from the `\Lotus\Domino` directory to the `\Novell\nds\lib` directory (or the `\novell\remote\loader\lib` directory if running Remote Loader).

This is necessary for product updates as well as new releases.

7 If you have previously modified the Domino server's `notes.ini` file ServerTasks line to auto-load ndsrep (as described in "Loading Ndsrep" on page 36), you must add an instance name (by default, the driver name) as a parameter to ndsrep.

For example:

```
ServerTasks=Router,Replica,Update,Amgr,AdminP,maps,
ndsrep notesdrv1,ndsrep notesdrv2
```

If you have multiple instances of ndsrep, you must do this for each instance. If the name of your driver includes spaces, then you must put quotes around the name.

For example, if the driver name is CN=Notes Driver, your `notes.ini` might look like the following:

```
ServerTasks=Router,Replica,Update,Amgr,AdminP,maps, ndsrep
notesdrv1,ndsrep "Notes Driver"
```

8 Restart ndsrep, or restart the Domino server.

9 Stop and restart eDirectory™ and the driver for the system to use the new driver shim file.

At this point, the driver should work even though you have not made changes to the configuration other than converting it to Identity Manager 3 format.

10 If you want to make changes to the driver configuration, such as using Named Passwords or global configuration values (GCVs) for multiple certifiers, you can do so.

See Chapter 4, "Customizing the Driver," on page 49.

NOTE: For an example of the new parameters and new features such as Named Passwords, review the sample driver configuration.

11 If you are using Lotus Notes 6.0.3, and you want to use the AdminP process features, you need to turn them on by adding the driver parameter named Allow Domino AdminP Support to the Subscriber Options.

For example:

```
<allow-adminp-support display-name="Allow Domino AdminP
Support">True</allow-adminp-support>
```

See Allow Domino AdminP Support in Section 4.4.2, "Subscriber Options," on page 55.

12 Consider adding the sample policy named Notes - Return Email Address (`NotesReturnEmail.xml`) to your driver configuration, in the Command Transformation policy set on the Subscriber channel.

When a new user in the Identity Vault is synchronized to Notes, this policy is used to write back the Notes e-mail address to the Identity Vault. In 1.*x* versions of the driver, this functionality was done differently. If you want to continue to have this functionality with the 3.5 driver version, you must use the new policy.

See "Importing a Policy to Write Back the Notes E-Mail Address for New Users" on page 47.

13 Activate the driver. See Chapter 5, "Activating the Driver," on page 105.

14 When your changes are complete, restart the driver.

Importing a Policy to Write Back the Notes E-Mail Address for New Users

This policy is designed to generate an e-mail address for user Add events on the Subscriber channel. It provides backwards compatibility for functionality that existed in the previous version of the driver. In 1.*x* versions of the driver, this functionality was done differently.

If you want to continue to have this functionality when upgrading a driver configuration to the 3.5.1 driver version, you must use the new policy. (The policy is already a part of the sample configuration provided with the 2.1 version of the driver.)

The default form of the e-mail address provided by the policy is a concatenation of the Given Name, a space, the Surname, and domain name entered when importing the policy. For example: Joe User@mydomain.com. The policy can be edited after import to customize the form of the e-mail address as needed.

1 In iManager, click *Identity Manager Utilities* > *Import Drivers*.

2 Select the driver set where your existing driver resides.

3 In the list of driver configurations that appears, scroll down to the *Additional Policies* heading, then select only the item labeled *Notes - Return Email Address*. Click *Next*.

A list of import prompts appears.

4 Select the name of your existing driver.

5 Specify the domain name to be used as the suffix for the e-mail address generated.

For example, mydomain.com.

6 Click *Next*.

A page appears with the message "A driver named *your_driver_name* already exists in the driver set. Select one of the options below."

7 Select the following items:

- *Update Only Selected Policies in That Driver*
- *Return Email Address (Subscriber - DirXML Script)*

8 Click *Next*, then click *Finish* to complete the wizard.

At this point, the new policy has been created as a policy object under the driver object, but is not yet part of the driver configuration. To link it in, you must manually insert it into a policy set.

9 Insert the new policy into the Command Transformation policy set on the Subscriber Channel.

9a Click *Identity Manager* > *Identity Manager Overview*. Select the driver set for the driver you are updating.

9b Click the driver you just updated. A page opens showing a graphical representation of the driver configuration.

9c Click the icon for the Command Transformation on the Subscriber channel.

9d Click *Insert* to add the new policy. In the Insert page that appears, click *Use an Existing Policy*, then browse for and select the new policy object. Click *OK*.

9e If you have more than one policy in the policy set, use the arrow buttons ▲ ▼ to move the new policy to the correct location in the list.

3.4.3 Upgrading the Driver Shim and Configuration from 2.x to 3.0

1. Stop the driver.
2. Install the new driver shim.
3. If you are using Lotus Notes 6.0.3 or later, and you want to use the AdminP process features, you need to turn them on by adding the driver parameter named Allow Domino AdminP Support to the Subscriber Options.

   For example:

   ```
   <allow-adminp-support display-name=”Allow Domino AdminP
   Support”>True</allow-adminp-support>
   ```

   See Allow Domino AdminP Support in Section 4.4.2, “Subscriber Options,” on page 55.
4. Consider adding the Publisher Options driver parameter named NDSREP Console Trace Level to your driver configuration. See NDSREP Console Trace Level in Section 4.4.3, “Publisher Options,” on page 61.
5. See Chapter 4, “Customizing the Driver,” on page 49 for other new driver configuration parameters that might be useful for your driver configuration.
6. When your changes are complete, restart the driver.

3.5 Upgrading on AIX, Linux, or Solaris

For AIX, Linux, and Solaris, if you upgrade Domino after installing the driver, you need to do one of the following:

- Check symbolic links, and re-create them manually if necessary.
- If you have customized the files `rdxml.startnotes`, `rdxml.stopnotes`, `findDomino`, or `rdxml.confignotes`, back them up and then reinstall the driver. Reinstalling the driver shim re-creates the symbolic links, but it overwrites those files.

For more information, see “Troubleshooting Installation Problems” on page 136.

Customizing the Driver

4

This section explains how to customize your driver for your specific business rules.

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.

4.1 Determining eDirectory Object Placement When a Notes Object is Moved

A Move is done by Identity Manager relative to either a parent's association key or dest-dn. Containment in Notes is purely logical, and as such, an OU in Notes never has an association to the Identity Vault, so it isn't possible to provide a parent association. Also, the driver shim has no reference of the eDirectory™ namespace or containment, so it can't provide a parent dest-dn (destination DN). Therefore, an appropriate parent dest-dn must be provided by a policy.

Notes - Move Sample is a sample Publisher channel policy that contains logic to determine eDirectory object placement when an associated Notes object is moved.

This policy is designed to provide the same functionality contained in the sample style sheet named `placemove.xsl`, provided with earlier versions of the driver. A sample configuration is also available with in `NotesConfig21.xml`, which demonstrates the driver object's `move` and `rename` capabilities.

On a move, the `dest-dn` is set for a particular source dn. After importing the Notes - Move Sample policy, you have a policy defining a single mapping between source and destination containers. You can define additional mappings by editing the resulting policy.

NOTE: Because of the way Notes manages CN and DN in FullName, it is not possible to distinguish between a Move and a Rename event in ndsrep. Therefore, when ndsrep determines that the FullName item has changed, it generates both a Move and a Rename event.

To add the Notes - Move Sample policy to your driver configuration:

1 In iManager, click *Identity Manager Utilities > Import Drivers*.

2 Select the driver set where your existing driver resides.

3 In the list of driver configurations that appears, scroll down to the Additional Policies heading, then select only the item labeled *Notes - Move Sample*. Click *Next*.

   A list of import prompts appears.

4 Select the name of your existing Notes driver.

5 Specify one container in Notes and the corresponding container in the Identity Vault.

   The import process uses this information to create one pair of "mappings" between Notes containers and eDirectory containers.

   5a Specify the source container from Notes where the move originates.

      For example, \MyOrganization\Engineering\Testing.

   5b Browse for and select the destination container where the object should be moved to.

      For example, Testing.MyOrganization.

6 Click *Next*.

   A page appears with the message "A driver named *your_driver_name* already exists in the driver set. Select one of the options below."

7 Select the following:

   - *Update Only Selected Policies in That Driver*
   - *Move Sample (Subscriber - DirXML Script)*

8 Click *Next*, then click *Finish* to complete the wizard.

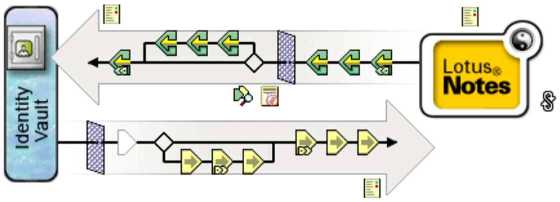
   At this point, the new policy has been created as a policy object under the driver object, but is not yet part of the driver configuration. To link it in, you must manually insert it into a policy set.

9 Insert the new policy in a policy set on the Publisher Channel.

   Place it where it would be appropriate in your driver configuration. For example, in the Input Transformation or Event Transformation policy set.

   9a Click *Identity Manager > Identity Manager Overview*. Select the driver set for the driver you are updating.

   9b Click the driver you just updated. A page opens showing a graphical representation of the driver configuration.



   9c Click the icon for the policy set on the Publisher Channel.

   9d Click the *Plus* button to add the new policy. In the Create Policy page, click *Make a copy of an existing policy*, then browse for and select the new policy object. Click *OK*.

9e If you have more than one policy in the policy set, use the arrow buttons ▲ ▼ to move the new policy to the correct location in the list.

10 Complete the “mappings” for all the containers in Notes and eDirectory by editing the XML for the policy.

Follow the example of the first pair that is created for you with the container names you provided in Step 5.

4.2 Automatically Determining Which Certifier to Use

Because most Notes environments use more than one certifier, you can use a policy to configure the NotesDriverShim to use different certifiers. The sample `Cert.xsl` style sheet, located in the `dirxml\drivers\lotusNotes\rules` directory on the `Identity_Manager_3_5_1_NW_Win.iso` CD, is an Output Transformation style sheet that contains logic to determine which Notes Certifier to use based on the src-dn attribute on the `<add>` XML attribute. Another example provided is the `NotesCertifierSelectionSampleSS.xsl` sample style sheet and in the file `NotesConfig21.xml`. (These are installed if you install from the `Identity_Manager_3_5_1_Unix.iso` or the `Identity_Manager_3_5_1_Linix.iso` CD.)

You can edit the choose/when statements to model your Notes system certifier structure. If using only the root certifier is acceptable, then using `Cert.xsl` is not necessary, because the driver parameters screen can contain the information for the root certifier.

To use `Cert.xsl` in your environment, first, change the existing xsl:when statements to match your configuration.

```
<xsl:when test="string($dn) = '\dirxml-ds\provo\notes\eng'">
  <xsl:attribute name="cert-id">c:\lotus\domino\data\eng.id</
xsl:attribute>
  <xsl:attribute name="cert-pwd">certify2eng</xsl:attribute>
  <xsl:attribute name="user-pwd">new2notes</xsl:attribute
</xsl:when>
```

Add as many xsl:when statements as you need to model your organization’s certification structure.

Then change the cert-id and cert-pwd in xsl:otherwise to match your root certifier information.

```
<xsl:otherwise>
  <xsl:attribute name="cert-id">d:\lotus\domino\data\cert.id</
xsl:attribute>
  <xsl:attribute name="cert-pwd">certify2notes</xsl:attribute>
</xsl:otherwise>
```

`Cert.xsl` communicates the certifier information by adding attributes to the add tag in the XML document. If NotesDriverShim doesn’t find these attributes, it uses the root certifier information from the driver Parameters passed during initialization.

NOTE: `Cert.xsl` also shows how to override several other parameters for the driver. See Section 4.5, “Custom Driver Parameters,” on page 64 for more information about these parameters.

4.3 Using Named Passwords

The Metadirectory engine provided with Identity Manager 2 added a new way of securing the passwords you need to use in your driver policies. The sample driver configuration shows an example and also with Identity Manager 3.x.

One use for this feature would be to store a password for each of your Notes certifiers. For example, if you had certifiers for Human Resources, Engineering, and Marketing, you could use Named Passwords to securely store the password for each respective certifier ID file in your driver parameters. In the driver configuration, you would click the *Edit XML* button and specify driver parameters something like this:

```
<cert-id-password display-name="Certifier Password" is-
sensitive="true" type="password-ref">HR</cert-id-password>
<cert-id-password display-name="Certifier Password" is-
sensitive="true" type="password-ref">Engineering</cert-id-password>
<cert-id-password display-name="Certifier Password" is-
sensitive="true" type="password-ref">Marketing</cert-id-password>
```

When you return to the graphical interface for the driver parameters, each of these passwords has prompts to enter the password and confirm the password. These passwords are encrypted and stored with the driver configuration. You can reference these passwords by name in your driver policies.

For an example of how to use Named Passwords, see the sample configuration and also the `NotesCertifierSelectionSampleSS.xsl` sample style sheet and in the file `NotesConfig21.xml`, listed in Section 4.6, "Example Files," on page 97.

4.4 Using Driver Parameters

To change driver parameters, edit the Driver Parameters page.

1. In iManager, click *Identity Manager* > *Identity Manager Overview*.
2. Find the driver in its driver set.
3. Click the driver icon to display the Driver Overview page.
4. Click the driver icon again to display the Modify Object page.
5. Click *Driver Configuration*.
6. Use the information in the sections that follow to upgrade driver parameters.

- Section 4.4.1, "Driver Options," on page 52
- Section 4.4.2, "Subscriber Options," on page 55
- Section 4.4.3, "Publisher Options," on page 61

4.4.1 Driver Options

The first column of the following table contains the parameter name followed by its common description. The parameters are alphabetized according to the XML parameter name.

The third column of the following table contains XML text that you can paste into the Driver Parameters XML editor. The XML text represents exactly what is necessary to display the parameters. You can also place the information that you see under the Description heading within the `<description>` `</description>` parameters.

Table 4-1 Driver Parameters

Parameter Name and Common Description	Description	XML to Define Driver Parameters

allow-document-locking
Allow Document Locking
	Enables Notes database documents to be locked by the Notes Driver shim if they are being modified. This parameter is only valid in Lotus Notes 6.5 or higher, and if the Notes database has the *Allow document locking* checkbox selected.	`<definition display-name="Allow Document Locking?" name="allow-document-locking" type="boolean"> <description> </description>`
		`<value>true</value> </definition>`

cert-id-file
Default Certifier ID file
	The default Notes Certifier ID file that is used to register user objects in the Notes Address Book. The full path of the file should be represented with respect to the operating system hosting Domino.	`<definition display-name="Default Certifier ID File" name="cert-id-file" type="string"> <description> </description>`
		`<value>c:\lotus654\domino\data\ids\people\ndriver.id</value> </definition>`

cert-id-password
Default Certifier ID password
	The default Notes Certifier ID file password that is used to register user objects in the Notes Address Book.
	When using the `type="password-ref"` attribute of this parameter, the password is encrypted and securely stored with the Driver Configuration. When securely stored in this fashion, the password can then be referenced by the Metadirectory engine or a driver using the key name specified. (In this example, defaultCertPwd.)
		`<definition display-name="Default Certifier Password" is-sensitive="true" name="cert-id-password" type="password-ref"> <description> </description>`
		`<value>defaultCertPwd</value> </definition>`

directory-file
Directory File or Input Database
	The filename of the database to be synchronized with the Identity Vault. Specify this item without full path information.	`<definition display-name="Directory File" name="directory-file" type="string"> <description> </description>`
		`<value>names.nsf</value> </definition>`

is-directory
Notes Address Book
	Specify *True* if the input database (directory file) is a Notes Address book; otherwise, specify *False*.	`<definition display-name="Notes Address Book?" name="is-directory" type="boolean"> <description> </description>`
		`<value>true</value> </definition>`

Parameter Name and Common Description	Description	XML to Define Driver Parameters

janitor-cleanup-interval
Janitor Cleanup Interval
	Janitor cleanup checks for and releases resources that might have been orphaned by unfinished query-ex sequences. This interval determines how often to perform this service.	`<definition display-name="Janitor Cleanup Interval (in minutes)" name="janitor-cleanup-interval" type="integer"> <description> </description>`
		`<value>30</value> </definition>`

notes-domain
Notes Domain Name
	The name of the Notes domain the driver is running against. It might be different from the Notes Organization name, and therefore can't be derived from the server name.	`<definition display-name="Notes Domain Name" name="notes-domain" type="string"> <description> </description>`
		`<value>PROVO1</value> </definition>`

server-id-file
Server ID File (deprecated)
	The Notes Server ID file associated with the Notes Server this driver authenticates to (This is optional). The full path of the file should be represented with respect to the operating system hosting Domino. This ID file need not be the server ID file. It can actually be an ID file that has no password (and need not have any access anywhere).	`<definition display-name="Domino Server ID File" name="server-id-file" type="string"> <description> </description>`
		`<value>c:\lotus654\domino\data\server.id</value> </definition>`

update-file
Update File or ndsrep polling cache
	The filename of the database used to cache database changes that need to be published to the Identity Vault. The default is `ndsrep.nsf`. Specify this item without full path information.
	The Driver's Domino add-in process ndsrep creates this database. Within this database, filtered updates are cached before being consumed by the Notes Driver's publisher.
		`<definition display-name="Update File" name="update-file" type="string"> <description> </description>`
		`<value>ndsrep.nsf</value> </definition>`

user-id-file
Notes User ID file
	The Notes User ID file associated with the Notes User this driver represents (this is required). The full path of the file should be represented with respect to the operating system hosting Domino. The password associated with this user ID file is input in the following user interface section: *Driver Configuration > Authentication > Specify the application password*.	`<definition display-name="Notes Driver User ID File" name="user-id-file" type="string"> <description> </description>`
		`<value>c:\lotus654\domino\data\ids\people\ndriver.id</value> </definition>`

4.4.2 Subscriber Options

The first column of the following table contains the parameter name followed by its common description. The parameters are alphabetized according to the XML parameter name.

The third column of the following table contains XML text that you can paste into the Driver Parameters XML Editor. The XML text represents exactly what is necessary to display the parameters. You can also place the information that you see under the Description heading within the `<description>` `</description>` parameters.

NOTE: Some parameters have values that are listed in second (Description) column. These parameters contain values such as Success, Warning, Error, Retry, or Fatal. The proper XML text for such parameters can be found in the file drvparamdefs.xml (../samples/drvparamdefs.xml).

Table 4-2 *Subscriber Channel Parameters*

Parameter Name and Common Description	Description	XML to Define Driver Parameters

allow-adminp-support
Allow Domino AdminP Support
	Specifies that AdminP features can be used. AdminP features are supported only for users of Lotus Notes 6.0.3 or later.
	If you have Lotus Notes 6.0.3 or later and you want to use the AdminP features, you must add this parameter and set it to True.
	If the parameter does not exist in the driver parameters, the default setting is False.
	This parameter can be overridden on a command-by-command basis by using the attribute Allow AdminP Support described in Table 4-4 on page 65.
		`<definition display-name="Allow Domino AdminP Support" name="allow-adminp-support" type="boolean"> <description> </description>`
		`<value>true</value> </definition>`

allow-http-password-set
Allow Notes Web (HTTP) Password Set
	Set the parameter to True to allow the Notes driver to set or to change the Web (HTTP) password attribute on user objects. Set the parameter to False to disallow the Notes driver from setting or changing the web (HTTP) password attribute on user objects. The default setting is True.	`<definition display-name="Allow Notes Web (HTTP) Password Set" name="allow-http-password-set" type="boolean"> <description> </description>`
		`<value>true</value> </definition>`

Parameter Name and Common Description	Description	XML to Define Driver Parameters
allow-userid-password-set Allow user.id Password Set	Specifies if the NotesDriverShim should modify the password of user.id files. This parameter can be overridden by <allow-userid-password-set> as an attribute to the <modify-password> element.	<definition display-name="Allow user.id password set" name="allow-userid-password-set" type="boolean"> <description> </description> <value>true</value> </definition>
cert-log Registration/ Certification Log File	The Notes Certification log file that is used to record the registration of user objects in the Notes Address Book. Specify this item without full path information.	<definition display-name="Registration/ Certification Log File" name="cert-log" type="string"> <description> </description> <value>certlog.nsf</value> </definition>
cert-path ID File Storage Location	This parameter specifies the default Notes User ID file (certifier) storage location that is used when user objects are registered and ID files are created. New ID files are placed in this location. The full path of the folder should be represented in relationship to the operating system hosting Domino. You can override this default setting by using the XML tag attribute user-id-path.	<definition display-name="User ID File Storage Location" name="cert-path" type="string"> <description> </description> <value>c:\lotus654\domino\data \ids\people</value> </definition>
cert-users Certify/Register Users	This parameter indicates the default behavior for the driver regarding Notes user account creation. Yes indicates the driver by default attempts to register users in the Notes Address book by certifying them and creating an ID file for each user when add events are received. You can override this default setting by using the XML certify-user attribute.	<definition display-name="Certify (register) Notes Users" name="cert-users" type="boolean"> <description> </description> <value>true</value> </definition>
create-mail Create Mail DB	This parameter indicates the default behavior for the driver regarding e-mail account creation. True indicates the driver by default attempts to create a Notes Mail database (mailfile) when adding a new user. You can override this default setting by using the XML create-mail attribute.	<definition display-name="Create User E-Mail Box" name="create-mail" tmpId="238" type="boolean"> <description> </description> <value>true</value> </definition>

Parameter Name and Common Description	Description	XML to Define Driver Parameters

db-creation-max-collisions
Number of File Creation Collision Retry Attempts
	Specify a positive integer value indicating the highest number to append to a filename when attempting to resolve filename collisions. If the NotesDriverShim cannot create a mailfile or a mailfile replica because of a filename collision, the NotesDriverShim appends an integer text value to the end of the attempted filename and tries again to create the file. Thus, if the mailfile `JohnDoe.nsf` already exists, then the NotesDriverShim attempts to create `JohnDoe1.nsf`. If this value is 0, then this file creation after filename collision feature is not invoked.	`<definition display-name="Number of File Creation Collision Retry Attempts" name="db-creation-max-collisions" type="integer">`
		`<value>`5`</value>` </`definition>`

default-http-password
Default HTTP Password
	The default Notes Web (HTTP) password set for newly created Notes users.
	You can override this default setting by using the XML `user-pwd attribute tag`.
		`<definition display-name="Default HTTP Password" name="default-http-password" type="string">` `<description>` </`description>`
		`<value>`notes`</value>` </`definition>`

default-password
Default Notes Password
	The default Notes User ID password for newly created Notes users.
	You can override this default setting by using the XML `user-pwd` attribute.
		`<definition display-name="Default Notes Password" name="default-password" type="string">` `<description>` </`description>`
		`<value>`notes`</value>` </`definition>`

expiration-term
Expiration Term
	The default expiration term (specified in years) for newly created Notes User ID files.
	You can override this default setting by using the XML attribute `expire-term`.
		`<definition display-name="Default User ID File/Registration Expiration Term (in years)" name="expiration-term" type="integer">` `<description>` </`description>`
		`<value>`2`</value>` </`definition>`

Parameter Name and Common Description	Description	XML to Define Driver Parameters

internet-mail-domain
Internet Mail Domain Name
	Obsolete in version 2.0.	`<definition display-name="Internet Mail Domain" name="internet-mail-domain" type="string"> <description> </description>`
		`<value></value> </definition>`

`mailfile-acl-level`
Add User E-Mail ACL Level
	The default ACL setting for the newly created mail file of newly created user objects. Valid values are NOACCESS, DEPOSITOR, READER, AUTHOR, EDITOR, DESIGNER, and MANAGER. When no ACL setting is specified, the setting defaults to MANAGER.
	You can override this default setting by using the XML attribute `mailfile-acl-level`.
		`<definition display-name="Add User E-Mail: E-Mail Database ACL Setting" name="mailfile-acl-level" type="enum"> <description> </description>`
		`<value>`default`</value> </definition>`
		See Note at the top of Subscriber Options (page 55), then use the values under the Description heading for this parameter.

`mailfile-path` User Mail File Storage Location	A mail storage path relative to the Domino data storage location where mail files are stored if created by the driver. For example, if the parameter is set to "mail," then new mail files created by the driver on the Domino server (running on Linux) are stored in the `/local/notesdata/mail` folder.	`<definition display-name="User Mail File Storage Location" name="mailfile-path" type="string"> <description> </description>` <value>mail</value> </definition>

mailfile-template
E-Mail File Template
	The `.ntf` database template to be used when creating a new mail database when the driver creates a user e-mail account. This template must be accessible to the Domino server in the Domino data folder.	`<definition display-name="Mail File Template" name="mailfile-template" type="string"> <description> </description>`
		`<value>`mail654.ntf`</value> </definition>`

mail-server
Domino Mail Server Name
	The DN of the Domino Server that holds the mail files.
	You can override this default setting by using the XML `<MailServer>` element as a child of the `add event` element, or the `mail-server` XML attribute tag.
		`<definition display-name="Domino Mail Server Name" name="mail-server" type="string"> <description> </description>`
		`<value>`CN=blackcap/O=novell`</value> </definition>`

Parameter Name and Common Description	Description	XML to Define Driver Parameters

minimum-pwd-len
Notes Password Strength
	The default minimum password strength (0-16) for newly created Notes User ID files.
	You can override this default setting by using the XML attribute tag `minimum-pwd-len`.
		`<definition display-name="Notes Password Strength (0 - 16)" name="minimum-pwd-len" type="integer"> <description> </description>`
		`<value>2</value> </definition>`

north-american-flag
Is Domino Server North American?
	North American Server User ID file (certifier) property. Set to True only if the Domino Server is in North America. According to Domino registration requirements, this attribute is required for user ID file creation.	`<definition display-name="Is Domino Server North American?" name="north-american-flag" type="boolean"> <description> </description>`
		`<value>true</value> </definition>`

notes-doc-lock-fail-action
Notes Document Locking Failure Action
	Specify the action (document return code) the Notes Driver returns to the Metadirectory engine if the Notes Driver fails to acquire a document lock. The value choices are Retry (default), Warning, Error, Fatal, and Success.
	This parameter is overwritten by `retry-status-return` and is only valid if the `allow-document-locking` is set to True.
		`<definition display-name="Document Lock Failure Action" name="notes-doc-lock-fail-action" type="enum">`
		`<value>retry</value> </definition>`
		See Note at the top of Subscriber Options (page 55).

notes-save-fail-action
Notes Document Save Failure Return Code
	Specify the action (document return code) the Notes Driver returns to the Metadirectory engine if the Notes Driver fails to properly save a document in the synchronized Notes database. If the parameter does not exist in the driver parameters, the default value is Retry.
	Possible values are Success, Warning, Error, Retry, or Fatal.
	You can use this parameter when troubleshooting and is overwritten by `retry-status-return` attribute.
		`<definition display-name="Notes Document Save Failure Return Code" name="notes-save-fail-action" type="enum"> <description> </description>`
		`<value>warning</value> </definition>`
		See Note at the top of Subscriber Options (page 55).

Parameter Name and Common Description	Description	XML to Define Driver Parameters

notes-v1-schema-format
Use NotesDriver v1 Schema Format
	Specifies whether the NotesDriver should publish Notes schema documents in its original v1 format.	`<definition display-name="Use NotesDriver v1 schema format" name="notes-v1-schema-format" type="boolean"> <description> </description>`
		`<value>`true`</value>` </`definition>`

retry-status-return Failed Command Reply Status	Specify the action (document return code) the Notes Driver returns to the Metadirectory engine if the Notes Driver cannot process a command due to connectivity or application processing failure. If the parameter does not exist in the driver parameters, the default setting is Retry. Possible values are Success, Warning, Error, Retry, or Fatal. This parameter can be used when troubleshooting critical situations, and takes precedence over `notes-save-fail action` and `notes-doc-lock-fail-action`.	`<definition display-name="Retry Status Return Code" name="retry-status-return" type="enum">` `<enum-choice display-name="Retry"> retry <enum-choice>`<`description`> </`description`> `<value>`retry`</value>` </`definition`> See Note at the top of Subscriber Options (page 55).
store-id-ab-flag Store User ID in Address Book	This flag indicates the default behavior for the driver for attaching user ID files on their respective user objects in the Notes Address Book at registration time. Setting the flag to True causes registered user objects in the Notes Address Book to be created with an attached user ID file. Setting the flag to False causes registered user objects in the Notes Address Book to be created without an attached user ID file. You can override this default setting by using the XML attribute `store-useridfile-in-ab`.	`<definition display-name="Store User ID File in Address Book" name="store-id-ab-flag" type="boolean">` <`description`> </`description`> `<value>`true`</value>` </`definition`>

Parameter Name and Common Description	Description	XML to Define Driver Parameters

update-ab-flag
Add Registered Users To Address Book
	This parameter indicates the default behavior for the driver for placing registered user objects in the Notes Address Book. Setting the flag to True causes registered users to be placed in the address book. Setting the flag to False causes users to be registered (meaning that a certifier ID file is created for the user) without the user object being placed into the Notes Address Book.
	You can override this default setting by using the XML attribute `update-addressbook`.
		`<definition display-name="Add Registered Users to Address Book" name="update-ab-flag" type="boolean"> <description> </description>`
		`<value>true</value> </definition>`

4.4.3 Publisher Options

The first column of the following table contains the parameter name followed by its common description. The parameters are alphabetized according to the XML parameter name.

The third column of the following table contains XML text that you can paste into the Driver Parameters XML Editor. The XML text represents exactly what is necessary to display the parameters. You can also place the information that you see under the Description heading within the `<description>` `</description>` parameters.

NOTE: Some parameters have values that are listed in second (Description) column. These parameters contain values such as Success, Warning, Error, Retry, or Fatal. The proper XML text for such parameters can be found in the file drvparamdefs.xml (../samples/drvparamdefs.xml).

Table 4-3 *Publisher Channel Parameters*

Parameter Name and Common Description	Description	XML to Define Driver Parameters

check-attrs-flag
Check Attributes
	The ndsrep check and publish attributes parameter. Set to True if only modified attributes within the Publisher filter should be sent to the Identity Vault via the Publisher channel when a Notes object is modified. Set to False if all sync attributes specified within the Publisher filter should be sent to the Identity Vault via the Publisher channel when a Notes object is modified.
	The default value is True.
		`<definition display-name="Check Attributes?" name="check-attrs-flag" type="boolean"> <description> </description>`
		`<value>true</value> </definition>`

Parameter Name and Common Description	Description	XML to Define Driver Parameters

config-db-name
NDSREP Configuration Database
	The ndsrep configuration database filename created and maintained by the driver. This parameter controls which `.nsf` database the driver shim uses to write its publication options.
	The full path of the filename should be represented with respect to the operating system hosting Domino. When using this parameter, ndsrep needs to be loaded with the -f filename parameter.
	ndsrep load example:
	`load ndsrep NotesDriver2 -f /home/notes/mycfg.nsf`
	If this parameter is not present, by default the Configuration database filename is set to `dsrepcfg.nsf` and is normally located in the Domino data folder.
	If the name of your driver includes spaces, then you must put quotes around the name.
		`<definition display-name="NDSREP Configuration database" name="config-db-name" type="string"> <description> </description>`
		`<value>mycfg.nsf</value> </definition>`

dn-format
DN Format
	The Distinguished Name format used by ndsrep. Valid values are NOTES_TYPED, NOTES, SLASH_TYPED, SLASH, LDAP, LDAP_TYPED, DOT, and DOT_TYPED. The default is NOTES_TYPED.	`<definition display-name="DN FORMAT" name="dn-format" type="enum"> <description> </description>`
		`<value>NOTES_TYPED</value> </definition>`

instance-id
NDSREP Configuration Instance
	The ndsrep configuration instance name created and maintained by the driver within the ndsrep configuration database. This parameter controls which database note the driver shim uses to read and write its publication options within the ndsrep configuration database. When using this parameter, ndsrep utilizes the settings of this configuration instance when loaded with this instance name as a parameter.
	If this parameter is not present, by default the configuration instance is set to the name of the driver (the driver RDN in eDirectory.)
	ndsrep load example:
	`load ndsrep NotesDriver2`
	If the name of your driver includes spaces, then you must put quotes around the name.
		`<definition display-name="NDSREP Configuration Instance" name="instance-id" type="string"> <description> </description>`
		`<value>NotesDriver2</value> </definition>`

Parameter Name and Common Description	Description	XML to Define Driver Parameters

loop-detect-flag
Enable Loop Back Detection
	Loopback detection parameter. Set to True to enable loopback detection. Set to False to disable loopback detection.	`<definition display-name="Enable Loop Back Detection" name="loop-detect-flag" type="boolean"> <description> </description>`
		`<value>`true`</value>` </`definition>`

ndsrep-console-trace-level
NDSREP Console Trace Level
	Possible values are SILENT, NORMAL, VERBOSE, or DEBUG.
	If this parameter is not present, the default setting is NORMAL.
		`<definition display-name="NDSREP Domino Console Trace Level" name="ndsrep-console-trace-level" type="enum"> <description> </description>`
		`<value>`NORMAL`</value>` </`definition>`
		See Note at the top of Publisher Options (page 61), then use the values under the Description heading for this parameter.

schedule-units
NDSREP Schedule Units
	The ndsrep polling interval unit. Valid values are SECONDS, MINUTES, HOURS, DAYS, and YEARS. The default value is SECONDS.	`<definition display-name="NDSREP Polling Units" name="schedule-units" type="enum"> <description> </description>`
		`<value>`SECONDS`</value>` </`definition>`
		See Note at the top of Publisher Options (page 61), then use the values under the Description heading for this parameter.

schedule-value
NDSREP Schedule Value
	The ndsrep polling interval unit value. This value is utilized in conjunction with the `<schedule-units>` configuration parameter.	`<definition display-name="NDSREP Polling interval" name="schedule-value" type="integer"> <description> </description>`
		`<value>`30`</value>` </`definition>`

Parameter Name and Common Description	Description	XML to Define Driver Parameters
polling-interval Polling Interval	Notes Driver Shim publisher polling interval, specified in SECONDS, MINUTES, HOURS, and DAYS,.	<definition display-name="Polling Interval (in seconds)" name="polling-interval" type="integer"> <description> </description> <value>30</value> </definition>
pub-heartbeat-interval Publication Heartbeat Interval	Publication Heartbeat Interval specified in minutes. If no documents are sent on the Publisher channel for this specified interval (duration of time), then a heartbeat document is sent by the driver. A value of 0 indicates that no heartbeat documents are to be sent. If this parameter is not present, by default the publication heartbeat interval is 0.	<definition display-name="Heartbeat Interval (in minutes)" name="pub-heartbeat-interval" type="integer"> <description> </description> <value>0</value> </definition>
pub-heartbeat-interval-seconds Publication Heartbeat Interval (in seconds)	Publication Heartbeat Interval specified in seconds. This parameter can be used instead of pub-heartbeat-interval to provide finer interval size granularity. If no documents are sent on the Publisher channel for this specified interval (duration of time), then a heartbeat document is sent by the driver. A value of 0 indicates that no heartbeat documents are to be sent. If this parameter is not present, by default the publication heartbeat interval is 0.	<definition display-name="Heartbeat Interval (in seconds)" name="pub-heartbeat-interval-seconds" type="integer"> <description> </description> <value>0</value> </definition>
write-timestamps-flag Write Time Stamps?	Whether ndsrep writes special driver time stamp on synchronized Notes parameter. Set to True to have ndsrep write a driver specific time stamp on all Notes objects that are synchronized. This special driver time stamp is used to more accurately determine Notes object attribute updates. Set to False to have ndsrep determine Notes object attribute updates based on existing Notes object time stamps. The default value is True.	<definition display-name="Write Time Stamps?" name="write-timestamps-flag" type="boolean"> <description> </description> <value>true</value> </definition>

4.5 Custom Driver Parameters

You can override many of the driver configuration parameters by using custom driver parameters in policies.

An example of two overrides is shown in Section 4.2, “Automatically Determining Which Certifier to Use,” on page 51. In the Cert.xsl sample style sheet, the certifier ID and certifier password are passed as attributes of the add XML element. The driver finds those parameters and uses the passed values instead of the default values from the driver parameters. The parameters apply as indicated in the Valid Use column of Table 4-4 on page 65.

If an attribute overriding a default configuration parameter is present, it is applied to the note with respect to the event type. Because these parameters often map to items on a note in Lotus Notes, these overrides are passed as the attributes of the event element, or `<add-attr>` children of the event element in the XML document.

Another example is in the sample driver configuration, in the style sheet named `AddAccountNotesOptions.xml`. It utilizes global configuration values (GCVs) specified in `NotesConfig21.xml` to determine which setting to apply.

For items that use *Yes* or *No* values, *True* or *False* values can also be used.

The Notes Driver can add or modify ACLEntries on the ACL record of a Lotus Notes database (`.nsf`) that is being synchronized (Subscriber channel only). Likewise, the ACL of a new mail file that the NotesDriverShim creates for a user can be modified with specific settings at creation time. These parameters are also included in Table 4-5 on page 82.

The Notes Driver can apply database replication settings and replication entry settings. The driver can create a database replica, as well as request to perform replication, which can also create a mailfile replica when a mailfile is initially created.

This enhancement also allows for modifying replication settings of the existing database that is being synchronized, to perform a replication request on the synchronized database, and to create a new replica on an accessible Domino server.

- Section 4.5.1, “ACL Custom Parameters,” on page 81
- Section 4.5.2, “mailfile Custom Parameters,” on page 84
- Section 4.5.3, “Rep Custom Parameters,” on page 94

Table 4-4 *Custom Parameters That Override Driver, Subscriber, and Publisher Parameters*

Parameter Name and Common Description	**Description**	**XML to Define Driver Parameters**
`adminp-server` Administration Process Server	Specifies the Domino server with which to establish an administration process session; or, specifies the Domino server where a console command should be sent. The default is the local server specified in the driver parameters. Example: `adminp-server=“myserver1/acme”`. Requires Notes 6.0.3 or later.	As an attribute to an `add`, `modify`, `move`, `delete`, or `domino-console-command` command element. String
`adminp-rename-user` AdminP Rename User	Specifies whether the NotesDriverShim will attempt to issue a Notes AdminP request to rename a Notes person.	As an XML attribute to the modify element (of a person object). Boolean: true | false

Parameter Name and Common Description	Description	XML to Define Driver Parameters
adminp-web-user-rename AdminP Web User Rename	Specifies whether the NotesDriverShim will attempt to issue a Notes AdminP request to rename a Notes Web User. By default, the NotesDriverShim uses its own detection mechanism and logic to determine if the AdminP rename operation request is for a Notes Web User (Initiate Web User Rename in Domino Directory) or a standard Notes person (Initiate Rename in Domino Directory). This custom parameter allows the modify command to specify if the AdminP rename request should be for a Notes Web user or not.	As an XML attribute to the modify element (of a person object). Boolean: true | false
allow-adminp-support Allow AdminP Support	Specifies whether the command received by the Notes driver shim should allow issuing AdminP requests if possible. The attribute can be True or False. The default is False if not set with the allow-adminp-support driver parameter in the subscriber-options section. This attribute can be used to override Allow Domino AdminP Support in Section 4.4.2, "Subscriber Options," on page 55, on a command-by-command basis. Example: allow-adminp-support="true". Requires Notes 6.0.3 or later.	As an attribute to an add, modify, move, rename, or delete command element. Boolean: true | false
allow-http-password-set Allow HTTP Password Set	Specifies if the modify-password command should attempt to modify the HTTPPassword for a given user. This parameter overrides the default driver parameter value allow-http-password-set, which you can specify in the driver's Subscriber Options parameter section.	As an XML attribute of a modify-password element. Boolean: true | false
allow-userid-password-set Allow user.id Password Set	Specifies if the modify-password command should attempt to modify the password of a given user.id file. This parameter overrides the default driver parameter value allow-userid-password-set, which you can specify in the driver's Subscriber Options parameter section.	As an XML attribute of a modify-password element. Boolean: true | false
AltFullNameLanguage Language of Alternate Full Name	This element specifies the language used for the Alternate Full Name when registering a new user. Like other user attributes, this can be synchronized using an attribute in eDirectory or inserted in a style sheet. See the Lotus Notes documentation for information on setting AltFullNameLanguage for a user.	As an add-attr child element of an add event. String

Parameter Name and Common Description	Description	XML to Define Driver Parameters
AltFullName Alternate Full Name	This element specifies the Alternate Full Name attribute in Notes when registering a new user. Like other user attributes, this can be synchronized using an attribute in eDirectory or inserted in a style sheet. See the Lotus Notes documentation for information on setting AltFullName for a user.	As an add-attr child element of an add event. String
alt-org-unit Alternate Organization Unit	Specifies the alternate Organization Unit when registering a new user in Notes.	As an attribute to an add command element. String
alt-org-unit-lang Alternate Organization Unit Language	Specifies the alternate Organization Unit language when registering a new user in Notes.	As an attribute to an add command element. String
cert-expire-date Certification Expiration Date	Specifies the date when a user certifier expires. This attribute can be applied to override the default expiration term specified in the driver parameters. It is used by the Notes Driver shim when processing events that result in AdminP requests that cause the recertification of the user, such as move, rename, or recertify, or on add event when creating new Notes users. The date format should be specified in text using the appropriate format of the locale of the machine hosting the Notes Driver shim. For example, in English, cert-expire-date="1 July 2010". An alternate to this attribute is expire-term.	As an attribute to an add, modify, or move command element. String
cert-id Certifier ID File	This XML attribute specifies the Notes Certifier ID file that is used to register this user object in the Notes Address Book. The full path of the file should be represented with respect to the operating system hosting Domino. Overrides the default Notes Certifier ID file parameter cert-id-file in the driver configuration.	As an attribute to an add command element String
cert-id-type User ID file certifier type	This XML attribute specifies the User ID file certifier type when user ID files are created at user registration time. Valid values are ID_FLAT, ID_HIERARCHICAL, and ID_CERTIFIER. The absence of this XML attribute sets the default certifier type of ID_HIERARCHICAL.	As an attribute to an add command element. String

Parameter Name and Common Description	Description	XML to Define Driver Parameters
cert-pwd Certifier Password	This XML attribute specifies the Notes Certifier ID password to be used with the certifier ID file. The password value is passed in clear text. The Notes Certifier ID file and password are used to register user objects in the Notes Address Book. Overrides the default Notes Certifier ID file password parameter cert-id-password in the driver configuration. Alternates are named-cert-pwd and drv-param-cert-pwd.	As an attribute to an add command element. String
certificate-authority-org Certificate Authority Organization	This custom parameter is deprecated. The value certificate-authority-org specifies a certificate name, and is used in conjunction with custom parameter use-certificate-authority="true". If Domino CA services are functional, this parameter can be used instead of cert-id (or its alternate: drv-param-cert-id) and cert-pwd (or its alternates: drv-param-cert-pwd or named-cert-pwd) XML attributes. Example: certificate-authority-org="\certwest\west". This custom parameter has an alternate of certifier-name.	As an attribute to an add, modify command element. String: Notes name of the Notes Certifier

Parameter Name and Common Description	Description	XML to Define Driver Parameters
`certifier-name` Certifier Name	For `add` or `modify`, the parameter specifies a certificate name. Use in conjunction with custom parameter `use-certificate-authority=”true”`. If Domino CA services are functional, you can use this parameter instead of `cert-id` (or its alternate: `drv-param-cert-id`) and `cert-pwd` (or its alternates: `drv-param-cert-pwd` or `named-cert-pwd`) XML attributes. Example: `certifier-name=”\certwest\west”`. This custom parameter has a deprecated alternate of certificate-authority-org. For `move`, this specifies the certifier name required to move a user in Notes from an old certifier to a new certifier. The value is the name of the new certifier where the user is moving to. Use this attribute in conjunction with `old-cert-id` or one of its alternates, `old-cert-pwd` or one of its alternates, `cert-id` or one of its alternates, and `cert-pwd` or one of its alternates. The `cert-id` specified should belong to the certifier-name. Example: `certifier-name=‟/mktg/acme”`. Requires Notes 6.0.3 or later.	As an attribute to an `add`, `modify`, or `move` command element. String: the Notes name of the Notes Certifier
`certify-user` Certify User Flag	Applying this XML attribute determines the behavior for the driver regarding Notes user account creation. Its value can be Yes or No. Yes indicates the driver will register this user in the Notes Address book by certifying the user (meaning it creates an ID file for the user). Overrides the default Certify Users flag `cert-users` in the driver configuration.	As an attribute to an `add` command element. String
`comparison-operator` Comparison Operator	Specifies the method of comparison for the value during the Notes database search in order to satisfy a query command.	Within an XDS query document, as an attribute to the value element. String - '=', '<', '>', '<=', '>=', '!='.

Parameter Name and Common Description	Description	XML to Define Driver Parameters
`compute-with-form` Compute With Form	Specifies whether the NotesDriverShim will attempt to `computeWithForm()` prior to saving a newly created or modified Notes Document. When passing a `compute-with-form="true"` XML attribute on a XDS driver command, the NotesDriverShim attempts to execute the Notes document's `computeWithForm()` method, prior to saving any changes to a modified (or newly created) Notes document. The `computeWithForm()` method should appropriately execute any database designs that are associated with computing fields defined by that particular form that is defined within the synchronized Notes database.	As an attribute to an `add`, or `modify` command element. Boolean: true | false Default is *False*
`create-mail` Create Mail File Flag	This XML attribute indicates whether the driver needs to create an e-mail account for this user. Its value can be Yes or No. Yes indicates the driver will attempt to create a Notes Mail database when adding (creating) this new user. Overrides the default Create Mail File flag `create-mail` in the driver configuration.	As an attribute to an `add` command element. Boolean: true | false
`delete-windows-group` Delete Windows Group	Specifies whether synchronized Windows groups should be deleted from Windows or not. The value is True or False. Domino has its own capability of synchronizing users and groups with Windows systems. When the Notes Driver shim utilizes AdminP to delete a group, the request can also indicate that this deletion should be synchronized with Windows. By default, this attribute is set to False. Example: `delete-windows-group=“true”`. Requires Notes 6.0.3 or later.	As an attribute to a `delete class=“group”` command element. Boolean: true | false
`delete-windows-user` Delete Windows User	Specifies whether synchronized Windows users should be deleted from Windows or not. The value is True or False. Domino has its own capability of synchronizing users and groups with Windows systems. When the Notes Driver shim utilizes AdminP to delete a user, the request can also indicate that this deletion should be synchronized with Windows. By default this attribute is set to False. Example: `delete-windows-user=“true”`. Requires Notes 6.0.3 or later.	As an attribute to a `delete class=“user”` command element. Boolean: true | false

Parameter Name and Common Description	Description	XML to Define Driver Parameters
deny-access-group-id Deny Access Group ID	Specifies the Notes deny access group UNID for a delete event. When the Notes Driver shim utilizes AdminP to delete users from Notes, it has the capability to attach a deny access group name to that AdminP delete user request, so the deleted user's name is inserted as a member of the specified deny access group. An alternate attribute is deny-access-group-name. Example: deny-access-group-id="7EFB951A3574521F87256E540001F140". Requires Notes 6.0.3 or later.	As an attribute to a delete command element. String
deny-access-group-name Deny Access Group Name	Specifies the Notes deny access group name for a delete event. When the Notes Driver shim utilizes AdminP to delete users from Notes, it has the capability to attach a deny access group name to that AdminP delete user request, so the deleted user's name is inserted as a member of the specified deny access group. An alternate attribute is deny-access-group-id. Example: deny-access-group-name="Deny Access". Requires Notes 6.0.3 or later.	As an attribute to a delete command element. String
drv-param-cert-id Certifier ID File Parameter Reference	This XML attribute can be used instead of the Certifier ID file cert-id XML attribute. This attribute specifies a driver parameter that holds the Notes Certifier ID file that is used to register this user object in the Notes Address Book. The driver parameter XML attribute can have any name, but its value needs to indicate the full path of the certifier ID file with respect to the operating system hosting Domino. Overrides the default Notes Certifier ID file parameter cert-id-file in the driver configuration.	As an attribute to an add command element String

Parameter Name and Common Description	Description	XML to Define Driver Parameters
drv-param-cert-pwd Certifier Password Parameter Reference	This attribute can be used instead of the Certifier Password cert-pwd XML attribute. This attribute specifies a driver parameter that holds the Notes Certifier ID password to be used with the certifier ID file that is used to register this user object in the Notes Address Book. The driver parameter XML attribute can have any name, but its value indicates the password of the Certifer ID file. The referenced driver parameter can be a clear-text password or an encrypted named password. The Notes Certifier ID file and password are used to register user objects in the Notes Address Book. Overrides the default Notes Certifier ID file password parameter cert-id-password in the driver configuration. Alternates are cert-pwd and named-cert-pwd.	As an attribute to an add command element. String
drv-param-old-cert-id Driver Parameter Old Certifier ID	Specifies the driver parameter holding the old certifier ID file name required to move a user in Notes from an old certifier to a new certifier. The value is the driver parameter XML attribute. An alternate to this attribute is old-cert-id. This attribute should be used in conjunction with certifier-name, old-cert-pwd or one of its alternates, cert-id or its alternate, and cert-pwd or one of its alternates. Example: drv-param-old-cert-id="mktg-cert-id-file". Requires Notes 6.0.3 or later.	As an attribute to a move command element. String
drv-param-old-cert-pwd Driver Parameter Old Certifier Password	Specifies the driver parameter holding the password for the old certifier ID file required to move a user in Notes from an old certifier to a new certifier. The value is the driver parameter XML attribute. An alternate to this attribute is named-old-cert-pwd or old-cert-pwd. This attribute should be used in conjunction with certifier-name, old-cert-id or one of its alternates, cert-id or one of its alternates, and cert-pwd or one of its alternates. Example: drv-param-old-cert-pwd="mktg-cert-id-password". Requires Notes 6.0.3 or later.	As an attribute to a move command element. String

Parameter Name and Common Description	Description	XML to Define Driver Parameters
`enforce-unique-short-name` Enforce Unique Short Name	Specifies whether to enforce uniqueness of short names when registering a new user in Notes. The value is True or False. The default is False. If specified as True, and the Notes user registration process determines that the short name for the new user already exists, then the new user information is overlaid onto the existing Notes user of the same short name, thereby preventing the existence of a duplicate short name. Example: `enforce-unique-short-name="true"`. Requires Notes 6.0.3 or later.	As an attribute to an `add` command element. String
`extended-ou` Extended OU	The value of the XML attribute is appended to the generated DN based on the selected certifier when registering a user.	As an attribute to an `add` or `modify` command element. String
`expire-term` User ID file Expiration Term	This XML attribute specifies the expiration term (specified in years) for the Notes User ID file of this user. Overrides the default Expiration Term parameter `expiration-term` in the driver configuration.	As an attribute to an `add` command element. Integer
`HTTPPassword` Notes HTTP Password	Specifies the user's Web (HTTP) password for Notes. This setting is ignored if the Allow HTTP Password Set parameter `allow-http-password-set` is set to False.	As an `add-attr` or `modify-attr` child element of `an add` or `modify` event. String
`immediate` Immediate	Specifies whether a delete event performed by AdminP immediately deletes a user from the Notes Address Book, or waits until the AdminP request is processed at its scheduled interval. The specified value should be True or False. The default is False. Example: `immediate="true"`. Requires Notes 6.0.3 or later.	As an attribute to a `delete` command element. Boolean: true | false
`InternetAddress` InternetAddress	This element specifies the user's Internet e-mail address in the Notes Address Book.	As an `add-attr` child element of an `add` event. String

Parameter Name and Common Description	Description	XML to Define Driver Parameters
internet-password-force-change Internet Password Force Change	Specifies whether a Notes user is forced to change his or her password on next login. The value of this attribute is True or False. If set to True, the user is forced to change his or her password on next login. If set to False (default), the user is not forced to change the password on next login. When this attribute is attached to a user Add or Modify event, an AdminP Set Password Information request is generated. Example: `internet-password-force-change="true"` Requires Notes 6.0.3 or later.	As an attribute to an `add`, or `modify` command element. Boolean: true | false
match-syntax Match Syntax	Specifies whether the NotesDriverShim's query processor is to interpret the search values using literal Lotus Notes Formula Language Match syntax. If set to True, match syntax special characters (\\, {, }, ?, *, &, !, |, +) are not used literally, but are used as wildcards to conform with the Lotus Script Formula Language's Match syntax.	Within an XDS query document, as an attribute to the search-class element, or as a `search-attr` element's child value element.Boolean: true | false Default is *False*.
minimum-pwd-len Notes Password Strength	Specifies a password strength to apply to the User ID file of newly registered users. Value can be 0 - 16. Overrides the default Notes User ID minimum password strength parameter `minimum-pwd-len` in the driver configuration.	As an attribute to an `add` command element. Integer: 1-16
named-cert-pwd Certifier Password Key Name Reference	This XML attribute can be used instead of the Certifier Password `cert-pwd` XML attribute. This attribute specifies a named-password key name that holds the Notes Certifier ID password to be used with the certifier ID file that is used to register this user object in the Notes Address Book. The Notes Certifier ID file and password are used to register user objects in the Notes Address Book. Overrides the default Notes Certifier ID file password parameter `cert-id-password` in the driver configuration. Alternates are `cert-pwd` and `drv-param-cert-pwd`.	As an attribute to an `add` command element. String

Parameter Name and Common Description	Description	XML to Define Driver Parameters
name-expire-date Name Expiration Date	Specifies the specific date when an old user name will expire after a move user is performed by AdminP. This attribute only has effect when moving non-certified (Web) users. This attribute can be applied to override the default expiration term of 21 days. The date format should be specified in text using the appropriate format of the locale of the Domino Server. For example, in English, name-expire-date="1 July 2010". An alternate to this attribute is name-expiration-days. Requires Notes 6.0.3 or later.	As an attribute to a modify command element. String
name-expiration-days Name Expiration Days	Specifies the specific number of days an old user name can be used before expiration after a move user is performed by AdminP. This attribute only has effect when moving non-certified (Web) users. This attribute can be applied to override the default expiration term of 21 days. An alternate to this attribute is name-expiration-date. Example: name-expiration-days="14". Requires Notes 6.0.3 or later.	As an attribute to a modify command element. Integer
named-old-cert-pwd Named Old Certifier Password	Specifies the Named Password for the old certifier ID file required to move a user in Notes from an old certifier to a new certifier. The value is the named password to be retrieved from the driver configuration. An alternate to this attribute is drv-param-old-cert-pwd or old-cert-pwd. This attribute should be used in conjunction with certifier-name, old-cert-id or one of its alternates, cert-id or one of its alternates, and cert-pwd or one of its alternates. Example: named-old-cert-pwd="mktgNamedPwd". Requires Notes 6.0.3 or later.	As an attribute to a move command element. String
no-id-file No ID File	Specifies whether the Notes registration process creates an ID file for the new user. The value is True or False. The default is False. Example: no-id-file="true". Requires Notes 6.0.3 or later.	As an attribute to an add command element. Boolean: true | false
notes-doc-lock-fail-action Notes Document Lock Failure Action	Specifies the status return code for the specific command if the Notes document locking method fails to obtain a lock.	As an attribute to an add, modify, or modify-pasword command element. String: FATAL, RETRY, ERROR, WARNING, SUCCESS, or UNKNOWN

Parameter Name and Common Description	Description	XML to Define Driver Parameters
notes-policy-name Notes Explicit Policy Name	Specifies an explicit policy name to attach to a user when the user is registered. This attribute does not execute Notes registration policies or any other policies at registration time. Requires Notes 6.0.3 or later.	As an attribute to an add command element. String
notes-save-fail-action Notes Document Save Failure Action	Specifies the status return code for the specific command if the Notes document save method fails.	As an attribute to an add, modify, rename, delete , or modify-pasword command element. String: FATAL, RETRY, ERROR, WARNING, SUCCESS, or UNKNOWN
notes-password-change-interval Notes Password Change Interval	Specifies a Notes user's password change interval. The value of this attribute is a number. The change interval specifies the number of days at which the user must supply a new password. The value defaults to zero. When this attribute is attached to a user add or modify event, an AdminP Set Password Information request is generated. Example: notes-password-change-interval="120" Requires Notes 6.0.3 or later.	As an attribute to an add or modify command element. Integer
notes-password-check-setting Notes Password Check Setting	Specifies a Notes user's password check setting. When this attribute is attached to a user Add or Modify event, an AdminP Set Password Information request is generated. Acceptable values are PWD_CHK_CHECKPASSWORD, PWD_CHK_DONTCHECKPASSWORD, and PWD_CHK_LOCKOUT. Example = notes-password-check-setting=" PWD_CHK_CHECKPASSWORD" Requires Notes 6.0.3 or later.	As an attribute to an add or modify command element. String or integer
notes-password-grace-period Notes Password Grace Period	Specifies a Notes user's password grace period. The value of this attribute is a number. The grace period specifies the number of days an old password is valid after it has expired. The value defaults to zero. When this attribute is attached to a user add or modify event, an AdminP Set Password Information request is generated. Example: notes-password-grace-period="10" Requires Notes 6.0.3 or later.	As an attribute to an add or modify command element. Integer

Parameter Name and Common Description	Description	XML to Define Driver Parameters
`old-certifier-name` Old Certifier Name	Specifies a certificate name that designates the origin of a Notes move operation. Use this parameter in conjunction with the parameter `use-certificate-authority=”true”`. If Domino CA services are functional, this custom parameter can be used instead of `cert-id` (or its alternate: `drv-param-cert-id`) and `cert-pwd` (or its alternates: `drv-param-cert-pwd` or `named-cert-pwd`) XML attributes. Example: `old-certifier-name=”\certwest\west”` This custom parameter has a deprecated alternate of `old-certificate-authority-org`.	As an attribute to an `add`, or `modify` command element. String: Notes name of the old Notes Certifier
`old-cert-use-certificate-authority` Old Certifier Use Certificate Authority	Specifies if the Domino CA process is to be used for the old certifier that is required for a Notes move operation. It is used in conjunction with custom parameter `old-certifier-name` or its deprecated alternate `old-certificate-authority-org`.	As an attribute to an `modify` command element. Boolean: true | false Default is *False*.
`old-certificate-authority-org` Old Certificate Authority Organization	This custom parameter is deprecated. The parameter `old-certificat-authority-org` specifies a certificate name that designates the origin of a Notes `move` operation. This is used in conjunction with custom parameter `old-cert-use-certificate-authority=”true”`. If Domino CA services are functional, this custom parameter can be used instead of `old-cert-id` (or its alternate: `drv-param-old-cert-id`) and `old-cert-pwd` (or its alternates: `drv-param-old-cert-pwd` or `named-old-cert-pwd`) XML attributes. Example: `old-certificate-authority-ord=”\certwest\west”` This custom parameter has an alternate of `old-certifier-name`.	As an attribute to an `add`, or `modify` command element. String: Notes name of the old Notes Certifier

Parameter Name and Common Description	Description	XML to Define Driver Parameters
`old-cert-id` Old Certification ID	Specifies the old certifier ID file required to move a user in Notes from an old certifier to a new certifier. The value is the full path and filename of the old certifier ID file. An alternate to this attribute is `drv-param-old-cert-id`. This attribute should be used in conjunction with `certifier-name, old-cert-pwd` or one of its alternates, `cert-id` or one of its alternates, and `cert-pwd` or one of its alternates. Example: `old-cert-id="c:\lotus\domino\data\mktgcert.id"` Requires Notes 6.0.3 or later.	As an attribute to a `move` command element. String
`old-cert-pwd` Old Certification Password	Specifies the password for the old certifier ID file required to move a user in Notes from an old certifier to a new certifier. The value is the password string. An alternate to this attribute is `drv-param-old-cert-pwd` or `named-old-cert-pwd`. This attribute should be used in conjunction with `certifier-name`, `old-cert-id` or one of its alternates, `cert-id` or one of its alternates, and `cert-pwd` or one of its alternates. Example: `old-cert-pwd="mktg-password1"` Requires Notes 6.0.3 or later.	As an attribute to a `move` command element. String
`recertify-user` Recertify User	Causes a recertify user request to be sent to AdminP. The attribute value is specified as True or False. Proper certifier ID and password attributes (`cert-id`, `cert-pwd,` or equivalents) must be provided or the default certifier is applied to the AdminP recertification request. This attribute should be used in conjunction with `cert-id` or its alternate, and `cert-pwd` or one of its alternates. Can be used in conjunction with `expire-term` or `cert-expire-date` elements for specify the new expiration term of the user's certifier. Example: `recertify-user="true"` Requires Notes 6.0.3 or later.	As an attribute to a `modify` command element. Boolean: true | false
`registration-dest-dn-check` Notes Registration FullName Uniqueness Check	Specifies whether the NotesDriverShim should check for `FullName` field uniqueness of a new Notes person prior to the Notes registration process.	As an attribute to an `add` command element. Boolean: true | false Default is *True*

Parameter Name and Common Description	Description	XML to Define Driver Parameters
`remove-all-group-membership` Group Membership Removal	This XML attribute indicates that this user object should be removed from the membership list of all groups in the Notes database, except for groups of type "Deny List" (GroupType=3). Valid values are True or False. The absence of this XML attribute defaults to False. This tag only applies to person (user) objects in the Notes Address Book.	As an attribute to a `modify` or `delete` command element. Boolean: true | false
`roaming-cleanup-period` Roaming Cleanup Period	Specifies the Notes client's cleanup interval in days for a roaming user when the roaming user's cleanup setting is set to `"CLEANUP_EVERY_NDAYS."` This attribute should be used in conjunction with `roaming-user="true,"` and `roaming-cleanup-setting="EVERY_NDAYS."` Example: `roaming-cleanup-period="90"` Requires Notes 6.0.3 or later.	As an attribute to an `add` command element. Integer
`roaming-cleanup-setting` Roaming Cleanup Setting	Specifies the roaming user cleanup process for Notes client data. Valid values are `AT_SHUTDOWN`, `EVERY_NDAYS`, `NEVER`, and `PROMPT`. The default value is `NEVER`. This attribute should be used in conjunction with roaming-user="true." Example: `roaming-cleanup-setting="AT_SHUTDOWN"` Requires Notes 6.0.3 or later.	As an attribute to an `add` command element. String
`roaming-server` Roaming Server	Specifies the name of the Domino server to store the roaming user data. This attribute should be used in conjunction with roaming-user="true." Example: `roaming-server="CN=myserver2/O=acme"` Requires Notes 6.0.3 or later.	As an attribute to an `add` command element. String
`roaming-subdir` Roaming Subdirectory	Specifies the subdirectory below the Domino server's data directory where user roaming data is stored. The last character of the value should be a file path separator (/ or \). Example: `roaming-subdir="roamdata\"`. Requires Notes 6.0.3 or later.	As an attribute to an `add` command element. String
`roaming-user` Roaming User	Specifies whether the Notes registration process creates this user with roaming capabilities. The value is True or False. The default is False. Example: `roaming-user="true"` Requires Notes 6.0.3 or later.	As an attribute to an `add` command element. Boolean: true | false

Parameter Name and Common Description	Description	XML to Define Driver Parameters
`store-useridfile-in-ab` Store User ID File In Notes Address Book	This XML attribute specifies if the driver attaches the user ID file for this user onto its user object in the Notes Address Book at registration time. Setting the attribute to True causes this registered user object in the Notes Address Book to be created with an attached user ID file. Setting the attribute to False causes this registered user object in the Notes Address Book to be created without an attached user ID file. Overrides the default Store UserID in Address Book parameter `store-id-ab-flag` in the driver configuration.	As an attribute to an `add` command element. Boolean: true | false
`sync-internet-password` Synchronize Internet Password	Specifies whether a user's Internet password (HTTPPassword) is synchronized to match the user's Notes Client ID password, by means of the background processes of the Domino server. The value is True or False. The default is False. Example: `sync-internet-password="true"` Requires Notes 6.0.3 or later.	As an attribute to an `add` command element. Boolean: true | false
`tell-adminp-process` Domino Console Command	Specifies the Domino console command to perform after an AdminP request has been performed by the Notes driver shim. For Domino console commands to succeed, the Notes Driver user must have appropriate Domino Console privileges granted. Example: `tell-adminp-process="tell adminp process new"`. Requires Notes 6.0.3 or later. See the instructions in Section 4.10, "TELL AdminP Commands," on page 102.	As an attribute to an `add`, `modify`, `move`, `delete` command element. String
`update-addressbook` Registered Users in Notes Address Book	This XML attribute specifies if the driver puts registered user objects in the Notes Address Book. Setting the attribute to True causes registered users to be placed in the address book. Setting the attribute to False causes users to be registered (that is, a certifier ID file is created for the user) without the user object being placed into the Notes Address Book. Overrides the default Update Address Book parameter `update-ab-flag` in the driver configuration.	As an attribute to an `add` command element. Boolean: true | false

Parameter Name and Common Description	Description	XML to Define Driver Parameters
`use-certificate-authority` Use Certificate Authority	Specifies if the Domino CA process should be used to register, recertify, rename, or move a user. Requires certify-name to be provided, instead of `cert-id` (or its alternate: `drv-param-cert-id`) and `cert-pwd` (or one of its alternates: `drv-param-cert-pwd` or `named-cert-pwd`).	As an attribute to an `add`, or `modify` command element. Boolean: true | false Default is *False*
`user-id-file` ID File Name	This XML attribute specifies the filename to be used for the user's ID file. The filename does not include the file path. When this XML attribute is absent, a default filename is generated by the Notes driver by using the first and last name attributes of the user (FirstNameLastName.id).	As an attribute to an `add` command element. String
`user-id-path` ID File Path	This XML attribute specifies the file path to the Notes User ID file storage location to be used when creating the user's ID file. The new ID file is placed in this location. The full path of the folder should be represented with respect to the operating system hosting Domino. Overrides the default Notes User ID certificate location parameter <cert-path> in the driver configuration.	As an attribute to an `add` command element. String
`user-pwd` Notes Password	The user's Notes password used to create the user's ID file (certifier). Overrides the default Notes Password parameter `default-password` in the driver configuration.	As an attribute to an `add` command element. String

4.5.1 ACL Custom Parameters

The following table lists the XML element attributes that can be inserted in the `add` or `modify` command element in an XDS document that is submitted to the NotesDriverShim to appropriately configure the synchronized database's ACL settings. The first column of the table contains the parameter name followed by its common description. The parameters are alphabetized according to the XML parameter name.

The third column of the following table contains XML text that you can paste into the Driver Parameters XML Editor. The XML text represents exactly what is necessary to display the parameters.

Table 4-5 ACL Custom Parameters

Parameter Name and Common Description	Description	XML to Define Driver Parameters
acl-admin-names ACL Admin Name Modifier	The ACL isAdminNames property indicates whether the administration server can modify the names fields in this database. Only available on Domino 6.0.3 or higher.	As an attribute to an add, modify, or delete command element. Boolean: true | false
acl-admin-reader-author ACL Admin Reader-Author Modifier	The ACL isAdminReaderAuthor property indicates whether the administration server can modify the Readers and Authors fields in this database. This is only available on Domino 6.0.3 or higher.	As an attribute to an add, modify, or delete command element. Boolean: true | false
acl-administration-server ACL Administration Server	The name of the Administration Server that can perform maintenance on this database. This value must be a Notes Distinguished Name of a valid Domino Administration Server. Only available on Domino 6.0.3 or higher.	As an attribute to an add, modify, or delete command element. String: distinguished Notes object name (i.e. "CN=Server1/O=myOrg")
acl-entry-admin-reader-author ACL Admin Reader-Author	Indicates if the admin-reader-author attribute is set.	As an attribute to an add, modify, or delete command element. Boolean: true | false
acl-entry-admin-server ACL Admin Server	Indicates if the admin-server attribute is set. If set to True, this setting overrides the ACL Administration Server property and places this entry's name in the ACLs Administration Server property. See acl-administration-server.	As an attribute to an add, modify, or delete command element. Boolean: true | false
acl-entry-can-create-documents ACL Create Documents	Indicates if the can-create-documents attribute is set.	As an attribute to an add, modify, or delete command element. Boolean: true | false
acl-entry-can-create-ls-or-java-agent ACL Create LS Or Java Agent	Indicates if the can-create-LS-or-Java attribute is set.	As an attribute to an add, modify, or delete command element. Boolean: true | false
acl-entry-can-create-personal-agent ACL Create Personal Agent	Indicates if the can-create-personal-agent attribute is set.	As an attribute to an add, modify, or delete command element. Boolean: true | false

Parameter Name and Common Description	Description	XML to Define Driver Parameters
acl-entry-can-create-personal-folder ACL Create Personal Folder	Indicates if the can-create-personal-folder attribute is set.	As an attribute to an add, modify, or delete command element. Boolean: true | false
acl-entry-can-create-shared-folder ACL Create Shared Folder	Indicates if the can-create-shared-folder attribute is set.	As an attribute to an add, modify, or delete command element. Boolean: true | false
acl-entry-can-delete-documents ACL Delete Documents	Indicates if the can-delete-documents is set.	As an attribute to an add, modify, or delete command element. Boolean: true | false
acl-entry-can-replicate-or-copy-documents ACL Replicate Or Copy Documents	Indicates if the can-replicate-or-copy-documents attribute is set. Only available on Domino 6.0.3 or higher.	As an attribute to an add, modify, or delete command element. Boolean: true | false
acl-entry-disable-role ACL Disable Role	Specifies the roles to set on the ACL.	As an attribute to an add, modify, or delete command element. String
acl-entry-enable-role ACL Enable Role	Specifies the roles to set on the ACL.	As an attribute to an add, modify, or delete command element. String
acl-entry-level ACL Entry Level	Indicates the ACLEntry level. Valid strings include: MANAGER DESIGNER EDITOR AUTHOR READER DEPOSITOR NOACCESS	As an attribute to an add, modify, or delete command element. String or Integer
acl-entry-name ACL Entry Name	The name of the ACLEntry. If not present, this value defaults to the Notes Distinguished Name (FullName) of the current object being synchronized.	As an attribute to an add, modify, or delete command element. String: distinguished Notes object name (i.e. “CN=John Doe/OU=myOrgUnit/O=myOrg”)
acl-entry-public-reader ACL Public Reader	Indicates if the public-reader attribute is set.	As an attribute to an add, modify, or delete command element. Boolean: true | false

Parameter Name and Common Description	Description	XML to Define Driver Parameters
`acl-entry-public-writer` ACL Public Writer	Indicates if the `public-writer` attribute is set.	As an attribute to an `add`, `modify`, or `delete` command element. Boolean: `true	false`
`acl-entry-remove` ACL Entry Remove	Indicates if the ACLEntry is to be removed from the ACL record of the database.	As an attribute to an `add`, `modify`, or `delete` command element. Boolean: `true	false`
`acl-entry-server` ACL Server	Indicates if the ACL Server attribute is set.	As an attribute to an `add`, `modify`, or `delete` command element. Boolean: `true	false`
`acl-entry-user-type` ACL User Type	Indicates the ACLEntry user type. Valid strings include: `MIXED_GROUP PERSON PERSON_GROUP SERVER SERVER_GROUP UNSPECIFIED`	As an attribute to an `add`, `modify`, or `delete` command element. String or Integer
`acl-extended-access` ACL Extended Access	The ACL isExtendedAccess property indicates whether extended access is enabled for this database. Only available on Domino 6.0.3 or higher.	As an attribute to an `add`, `modify`, or `delete` command element. Boolean: `true	false`
`acl-internet-level` ACL Internet Level	The maximum internet access level for this database. Valid strings include: `MANAGER DESIGNER EDITOR AUTHOR READER DEPOSITOR NOACCESS`	As an attribute to an `add`, `modify`, or `delete` command element. String or Integer
`acl-uniform-access` ACL Uniform Access	The ACL isUniformAccess property indicates whether a consistent ACL is enforced across all replicas of this database.	As an attribute to an `add`, `modify`, or `delete` command element. Boolean: `true	false`

Table 4-6 *ACL Description and Its Java ACL Constant*

ACL Description	Notes Java ACL Constant
NOACCESS DEPOSITOR READER	ACL.LEVEL_NOACCESS ACL.LEVEL_DEPOSITOR ACL.LEVEL_READER
AUTHOR EDITOR DESIGNER	ACL.LEVEL_AUTHOR ACL.LEVEL_EDITOR ACL.LEVEL_DESIGNER
MANAGER	ACL.LEVEL_MANAGER

4.5.2 mailfile Custom Parameters

The following table lists the XML element attributes that can be inserted into the `add` command element in an XDS document that is submitted to the NotesDriverShim to appropriately configure a mailfile database. The first column of the table contains the parameter name followed by its common description. The parameters are alphabetized according to the XML parameter name.

The third column of the following table contains XML text that you can paste into the Driver Parameters XML Editor. The XML text represents exactly what is necessary to display the parameters.

XML element attributes used for mailfile replication settings (prefixed with mailfile-) can only be used within add command elements. These attributes are divided into `mailfile-ACL` and MailFile Rep attributes in this table.

Table 4-7 *Mailfile Custom Parameters*

Parameter Name and Common Description	Description	XML to Define Driver Parameters
`mail-acl-manager-id` Mailfile ACL Manager ID	Specifies the UNID of a user to be given manager credentials on the ACL of the mailfile of the newly created user. An alternate is `mail-acl-manager-name`. Example: `mail-acl-manager-id=" BB888BB0C35D13EC87256EA8006296CE"` As of NotesDriverShim v3.5, `mail-acl-manager-id` can appropriately handle Notes Person UNID values and Notes Group UNID values.	As an attribute to an `add` command element.
`mail-acl-manager-name` Mailfile ACL Manager Name	Specifies the name of a user to be given manager credentials on the ACL of the mailfile of the newly created user. An alternate is `mail-acl-manager-id`. Example: `mail-acl-manager-name="CN=Notes Admin/O=acme"`	As an attribute to an `add` command element.
`mail-acl-manager-group` Mailfile ACL Manager Group	Specifies the name of a Notes group or groups to be given manager credentials on the ACL of the mailfile of the newly created user. The ACL entry type created when using this attribute is MIXED_GROUP. More than one group name can be specified by using a semicolon separator. Example: `mail-acl-manager-group="LocalDomainAdmins;MailAdmins"`.	As an XML attribute to an `add` command element. String - name of Notes Group (LocalDomainAdmins).
`MailDomain` MailDomain	This element specifies the name of the Notes Mail Domain when creating an e-mail database file.	As an `add-attr` child element of an `add` event. String
`MailFile` MailFile	This element specifies the filename to be used when creating the user's e-mail database file. The filename does not include the file path. When this tag is absent, a default filename is generated by the Notes driver by using the first and last name attributes of the user (`FirstNameLastName.nsf`).	As an `add-attr` child element of an `add` event. String

Parameter Name and Common Description	Description	XML to Define Driver Parameters
`mail-file-action` MailFile Action	Specifies the AdminP action to perform on the mailbox of a deleted user. This action is included in an AdminP user delete request. Acceptable values are ALL,HOME, and NONE. The default value is NONE. ALL indicates to delete the mailbox on the home mail server and all mailbox replicas. HOME indicates to delete the mailbox on only the home mail server. All AdminP delete mailbox requests must be approved by a Domino Administrator before they are performed. Example: `mail-file-action="ALL"`. Requires Notes 6.0.3 or later.	As an attribute to a `delete` command element.
`mail-file-inherit-flag` Database Inheritance for Mail File Template	This XML attribute specifies whether database structures based on a particular template are updated when that template is updated. Its value can be `True` or `False`. The default (the absence of this tag) is `True`. You can override the default and set this XML attribute to False if you don't want a change to a mail file template to affect existing database design.	As an attribute to an `add` command element. Boolean: true | false
`mail-file-subdir` MailFile Subdirectory	Specifies the subdirectory below the Domino server's data directory where the mailfile of a new user should be created. Example: `mail-file-subdir="mail-dbs"`	As an attribute to an `add` command element.
`mail-file-quota` Mail File Size Quota	This XML attribute specifies the value of the mail file quota (size in KB) that is applied to the e-mail database file when it is created.	As an attribute to an `add` command element. Integer
`mailfile-acl-administration-server` MailFile ACL Administration Server	The name of the Administration Server that can perform maintenance on this database. This value must be a Notes Distinguished Name of a valid Domino Administration Server. Only available on Domino 6.0.3 or higher.	As an attribute to an `add` command element. String: distinguished Notes object name (i.e. "`CN=Server1/O=myOrg`")
`mailfile-acl-admin-names` MailFile ACL Admin Names	The ACL isAdminNames property indicates whether the administration server can modify the names fields in this database. Only available on Domino 6.0.3 or higher.	As an attribute to an `add` command element. Boolean: `true | false`
`mailfile-acl-admin-reader-author` MailFile ACL Admin Reader Author	The ACL isAdminReaderAuthor property indicates whether the administration server can modify the Readers and Authors fields in this database. Only available on Domino 6.0.3 or higher.	As an attribute to an `add` command element. Boolean: `true | false`

Parameter Name and Common Description	Description	XML to Define Driver Parameters
mailfile-acl-entry-admin-reader-author MailFile ACL Entry Admin Reader-Author	Indicates if the admin-reader-author attribute is set.	As an attribute to an add command element. Boolean: true | false
mailfile-acl-entry-admin-server MailFile ACL Entry Admin Server	Indicates if the admin-server attribute is set. If set to True, this setting overrides the ACL Administration Server property and places this entry's name in the ACLs Administration Server property. See acl-administration-server.	As an attribute to an add command element. Boolean: true | false
mailfile-acl-entry-can-create-documents MailFile ACL Entry Create Documents	Indicates if the can-create-documents attribute is set.	As an attribute to an add command element. Boolean: true | false
mailfile-acl-entry-can-create-ls-or-java-agent MailFile ACL Entry Create LS Or Java Agent	Indicates if the can-create-LS-or-Java attribute is set.	As an attribute to an add command element. Boolean: true | false
mailfile-acl-entry-can-create-personal-agent MailFile ACL Entry Create Personal Agent	Indicates if the can-create-personal-agent attribute is set.	As an attribute to an add command element. Boolean: true | false
mailfile-acl-entry-can-create-personal-folder MailFile ACL Entry Create Personal Folder	Indicates if the can-create-personal-folder attribute is set.	As an attribute to an add command element. Boolean: true | false
mailfile-acl-entry-can-create-shared-folder MailFile ACL Entry Create Shared Folder	Indicates if the can-create-shared-folder attribute is set.	As an attribute to an add command element. Boolean: true | false
mailfile-acl-entry-can-delete-documents MailFile ACL Entry Delete Documents	Indicates if the can-delete-documents is set.	As an attribute to an add command element. Boolean: true | false

Parameter Name and Common Description	Description	XML to Define Driver Parameters
mailfile-acl-entry-can-replicate-or-copy-documents MailFile ACL Entry Replicate Or Copy Documents	Indicates if the can-replicate-or-copy-documents attribute is set. This is only available on Domino 6.0.3 or higher.	As an attribute to an add command element. Boolean: true | false
mailfile-acl-entry-disable-role MailFile ACL Entry Disable Role	Specifies the roles to set on the ACL (by default, mailfile ACLs do no have any roles defined).	As an attribute to an add command element. String
mailfile-acl-entry-enable-role MailFile ACL Entry Enable Role	Specifies the roles to set on the ACL (by default, mailfile ACLs do no have any roles defined).	As an attribute to an add command element. String
mailfile-acl-entry-level MailFile ACL Entry Level	Is the equivalent to mailfile-acl-level. Indicates the ACLEntry level. Valid strings include: MANAGER DESIGNER EDITOR AUTHOR READER DEPOSITOR NOACCESS	As an attribute to an add command element. String or Integer
mailfile-acl-entry-name MailFile ACL Entry Name	The name of the ACLEntry. If not present, this value defaults to the Notes Distinguished Name (FullName) of the current object being synchronized.	As an attribute to an add command element. String: distinguished Notes object name (i.e. “CN=John Doe/OU=myOrgUnit/O=myOrg”)
mailfile-acl-entry-public-reader MailFile ACL Entry Public Reader	Indicates if the public-reader attribute is set.	As an attribute to an add command element. Boolean: true | false
mailfile-acl-entry-public-writer MailFile ACL Entry Public Writer	Indicates if the public-writer attribute is set.	As an attribute to an add command element. Boolean: true | false
mailfile-acl-entry-remove MailFile ACL Entry Remove	Indicates if the ACLEntry is to be removed from the ACL record of the mailfile database.	As an attribute to an add command element. Boolean: true | false

Parameter Name and Common Description	Description	XML to Define Driver Parameters
`mailfile-acl-entry-server` MailFile ACL Entry Server	Indicates if the ACLEntry server attribute is set	As an attribute to an `add` command element. Boolean: `true	false`
`mailfile-acl-entry-user-type` MailFile ACL Entry User Type	Indicates the ACLEntry user type. Valid strings include: `MIXED_GROUP PERSON PERSON_GROUP SERVER SERVER_GROUP UNSPECIFIED`	As an attribute to an `add` command element. String or Integer
`mailfile-acl-extended-access` MailFile ACL Extended Access	The ACL isExtendedAccess property indicates whether extended access is enabled for this database. This is only available on Domino 6.0.3 or higher.	As an attribute to an `add` command element. Boolean: true | false
`mailfile-acl-internet-level` MailFile ACL Internet Level	The maximum Internet access level for this database. Valid strings include: `MANAGER DESIGNER EDITOR AUTHOR READER DEPOSITOR NOACCESS`	As an attribute to an `add` command element. String or Integer
`mailfile-acl-level` MailFile ACL Control	The default ACL setting for the newly created mail file of newly created user objects. Valid values are: `NOACCESS, DEPOSITOR, READER, AUTHOR, EDITOR, DESIGNER`, and `MANAGER`. Values can be specified either as the Java ACL constant or the role name, as found in Table 4-6 on page 84. This attribute should be added in the same rule where the certification attributes are calculated and set and it should be added using the same XSL constructs. Overrides the default Mail File ACL Level parameter `mailfile-acl-level` in the driver configuration.	As an attribute to an `add` command element.
`mailfile-acl-uniform-access` MailFile ACL Uniform Access	The ACL isUniformAccess property indicates whether a consistent ACL is enforced across all replicas of this database.	As an attribute to an `add` command element. Boolean: true | false
`mailfile-adminp-create` Administration Process Mailfile Creation	Specifies if the creation of a mailfile is to be performed via AdminP (in the background). When set to *True*, the Notes user's mailfile is created via AdminP. Example: `mailfile-adminp-create ="true"`.	As an attribute to an add command element. Boolean: true | false
`mailfile-calprofile-create` Mailfile Owner Attribute Creation	Specifies if a mailfile calendar profile document is created within a newly created mailfile, indicating the mailfile owner for the mailfile database.	As an attribute to an `add`, or `modify` command element. Boolean: *true* | *false* Default is *True*

Parameter Name and Common Description	Description	XML to Define Driver Parameters
`mailfile-calprofile-owner` Mailfile Owner	Specifies the mailfile owner field that is inserted in the mailfile calendar profile document of the a newly created mailfile database.	As an attribute to an `add`, or `modify` command element. String: The owner's name of the newly created Notes mailfile. The name should be in Notes canonical format.
`mailfile-template` MailFileTemplate	This XML attribute specifies the filename of the `.ntf` database template to use when creating the user's new mail file for an e-mail account. This template must be accessible to the Domino server in the Domino data folder. Overrides the default Mail File Template `mailfile-template` in the driver configuration.	As an attribute to an `add` command element
`mailfile-rep-abstract` MailFile Rep Abstract	Specifies whether large documents should be truncated and attachments removed during replication. Set to True to truncate large documents and remove attachments, and False to replicate large documents in their entirety.	As an attribute to an `add` command element. Boolean: true | false
`mailfile-rep-background` Background Mailfile Replica Creation	Specifies if creating a mailfile replica is performed via AdminP (in the background). When set to True, the mailfile replica is created via AdminP on the server(s) specified by the `mailfile-rep-new-server` custom parameter. Example: `mailfile-rep-background="true"`.	As an attribute to an `add` command element. Boolean: true | false
`mailfile-rep-clear-history` MailFile Rep Clear History	Set to True to clear the replication history from the replication object, and set to False affects nothing.	As an attribute to an `add` command element. Boolean: true | false
`mailfile-rep-cutoff-delete` MailFile Rep Cuttoff Delete	Specifies whether documents that are older than the cutoff date should be automatically deleted. The cutoff date is determined by today's date minus the cutoff interval (`mailfile-rep-cutoff-interval`). Set to True to automatically delete documents that are older than the cutoff date, and False to not delete old documents.	As an attribute to an `add` command element. Boolean: true | false
`mailfile-rep-cutoff-interval` MailFile Rep Cutoff Interval	Specifies the number of days after which documents are automatically deleted if the CutoffDelete property is set (see `mailfile-rep-cutoff-delete`).	As an attribute to an `add` command element. Integer: numeric value
`mailfile-rep-disabled` MailFile Rep Disabled	Specifies whether replication is disabled. Set to True to disable replication, and False to enable replication.	As an attribute to an `add` command element. Boolean: true | false

Parameter Name and Common Description	Description	XML to Define Driver Parameters
`mailfile-rep-dest-server` MailFile Rep Destination Server	Specifies the Domino destination server (When computer) of a replication entry (within the replication object). If specified, and the source server/destination server pair does not already exist within the replication entry list, a new replication entry is created. If this parameter is absent, the destination server defaults to “any server” (a hyphen “-” means any server). To be used in connection with `mailfile-rep-src-server`. Only available on Domino 6.0.3 or higher.	As an attribute to an `add` command element. String: distinguished name of a replica Domino destination server, such as i.e. `CN=server1/O=acme`.
`mailfile-rep-dont-send-local-security-updates` MailFile Rep Don’t Send Local Security Updates	Specifies whether local security (encryption) updates are sent. Set to True to not send local security updates, and False to send local security updates. Only available on Domino 6.0.3 or higher.	As an attribute to an `add` command element. Boolean: true | false
`mailfile-rep-entry-remove` MailFile Rep Entry Remove	Set to True to remove the current replication entry from the replication object, and set to False affects nothing. Used in connection with `mailfile-rep-src-server` and/or `mailfile-rep-dest-server`.	As an attribute to an `add` command element. Boolean: true | false
`mailfile-rep-formula` MailFile Rep Formula	Specifies the replication formula for a replication entry. By default, a new replication entry contains the @All formula. The formula must be a valid replication formula. Used in connection with `mailfile-rep-src-server` and/or `mailfile-rep-dest-server`. Only available on Domino 6.0.3 or higher.	As an attribute to an `add` command element. String: replication formula
`mailfile-rep-immediate` MailFile Rep Immediate	Indicates that database replication must begin immediately to the specified Domino server on which an existing database replica already exists.	As an attribute to an `add` command element. String: distinguished name of a replica Domino destination server, such as `CN=server1/O=acme.`
`mailfile-rep-include-acl` MailFile Rep Include ACL	Specifies the inclusion of the ACL during replication for a replication entry. Set True to include the ACL, and False to exclude the ACL. Used in connection with mailfile-rep-src-server and/or mailfile-rep-dest-server. Only available on Domino 6.0.3 or higher.	As an attribute to an `add` command element. Boolean: true | false

Parameter Name and Common Description	Description	XML to Define Driver Parameters
mailfile-rep-include-agents MailFile Rep Include Agents	Specifies the inclusion of agents during replication for a replication entry. Set to True to include agents, and False to exclude agents. Used in connection with mailfile-rep-src-server and/or mailfile-rep-dest-server. Only available on Domino 6.0.3 or higher.	As an attribute to an add command element. Boolean: true | false
mailfile-rep-include-documents MailFile Rep Include Documents	Specifies the inclusion of documents during replication for a replication entry. Set to True to include documents, and False to exclude documents. Used in connection with mailfile-rep-src-server and/or mailfile-rep-dest-server. Only available on Domino 6.0.3 or higher.	As an attribute to an add command element. Boolean: true | false
mailfile-rep-include-forms MailFile Rep Include Forms	Specifies the inclusion of forms during replication for a replication entry. Set to True to include forms, and False to exclude forms. Used in connection with mailfile-rep-src-server and/or mailfile-rep-dest-server. Only available on Domino 6.0.3 or higher.	As an attribute to an add command element. Boolean: true | false
mailfile-rep-include-formulas MailFile Rep Include Formulas	Specifies the inclusion of formulas during replication for a replication entry. Set to True to include formulas, and False to exclude formulas. Used in connection with mailfile-rep-src-server and/or mailfile-rep-dest-server. Only available on Domino 6.0.3 or higher.	As an attribute to an add command element. Boolean: true | false
mailfile-rep-ignore-deletes MailFile Rep Ignore Deletes	Specifies whether outgoing deletions should not be replicated to other databases. Set to True to not replicate outgoing deletions, and False to replicate outgoing deletions.	As an attribute to an add command element. Boolean: true | false
mailfile-rep-ignore-dest-deletes MailFile Rep Ignore Destination Deletes	Specifies whether incoming deletions should not be replicated to the current database. Set to True to not replicate incoming deletions, and False to replicate incoming deletions.	As an attribute to an add command element. Boolean: true | false
mailfile-rep-new-db-name MailFile Rep New DB Name	The filename of the newly created replica. If mailfile-rep-new-db-name is not present, the filename of the original database is used. The default location of the new file is in the Domino server's data folder. Can be used in connection with mailfile-rep-new-server.	As an attribute to an add command element. String: file name of the new replica, such as mail/JohnDoeRep2.nsf.

Parameter Name and Common Description	Description	XML to Define Driver Parameters
`mailfile-rep-new-server` MailFile Rep New Server	The name of the Domino server where a new replica will be created. The Domino server must be accessible on the network. Depending on the size of the database, this might be a time-consuming process for the NotesDriverShim. Can be used in connection with `mailfile-rep-new-db-name`.	As an attribute to an `add` command element. String: distinguished name of Domino server where a new replica will be created, such as `CN=server1/O=acme`.
`mailfile-rep-priority` MailFile Rep Priority	Specifies the replication priority. The choices are `HIGH`, `MED`, and `LOW`. The default is Medium (`MED`).	As an attribute to an `add` command element. String or integer
`mailfile-rep-src-server` MailFile Rep Source Server	Specifies the Domino source server (Receives from) of a replication entry within the replication object. If specified, and the source server/destination server pair does not already exist within the replication entry list, a new replication entry is created. If this parameter is absent, the source server defaults to “any server” (a hyphen '-' means any server). To be used in connection with `mailfile-rep-dest-server`. Only available on Domino 6.0.3 or higher.	As an attribute to an `add` command element. String: distinguished name of a replica Domino source server, such as `CN=server2/O=acme`.
`mailfile-rep-view-list` MailFile Rep View List	Specifies a list of view names to be replicated for a replication entry. The string specifies the views as a list, separating view names with semicolons, such as Inbox; Sent; Calendar; Meetings. Used in connection with `mailfile-rep-src-server` and/or `mailfile-rep-dest-server`. Only available on Domino 6.0.3 or higher.	As an attribute to an `add` command element. String: ViewList
`mail-quota-warning-threshold` MailFile Quota Warning Threshold	Specifies the value of the mailfile quota warning threshold (size in KB) that is applied to the e-mail database file when it is created. Example: `mail-quota-warning-threshold=“120000”`. Requires Notes 6.0.3 or later.	As an attribute to an `add` command element.
`MailServer` MailServer	This element specifies the name of the Notes Server where the mail file should be created when creating an e-mail account (mail database file).	As an `add-attr` child element of an `add` event. String
`mail-server` Mail Server	Specifies the mail server to be used to create a mailfile for a new user. This attribute overrides the value specified in the driver parameters. Example: `mail-server=“CN=ms2/O=acme”` Alternate: See the `MailServer` custom parameter.	As an attribute to an `add` command element. String

Parameter Name and Common Description	Description	XML to Define Driver Parameters
mail-system Mail System	Specifies the mail system type set for the new user being created. Valid values are NOTES, POP, INTERNET, OTHER, NONE. The default value is NOTES. Requires Notes 6.0.3 or later.	As an attribute to an add command element. String or integer
match-syntax Match Syntax	Specifies whether the NotesDriverShim's query processor is to interpret the search values using literal Lotus Notes Formula Language Match syntax. If set to True, match syntax special characters (\, {, }, ?, *, &, !, |, +) are not used literally, but are used as wildcards to conform with the Lotus Script Formula Language's Match syntax.	Within an XDS query document, as an attribute to the search-class element, or as a search-attr element's child value element.Boolean: true | false Default is *False*.

4.5.3 Rep Custom Parameters

The following table lists the XML element attributes that can be inserted into the add or modify command element in an XDS document that is submitted to the NotesDriverShim to appropriately configure a synchronized database’s replication options. The first column of the table contains the parameter name followed by its common description. The parameters are alphabetized according to the XML parameter name.

The third column of the following table contains XML text that you can paste into the Driver Parameters XML Editor. The XML text represents exactly what is necessary to display the parameters.

Table 4-8 *Rep Custom Parameters*

Parameter Name and Common Description	Description	XML to Define Driver Parameters
rep-abstract Rep Abstract	Specifies whether large documents should be truncated and attachments removed during replication. Set to True to truncate large documents and remove attachments and False to replicate large documents in their entirety.	As an attribute to an add, modify, or delete command element. Boolean: true | false
rep-background Background Replica Creation	Specifies if creating the synchronized database replica is performed via AdminP (in the background). When set to *True*, a replica is created via AdminP on the server(s) specified by the rep-new-server custom parameter. Example: rep-background=”true”.	As an attribute to an add command element. Boolean: true | false

Parameter Name and Common Description	Description	XML to Define Driver Parameters
`rep-clear-history` Rep Clear History	Set to True to clear the replication history from the replication object, and set to False to affect nothing.	As an attribute to an `add`, `modify`, or `delete` command element. Boolean: true | false
`rep-cutoff-delete` Rep Cutoff Delete	Specifies whether documents that are older than the cutoff date should be automatically deleted. The cutoff date is determined by today's date minus the cutoff interval (`rep-cutoff-interval`). Set to True to automatically delete documents that are older than the cutoff date, and False to not delete old documents.	As an attribute to an `add`, `modify`, or `delete` command element. Boolean: true | false
`rep-cutoff-interval` Rep Cutoff Interval	Specifies the number of days after which documents are automatically deleted if the CutoffDelete property is set (see `rep-cutoff-delete`).	As an attribute to an `add`, `modify`, or `delete` command element. Integer: numeric value
`rep-dest-server` Rep Destination Server	Specifies the Domino destination server (When computer) of a replication entry (within the replication object). If specified, and the source server/destination server pair does not already exist within the replication entry list, a new replication entry is created. If this parameter is absent, the destination server defaults to "any server" (a hyphen "-" means any server). To be used in connection with `rep-src-server`. Only available on Domino 6.0.3 or higher.	As an attribute to an `add`, `modify`, or `delete` command element. String: distinguished name of a replica Domino destination server, such as `CN=server1/O=acme`.
`rep-disabled` Rep Disabled	Specifies whether replication is disabled. Set to True to disable replication, and False to enable replication.	As an attribute to an `add`, `modify`, or `delete` command element. Boolean: true | false
`rep-dont-send-local-security-updates` Rep Don't Sent Local Security Updates	Specifies whether local security (encryption) updates are sent. Set to True to not send local security updates, and False to send local security updates. Only available on Domino 6.0.3 or higher.	As an attribute to an `add`, `modify`, or `delete` command element. Boolean: true | false
`rep-entry-remove` Rep Entry Remove	Set to True to remove the current replication entry from the replication object, and set to False to affect nothing. Used in connection with `rep-src-server` and/or `rep-dest-server`.	As an attribute to an `add`, `modify`, or `delete` command element. Boolean: true | false

Parameter Name and Common Description	Description	XML to Define Driver Parameters
`rep-formula` Rep Formula	Specifies the replication formula for a replication entry. By default a new replication entry contains the @All formula. The formula must be a valid replication formula. Used in connection with `rep-src-server` and/or `rep-dest-server`. Only available on Domino 6.0.3 or higher.	As an attribute to an `add`, `modify`, or `delete` command element String: replication formula
`rep-include-acl` Rep Include ACL	Specifies the inclusion of the ACL during replication for a replication entry. Set to True to include the ACL, False to exclude the ACL. Used in connection with `rep-src-server` and/or `rep-dest-server`. Only available on Domino 6.0.3 or higher.	As an attribute to an `add`, `modify`, or `delete` command element. Boolean: true | false
`rep-include-agents` Rep Include Agents	Specifies the inclusion of agents during replication for a replication entry. Set to True to include agents, False to exclude agents. Used in connection with `rep-src-server` and/or `rep-dest-server`. Only available on Domino 6.0.3 or higher.	As an attribute to an `add`, `modify`, or `delete` command element. Boolean: true | false
`rep-include-documents` Rep Include Documents	Specifies the inclusion of documents during replication for a replication entry. Set to True to include documents, False to exclude documents. Used in connection with `rep-src-server` and/or `rep-dest-server`. Only available on Domino 6.0.3 or higher.	As an attribute to an `add`, `modify`, or `delete` command element. Boolean: true | false
`rep-include-forms` Rep Include Forms	Specifies the inclusion of forms during replication for a replication entry. Set to True to include forms, False to exclude forms. Used in connection with `rep-src-server` and/or `rep-dest-server`. Only available on Domino 6.0.3 or higher.	As an attribute to an `add`, `modify`, or `delete` command element. Boolean: true | false
`rep-include-formulas` Rep Include Formulas	Specifies the inclusion of formulas during replication for a replication entry. Set to True to include formulas, False to exclude formulas. Used in connection with `rep-src-server` and/or `rep-dest-server`. Only available on Domino 6.0.3 or higher.	As an attribute to an `add`, `modify`, or `delete` command element. Boolean: true | false
`rep-ignore-deletes` Rep Ignore Deletes	Specifies whether outgoing deletions should not be replicated to other databases. Set to True to not replicate outgoing deletions, and False to replicate outgoing deletions.	As an attribute to an `add`, `modify`, or `delete` command element. Boolean: true | false
`rep-ignore-dest-deletes` Rep Ignore Destination Deletes	Specifies whether incoming deletions should not be replicated to the current database. Set to True to not replicate incoming deletions, and False to replicate incoming deletions.	As an attribute to an `add`, `modify`, or `delete` command element. Boolean: true | false

Parameter Name and Common Description	Description	XML to Define Driver Parameters
`rep-immediate` Rep Immediate	Indicates that database replication must begin immediately to the specified Domino server on which a existing database replica already exists.	As an attribute to an `add`, `modify`, or `delete` command element. String: distinguished name of a replica Domino destination server (such as `CN=server1/Acme`).
`rep-new-db-name` Rep New Database Name	The filename of the newly created replica. If `rep-new-db-name` is not present, then the filename of the original database is used. The default location of the new file is in the Domino server's data folder. Used in connection with `rep-new-server`.	As an attribute to an `add`, `modify`, or `delete` command element. String: file name of the new replica, such as `mail/JohnDoeRep2.nsf`.
`rep-new-server` Rep New Server	The name of the Domino server where a new replica will be created. The Domino server must be accessible on the network. Depending on the size of the database, this may be a time-consuming process for the NotesDriverShim. Can be used in connection with `rep-new-db-name`.	As an attribute to an `add`, `modify`, or `delete` command element. String: distinguished name of Domino server where a new replica will be created, such as `CN=server1/O=acme`.
`rep-priority` Rep Priority	Specifies the replication priority. Default is Medium (`MED`).	As an attribute to an `add`, `modify`, or `delete` command element. String: `HIGH, MED, LOW`
`rep-src-server` Rep Source Server	Specifies the Domino source server (Receives from) of a replication entry (within the replication object). If specified, and the source server/destination server pair does not already exist within the replication entry list, a new replication entry is created. If this parameter is absent, the source server defaults to "any server" (a hyphen "-" means any server). To be used in connection with `rep-dest-server`. Only available on Domino 6.0.3 or higher.	As an attribute to an `add`, `modify`, or `delete` command element. String: distinguished name of a replica Domino source server, such as `CN=server2/O=acme`.
`rep-view-list` Rep View List	Specifies a specific list of view names be replicated for a replication entry. The string specifies the views as a list, separating view names with semicolons, such as Inbox; Sent; Calendar; Meetings. Used in connection with `rep-src-server` and/or `rep-dest-server`. Only available on Domino 6.0.3 or higher.	As an attribute to an `add`, `modify`, or `delete` command element. String: ViewList

4.6 Example Files

You can configure and create rules, policies, and style sheets using iManager. (Style sheets are XSLT documents that define transformations or modifications of XML documents.)

Identity Manager provides the following examples:

- **NotesMoveSample.xml:** This sample policy is a Publisher channel policy that contains logic to determine eDirectory object placement when an associated Notes object is moved.

  In the Import Drivers Wizard, this policy is named Notes - Move Sample and is available under the Additional Policies heading. See Section 4.1, "Determining eDirectory Object Placement When a Notes Object is Moved," on page 49.
- **NotesReturnEmail.xml:** This sample policy is a Command Transformation policy designed to generate an e-mail address for user Add events on the Subscriber channel.

  It is necessary only when upgrading the driver shim and configuration from 1.*x* to 3. The policy is already a part of the sample configuration provided with the 2.1 version of the driver and later.

  In the Import Drivers Wizard, this policy is named Notes - Return Email Address and is available under the Additional Policies heading. See "Importing a Policy to Write Back the Notes E-Mail Address for New Users" on page 47.
- **Cert.xsl:** An Output Transformation style sheet that contains logic to determine which Notes certifier to use based on the src-dn attribute on the `add` XML attribute.

  See Section 4.2, "Automatically Determining Which Certifier to Use," on page 51 for more information.
- **Override.xsl.** Shows an example of how to use attributes to override parameters. See the list in Table 4-4 on page 65.
- **Placemove.xsl:** An Input Transformation style sheet that contains logic to determine placement containment when synchronizing a move from Lotus Notes to the Identity Vault.

  See Section 4.1, "Determining eDirectory Object Placement When a Notes Object is Moved," on page 49 for more information.
- **AddUniqueName.xsl.** Simple example of how a unique name can be created for a Notes user.
- **EntitlementGrpCmdCompletionSS.xsl.** If you choose to use Role-Based Entitlements when importing the sample configuration, this style sheet is included. This is an example of how to process the payload of an `operation-data` element.
- **NotesCertifierSelectionSampleSS.xsl.** Based on `Cert.xsl`, this shows an enhanced sample of how to utilize multiple Notes certifiers. It demonstrates using named passwords in multiple ways. See Section 4.2, "Automatically Determining Which Certifier to Use," on page 51 and Section 4.3, "Using Named Passwords," on page 52.

NOTE: Most of these are located in the product distribution in `nt/dirxml/drivers/lotusNotes/rules`. Some of them are used in the sample driver configuration.

4.7 Synchronizing a Database Other Than Names.nsf

Although the driver is intended as a directory synchronization driver for the Notes directory, it is possible to configure the driver to use a Notes database other than `names.nsf`. In this case, you need to make sure that the Schema Mapping policy is appropriate for the schema in the target database.

4.8 Schema Mapping Type and Form

In a Notes names and address book, each document contains a *Type* field as well as a *Form* field. The *Type* field supports the LDAP Server on Notes by providing a class name. The *Form* field is a standard Notes document field that indicates which form should be used to display the document. The *Form* item is not required, and if it is not present, the Notes client uses a default form.

Identity Manager does not provide the ability to map a single DS attribute to multiple target application attributes. This means that the Schema Mapping policy can't be used to map the object class to Form and Type. To handle this, the Driver Configuration asks if the directory database is really a Notes directory. If it is, the class name on DSEntry (translated into the Notes namespace) is used as the value for Type.

The object-class attribute on the DSAttribute object can be used to update the Form item if specified in the Schema Mapping policy. This provides a way to set both of those attributes, as well as providing mappings to allow the Type and Form values to differ. If the Schema Mapping policy contains a mapping between an eDirectory attribute and Form, it might be necessary to translate the content of the eDirectory attribute. This can be done by using an Output Transform policy. Conversely, an Input Transform policy is used to translate content from the Notes namespace to the eDirectory namespace.

If the directory source is not a Notes Directory, the driver writes no Type item and the Class Name attribute is written to the Form item. If a Form item appears in the filter, the driver and ndsrep ignore it.

If the driver is configured against the Notes directory, the translated values for classname are written to a Type item in the Notes database, and Form can be included in the Schema Mapping policy. If the driver is configured against a Notes database other than the directory, the translated values for classname are written to a Form item in the Notes database, and Form might not be included in the Schema Mapping policy.

4.9 Move/Rename

Move and Rename events are not supported in the default configuration. However, you can synchronize a Move or Rename event in Notes across the Publisher channel and into the Identity Vault if you modify the default Schema Map and the default Publisher Filter, and add a policy.

In addition, you can synchronize a Move or Rename event on the Subscriber channel if you have Notes 6.0.3 or later, enable AdminP support, and add policies that provide the necessary attributes.

- Section 4.9.1, “Subscriber Channel,” on page 99
- Section 4.9.2, “Publisher Channel,” on page 101
- Section 4.9.3, “Considerations for Using AdminP,” on page 101

4.9.1 Subscriber Channel

- “Moving a User” on page 100
- “Modifying a User Name in eDirectory (a Rename Event in Notes)” on page 100
- “Renaming a Group” on page 100

Moving a User

1. Make sure you are using Notes 6.0.3 or later and have reviewed Section 4.9.3, “Considerations for Using AdminP,” on page 101.
2. Make sure you have turned on support for the AdminP process by adding the following parameter to the Subscriber Options in the driver parameters:

   ```
   <allow-adminp-support display-name=“Allow Domino AdminP
   Support”>True</allow-adminp-support>
   ```

   See Allow Domino AdminP Support in Section 4.4.2, “Subscriber Options,” on page 55.
3. Create driver policies that add the following attributes to the Move event:
   - The certifier name of the destination certifier in Notes.
   - The certifier ID and a password for the destination certifier in Notes (the certifier that the user is going to)
   - The old certifier ID and password for the source certifier in Notes (the certifier that the user is coming from)

   A sample of a command to the driver shim that moves a user is included in Section D.3, “Sample of Moving a User,” on page 188.

Modifying a User Name in eDirectory (a Rename Event in Notes)

When a user's given name, middle initial, or surname changes in eDirectory, this event can cause the Rename of an object in Lotus Notes. If you have Notes 6.0.3 or later with AdminP support enabled, you can perform the Rename in Notes.

1. Make sure you are using Notes 6.0.3 or later and have reviewed Section 4.9.3, “Considerations for Using AdminP,” on page 101.
2. Make sure you have turned on support for the AdminP process by adding the following parameter to the Subscriber Options in the driver parameters:

   ```
   <allow-adminp-support display-name=“Allow Domino AdminP
   Support”>True</allow-adminp-support>
   ```

   See Allow Domino AdminP Support in Section 4.4.2, “Subscriber Options,” on page 55.
3. Create driver policies that provide the correct certifier and password for the Notes user that is being renamed.

   If a certifier and password are not specified in the event, the default certifier and password specified in the driver parameters are used.

   A sample of a command to the driver shim that renames a user is included in Section D.2, “Sample of Renaming: Modifying a User Last Name,” on page 187.

Renaming a Group

If you have Notes 6.0.3 or later with AdminP support enabled, you can rename groups. Rename events from eDirectory for groups do not require you to create any additional driver policies.

Rename events from eDirectory can be applied only to group objects in Notes. (For users, the driver shim uses an appropriate modify event to rename a user in Notes, as described in “Modifying a User Name in eDirectory (a Rename Event in Notes)” on page 100.)

1. Make sure you are using Notes 6.0.3 or later and have reviewed Section 4.9.3, “Considerations for Using AdminP,” on page 101.

2. Make sure you have turned on support for the AdminP process by adding the following parameter to the Subscriber Options in the driver parameters:

   ```
   <allow-adminp-support display-name=“Allow Domino AdminP
   Support”>True</allow-adminp-support>
   ```

   See Allow Domino AdminP Support in Section 4.4.2, “Subscriber Options,” on page 55.

4.9.2 Publisher Channel

To enable the one-way object move/rename synchronization:

1. Modify the schema mapping to map eDirectory Full Name to Notes FullName.
2. Enable the Full Name attribute in the Publisher filter.
3. Make sure that the Full Name attribute in the Subscriber filter is not enabled.
4. Make sure that the Public/Private AB setting is `Yes` in the driver configuration parameters.
5. Use a policy in your driver configuration like the one described in Section 4.1, “Determining eDirectory Object Placement When a Notes Object is Moved,” on page 49.

After these modifications have been made, ndsrep detects changes to FullName. Because FullName contains both name and location information in a single attribute, ndsrep cannot distinguish between a Move and a Rename. Therefore, a change to FullName initiates both a Move and a Rename event to be synchronized into eDirectory.

4.9.3 Considerations for Using AdminP

AdminP support provides several new features, but to use them effectively you must keep in mind the following:

- You need to have an understanding of AdminP and of Notes administration.
- A success message returned to the driver for an AdminP request means only that the request was successfully received by AdminP, not that it was completed successfully.
- AdminP requests made by the driver are not completed until AdminP attempts the action. The timing depends on the configuration of the Administration Process by the Notes administrator, the Domino server network, and the complexity of the action requested.
- Some AdminP requests require manual approval by the Notes administrator before they are completed.
- AdminP requests typically include the FullName of the Notes user (or ListName for a group). The driver sends requests based on the FullName of the user at the time the request was initiated, but AdminP does not necessarily complete the request immediately, and other requests that affect the FullName of the same user object might be waiting to be processed. If the FullName of the user is changed by a request, a subsequent request might fail because AdminP can't find the user.

  For example, consider this scenario:

  - You send a request from the driver to change a user's first name in Notes, and you use the AdminP feature to also rename the user object (changing FullName).
  - You immediately send a second request from the driver to change the same user's last name in Notes and also rename the user object (changing FullName).

Both requests are received by AdminP. Both requests refer to the user with the same FullName. At midnight, AdminP begins processing the requests. The first one succeeds. However, the second one fails because the FullName was changed by the first request.

To help you use AdminP effectively, the following features are provided:

- You can cause the driver to send commands directly to the Domino Console. For example, you can issue a command to process all AdminP requests immediately. See Section 4.10, "TELL AdminP Commands," on page 102, and Domino Console Command in Section 4.5, "Custom Driver Parameters," on page 64.
- You can enable or disable AdminP support for an individual command. See Allow AdminP Support in Section 4.5, "Custom Driver Parameters," on page 64.

4.10 TELL AdminP Commands

When the driver issues a request to the Domino AdminP process, these requests are delayed until the AdminP process completes them. (Refer to Notes documentation for information about administration processing intervals.)

If desired, you can attach a `tell-adminp-process` attribute to an event. If the event contains AdminP tasks that need to be performed, the command you specify is sent to the Domino server console. The attribute is described in Domino Console Command in Section 4.5, "Custom Driver Parameters," on page 64.

For example, when sending a Move user event, you could include the following attribute on the move event:

```
tell-adminp-process="tell adminp process new"
```

This example command causes the driver to request the AdminP process to process all the new tasks, which would include the move that was requested in that event.

To use `tell-adminp-process` commands:

1. Make sure you are using Notes 6.0.3 or later.
2. Make sure you have turned on support for the AdminP process by adding the following parameter to the Subscriber Options in the driver parameters:

   ```
   <allow-adminp-support display-name="Allow Domino AdminP
   Support">True</allow-adminp-support>
   ```

   See Allow Domino AdminP Support in Section 4.4.2, "Subscriber Options," on page 55.
3. Make sure the Notes user for the driver has rights to send commands to the Domino server console.
4. Make sure that the event contains AdminP tasks.

   The `tell-adminp-process` command is sent only if AdminP tasks need to be performed as part of the event.
5. Use the correct syntax.

   Samples of using `tell-adminp-process` commands are included in Appendix D, "Sample XDS Documents for Custom Parameters," on page 185.
6. To find out whether the AdminP request was completed successfully, use Lotus Notes tools such as the Domino Administrator.

Completion of a command by the Notes driver shim involving an AdminP request does not mean that the command has been successfully completed. It means only that the request has been made to AdminP.

For example, the driver might successfully make a request to AdminP to move a user. However, if incorrect certifiers are specified in the event, the move would fail when the AdminP process attempts it.

4.11 User.id Password Set

The NotesDriverShim provides password-set capabilities through the `modify-password` command. When issuing the command on the Subscriber channel, the NotesDriverShim uses a shared native library to access the appropriate Lotus Notes APIs that allow for changing a password within a `user.id` file.

The shared native library that comes with the driver is named `notesdrvjni`. On the Windows platform, `notesdrvjni.dll` is placed in the Identity Manager binaries folder (`c:\novell\nds`) where the `NotesDriverShim.jar` can find it when the driver launches. On the AIX, Linux, and Solaris platforms, the `notesdrvjni.so` file is linked to the `Notes/Domino` execution directory.

If the NotesDriverShim initializes with the `notesdrvjni` shared library present, then text similar to the following should show in a level 3 (or above) trace:

```
"NotesDriverShimLotusCAPIAccess: notesdrvjni (notesdrvjni.dll)
successfully loaded."
```

If this text does not appear, or if you see a different message, then the feature is automatically disabled within the NotesDriverShim.

Upon receiving the `modify-password` command, the NotesDriverShim attempts to modify the Notes `user.id` file passwords if the `modify-password` command has the following elements:

- An old-password
- A password element
- A `user-id-file="c:\Lotus\Domino\ids\people\JohnDoe.id"` XML attribute as a custom parameter

This custom parameter is the same as that used for `add` commands, but in this case the specified filename is never used in conjunction with the `user-id-path` parameter because it is with an `add` command. If a full path is not specified, the NotesDriverShim directs the `user.id` file search to the default directory that the driver is using (the `Notes/Domino` data folder).

This `user-id-file` parameter can also be used to specify multiple ID files by separating the ID filenames with semicolons (";"). (such as `user-id file="ids\people\JohnDoe.id;ids\people\johnnydoe.id;c:\Temp\jd.id"`.

ID files that require multiple passwords cannot be set by using this feature. However, it is possible to use this feature to change the password for a `user.id` file, a `server.id` file, or a `cert.id` file. Below is a general example of an XDS command setting a password for `JohnDoe.id`:

```
<input>
   <modify-password
      class-name="Person"
      event-id="pwd-subscribe"
      src-dn="\PWDSYNCTREE\sync\dom\unit\JohnDoe"
      src-entry-id="35952"
      user-id-file="c:\Lotus\Domino\ids\people\JohnDoe.id">
      <association>D9628831A988381AC12570F9005BE6B3</association>
      <old-password>zyxwvut321</old-password>
      <password>abcdefg123</password>
    </modify-password>
</input>
```

The NotesDriverShim must have appropriate read and write file system access to the specified `user.id` file. When a correct `user-id-file="user.id"` XML attribute and valid `<old-password>` and `<password>` elements are available to the `modify-password` command, this command then modifies an existing `user.id` file password, as well as the HTTPPassword within the NAB.

You can control whether the NotesDriverShim modifies the `user.id` password, and/or the HTTPPassword by using the following driver parameters that you set within subscriber-options section of the driver configuration:

`allow-http-password-set allow-userid-password-set`

These subscriber-options parameters are Boolean and can be set to True or False, depending on the desired default behavior of the NotesDriverShim. They can also be used as override parameters when you set them as attributes to the `<modify-password>` element.

If you only want to modify the HTTPPassword, omit the `user-id-file` attribute from the XDS command, or apply the `allow-userid-password-set="false"` attribute. If you only want to modify the `user.id` password, then apply the `allow-http-password-set="false"` attribute.

Activating the Driver

5

Novell® Identity Manager, Integration Modules, and the Provisioning Module must be activated within 90 days of installation, or they shut down. At any time during the 90 days, or afterward, you can choose to activate Identity Manager products.

To activate the driver, see "Activating Novell Identity Manager Products" in the *Identity Manager 3.5.1 Installation Guide*.

Managing the Driver

6

The driver can be managed through Designer, iManager, or the DirXML® Command Line utility.

- Section 6.1, "Starting, Stopping, or Restarting the Driver," on page 107
- Section 6.2, "Migrating and Resynchronizing Data," on page 108
- Section 6.3, "Using the DirXML Command Line Utility," on page 108
- Section 6.4, "Viewing Driver Versioning Information," on page 109
- Section 6.5, "Reassociating a Driver Set Object with a Server Object," on page 113
- Section 6.6, "Changing the Driver Configuration," on page 113
- Section 6.7, "Storing Driver Passwords Securely with Named Passwords," on page 114
- Section 6.8, "Adding a Driver Heartbeat," on page 120

6.1 Starting, Stopping, or Restarting the Driver

- Section 6.1.1, "Starting the Driver in Designer," on page 107
- Section 6.1.2, "Starting the Driver in iManager," on page 107
- Section 6.1.3, "Stopping the Driver in Designer," on page 107
- Section 6.1.4, "Stopping the Driver in iManager," on page 107
- Section 6.1.5, "Restarting the Driver in Designer," on page 108
- Section 6.1.6, "Restarting the Driver in iManager," on page 108

6.1.1 Starting the Driver in Designer

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

6.1.2 Starting the Driver in iManager

1. In iManager, click *Identity Manager* > *Identity Manager Overview*.
2. Browse to the driver set where the driver exists, then click *Search*.
3. Click the upper right corner of the driver icon, then click *Start driver*.

6.1.3 Stopping the Driver in Designer

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

6.1.4 Stopping the Driver in iManager

1. In iManager, click *Identity Manager* > *Identity Manager Overview*.

2 Browse to the driver set where the driver exists, then click *Search*.

3 Click the upper right corner of the driver icon, then click *Stop driver*.

6.1.5 Restarting the Driver in Designer

1 Open a project in the Modeler, then right-click the driver line.

2 Select *Live* > *Restart Driver*.

6.1.6 Restarting the Driver in iManager

1 In iManager, click *Identity Manager* > *Identity Manager Overview*.

2 Browse to the driver set where the driver exists, then click *Search*.

3 Click the upper right corner of the driver icon, then click *Restart driver*.

6.2 Migrating and Resynchronizing Data

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

- **Migrate Data from Identity Vault:** Allows you to select containers or objects you want to migrate from the Identity Vault to an application. When you migrate an object, the Identity Manager engine applies all of the Matching, Placement, and Create policies, as well as the Subscriber filter, to the object.
- **Migrate Data into Identity Vault:** Assumes that the remote application (usually a Web Service) can be queried for entries that match the criteria in the publisher filter. However, because of the general nature of the PeopleSoft driver the method for querying the Web Service (if there is one) is not known to the driver shim. Therefore, this feature does not usually work with the PeopleSoft driver.
- **Synchronize:** The Identity Manager engine 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 explained above:

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 .

6.3 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 B, “DirXML Command Line Utility,” on page 147 for detailed information about the DirXML Command Line utility. For daily tasks, use iManager or Designer.

6.4 Viewing Driver Versioning Information

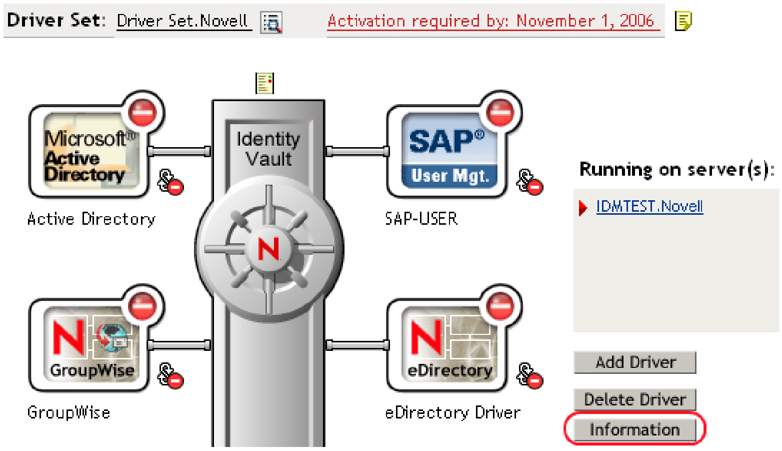
The Versioning Discovery tool only exists in iManager.

- Section 6.4.1, “Viewing a Hierarchical Display of Versioning Information,” on page 109
- Section 6.4.2, “Viewing the Versioning Information As a Text File,” on page 110
- Section 6.4.3, “Saving Versioning Information,” on page 112

6.4.1 Viewing a Hierarchical Display of Versioning Information

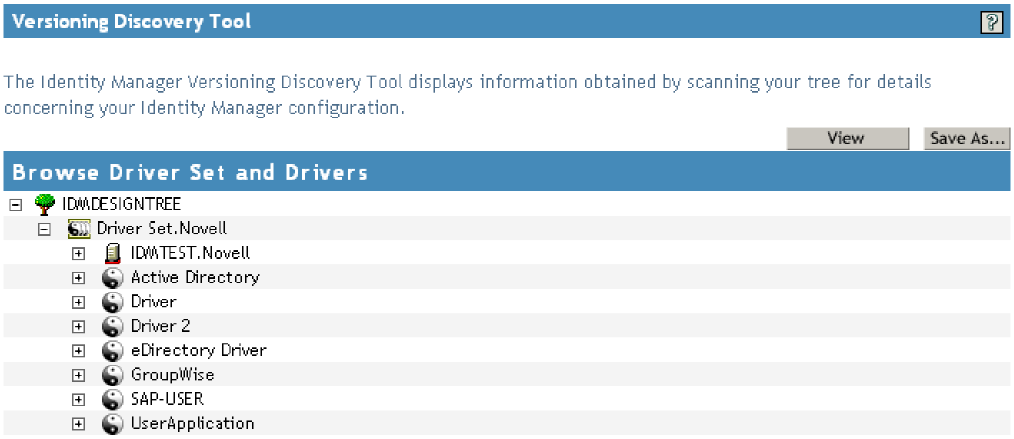
1 To find your Driver Set object in iManager, click *Identity Manager* > *Identity Manager Overview*, then click *Search*.

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



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

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



Versioning Discovery Tool

The Identity Manager Versioning Discovery Tool displays information obtained by scanning your tree for details concerning your Identity Manager configuration.

View Save As...

Browse Driver Set and Drivers

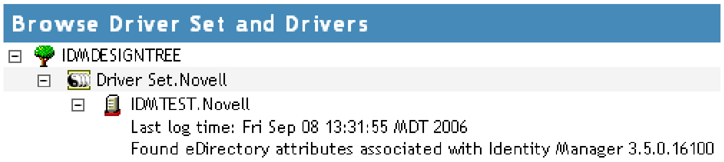
```
⊟ IDMDESIGNTREE
   ⊟ Driver Set.Novell
      ⊞ IDMTEST.Novell
      ⊞ Active Directory
      ⊞ Driver
      ⊞ Driver 2
      ⊞ eDirectory Driver
      ⊞ GroupWise
      ⊞ SAP-USER
      ⊞ UserApplication
```

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 versioning information related to servers by expanding the server icon.



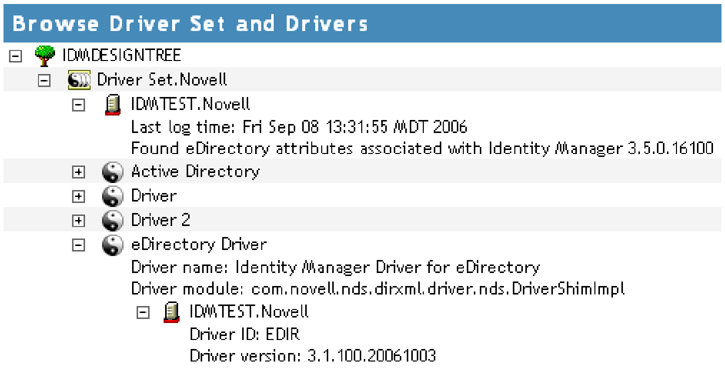
Browse Driver Set and Drivers

```
⊟ IDMDESIGNTREE
   ⊟ Driver Set.Novell
      ⊟ IDMTEST.Novell
         Last log time: Fri Sep 08 13:31:55 MDT 2006
         Found eDirectory attributes associated with Identity Manager 3.5.0.16100
```

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 versioning information related to drivers by expanding the driver icon.



Browse Driver Set and Drivers

```
⊟ IDMDESIGNTREE
   ⊟ Driver Set.Novell
      ⊟ IDMTEST.Novell
         Last log time: Fri Sep 08 13:31:55 MDT 2006
         Found eDirectory attributes associated with Identity Manager 3.5.0.16100
      ⊞ Active Directory
      ⊞ Driver
      ⊞ Driver 2
      ⊟ eDirectory Driver
         Driver name: Identity Manager Driver for eDirectory
         Driver module: com.novell.nds.dirxml.driver.nds.DriverShimImpl
         ⊟ IDMTEST.Novell
            Driver ID: EDIR
            Driver version: 3.1.100.20061003
```

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:

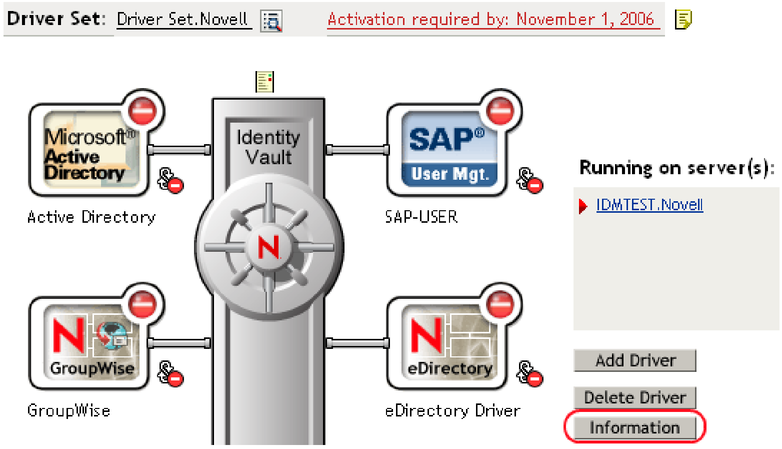
- The driver ID
- The version of the instance of the driver running on that server

6.4.2 Viewing the Versioning Information As a Text File

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

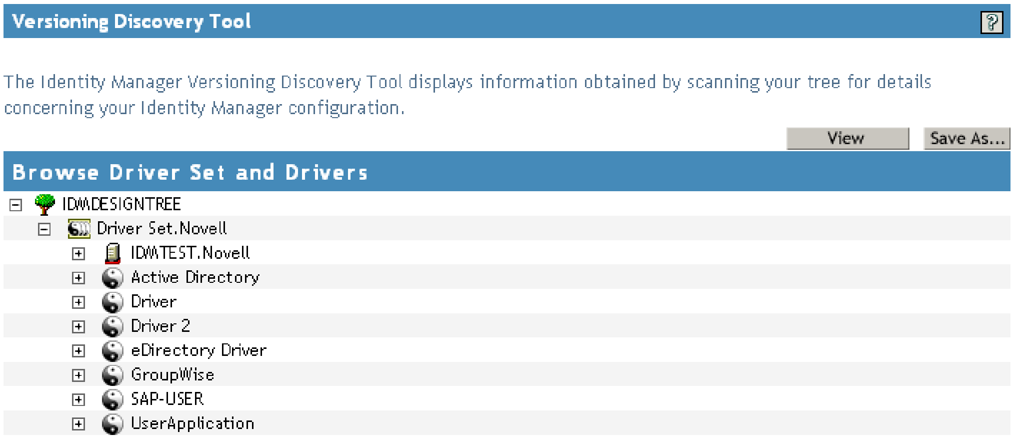
1 To find your Driver Set object in iManager, click *Identity Manager > Identity Manager Overview*, then click *Search*.

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



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

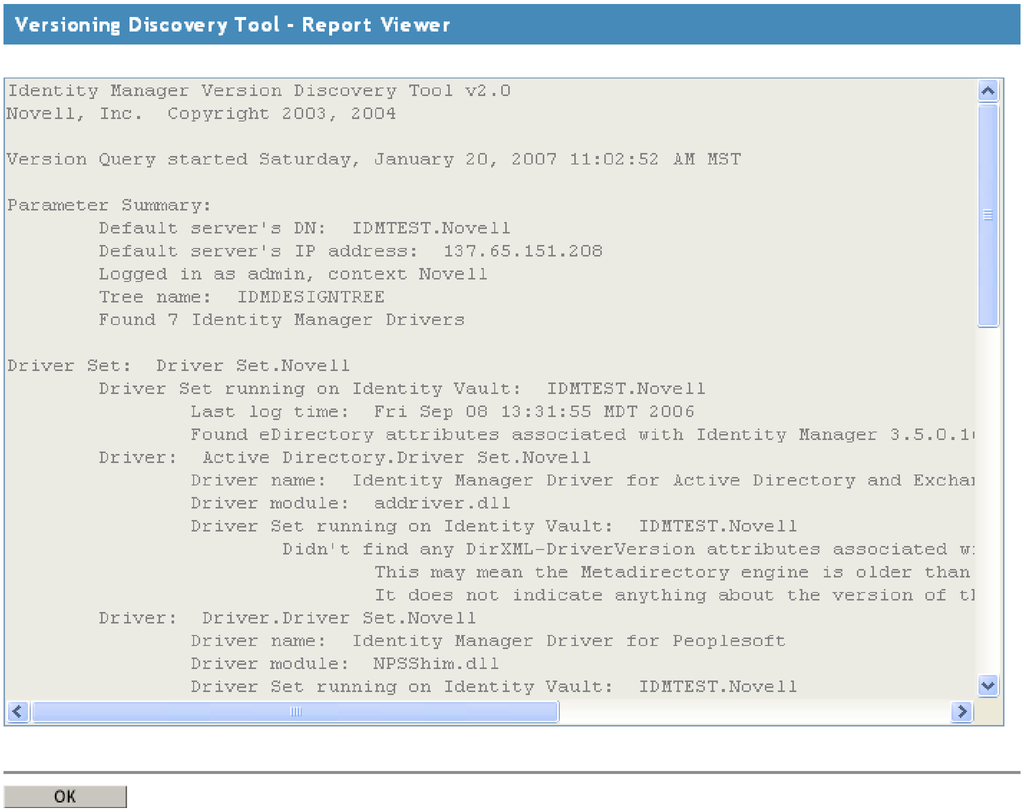
3 In the Versioning Discovery Tool dialog box, click *View*.



The Identity Manager Versioning Discovery Tool displays information obtained by scanning your tree for details concerning your Identity Manager configuration.

- IDMDESIGNTREE
  - Driver Set.Novell
    - IDMTEST.Novell
    - Active Directory
    - Driver
    - Driver 2
    - eDirectory Driver
    - GroupWise
    - SAP-USER
    - UserApplication

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



Versioning Discovery Tool - Report Viewer

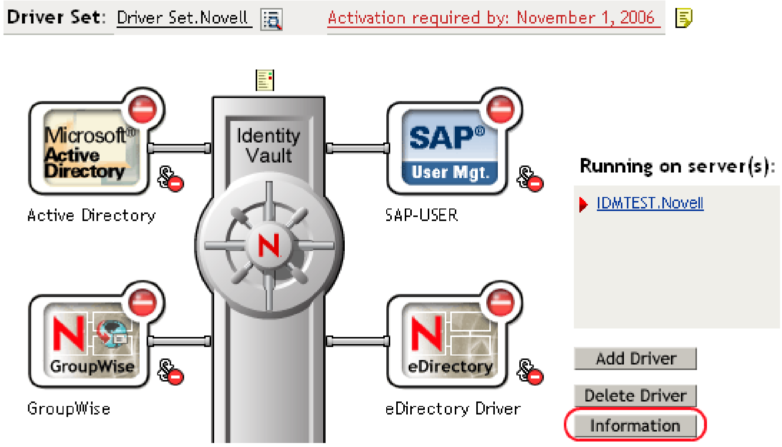
```
Identity Manager Version Discovery Tool v2.0
Novell, Inc.  Copyright 2003, 2004

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

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

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

6.4.3 Saving Versioning Information

You can save versioning 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.*

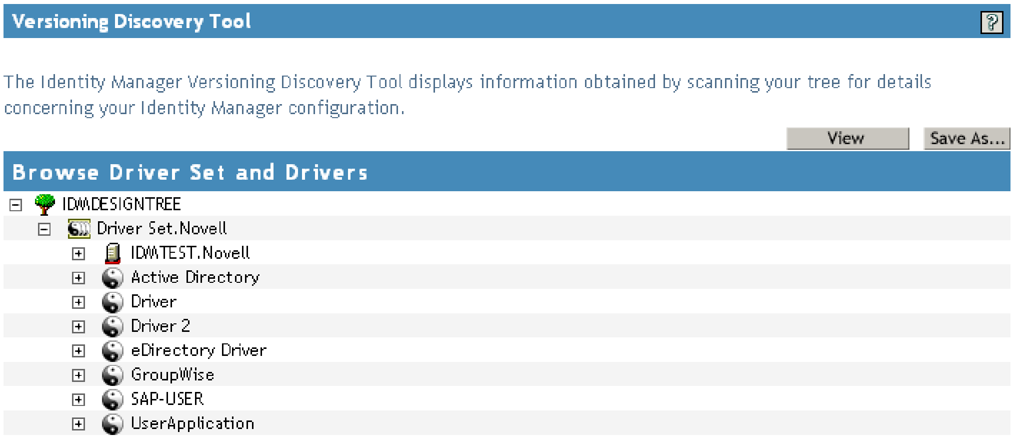


Driver Set: Driver Set.Novell

Activation required by: November 1, 2006

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

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



Versioning Discovery Tool

The Identity Manager Versioning Discovery Tool displays information obtained by scanning your tree for details concerning your Identity Manager configuration.

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.

6.5 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 is not associated with a server object, none of the drivers in the driver set can start.

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

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

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

1 In iManager click *Identity Manager > Identity Manager Overview*, then click *Search* to find the driver set object that the driver should be associated with.

2 Click the *Remove server* icon, then click *OK*.

3 Click the *Add server* icon, then browse to and select the server object.

4 Click *OK*.

6.6 Changing the Driver Configuration

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

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*.

To change the driver configuration in Designer:

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

For a listing of all of the configuration fields, see .

6.7 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.

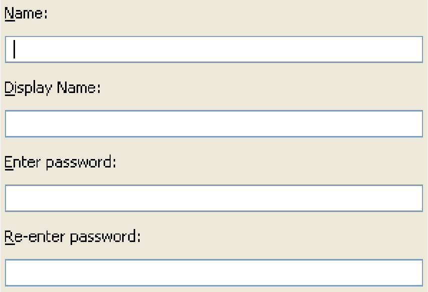
You can also use the Named Passwords feature to store other pieces of information securely, such as a user name.

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 6.7.1, "Using Designer to Configure Named Passwords," on page 114
- Section 6.7.2, "Using iManager to Configure Named Passwords," on page 115
- Section 6.7.3, "Using Named Passwords in Driver Policies," on page 116
- Section 6.7.4, "Using the DirXML Command Line Utility to Configure Named Passwords," on page 117

6.7.1 Using Designer to Configure Named Passwords

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



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.

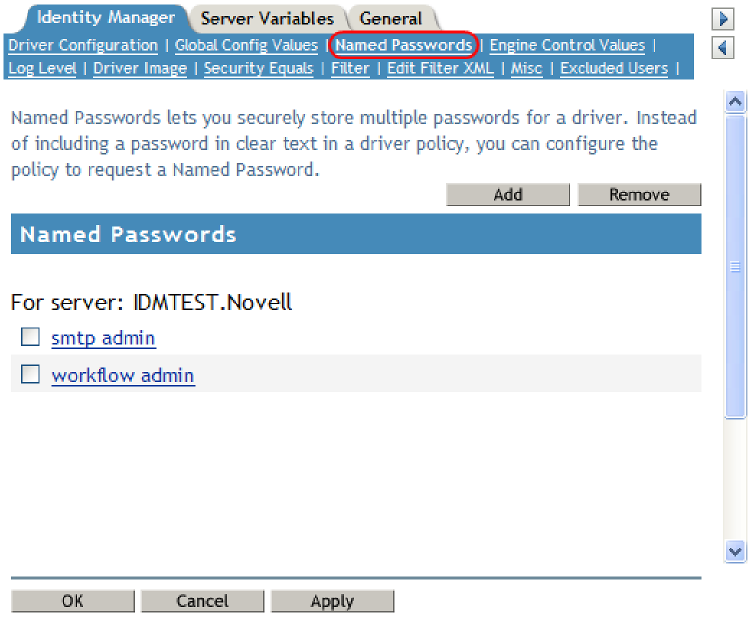
6.7.2 Using iManager to Configure Named Passwords

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

2 In the Identity Manager Overview, click the upper right corner of the driver icon, then click *Edit properties*.

3 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.


Identity Manager
Server Variables
General
Driver Configuration | Global Config Values | Named Passwords | Engine Control Values |
Log Level | Driver Image | Security Equals | Filter | Edit Filter XML | Misc | Excluded Users |

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.

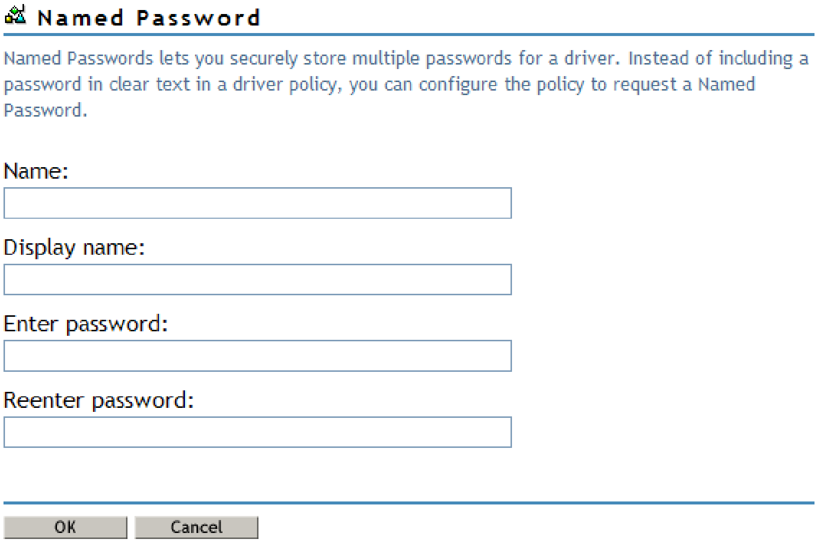
Named Passwords

For server: IDMTEST.Novell

□ smtp admin

□ workflow admin

4. 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:

5. 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.
6. Click *OK* to restart the driver and have the changes take effect.
7. To remove a Named Password, select the password name, then click *Remove*.

   The password is removed without prompting you to confirm the action.

6.7.3 Using Named Passwords in Driver Policies

- "Using the Policy Builder" on page 116
- "Using XSLT" on page 117

Using the Policy Builder

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.

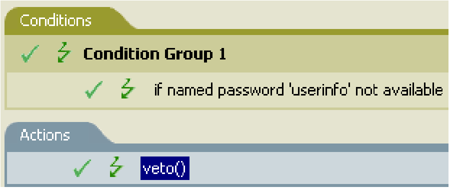
1. In Designer, launch Policy Builder, right-click, then click *New* > *Rule*.
2. Specify the name of the rule, then click *Next*.
3. Select the condition structure, then click *Next*.
4. Select *named password* for the *Condition*.
5. Browse to and select the Named Password that is stored on the driver.

   In this example, it 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 6-1 A Policy Using Named Passwords



Using XSLT

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/>
```

6.7.4 Using the DirXML Command Line Utility to Configure Named Passwords

- "Creating a Named Password in the DirXML Command Line Utility" on page 117
- "Using the DirXML Command Line Utility to Remove a Named Password" on page 118

Creating a Named Password in the DirXML Command Line Utility

1 Run the DirXML Command Line utility.

For information, see Appendix B, "DirXML Command Line Utility," on page 147.

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.

Using the DirXML Command Line Utility to Remove a Named Password

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 B, "DirXML Command Line Utility," on page 147.

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 passwor
 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, you can use the 99 option twice to exit the menu and quit the DirXML Command Line utility.

6.8 Adding a Driver Heartbeat

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

   The value of the interval cannot be less than 1. A value of 0 means 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 like the following example, as a child of `<publisher-options>`. (For an AD driver, make it a child of `<driver-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, and 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

7

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.

7.1 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.

7.2 When Is Synchronization Done?

The Metadirectory engine performs object synchronization or merging in the following circumstances:

- A `<sync>` event element is submitted on the Subscriber or Publisher channel.
- 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.
- 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. These <sync> events are submitted using the Subscriber thread, but are processed using the Publisher channel filter and policies.
- An <add> event (real or synthetic) is submitted on a channel and the channel Matching policy finds a matching object in the target system.
- 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.
- 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 7.3, "How Does the Metadirectory Engine Decide Which Object to Synchronize?," on page 124.

7.3 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. In DirXML 1.1a there is 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 driver cache for each unique object found that has an entry modification time stamp greater than or equal to the starting filter time and all objects and classes that are in the Subscriber filter channel in the driver being synchronized.

7.4 How Does Synchronization Work?

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 7-1 on page 126, Table 7-2 on page 127, and Table 7-3 on page 128.

   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.

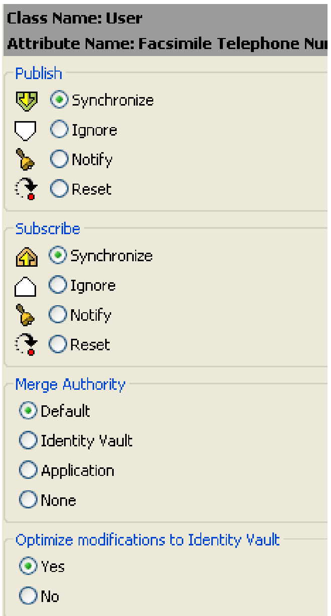
There are three different combinations of selected items in the filter, and each one creates a different output.

- Section 7.4.1, "Scenario One," on page 125
- Section 7.4.2, "Scenario Two," on page 127
- Section 7.4.3, "Scenario Three," on page 128

7.4.1 Scenario One

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

Figure 7-1 *Scenario One*



Class Name: User
Attribute Name: Facsimile Telephone Nu

Publish

● Synchronize
○ Ignore
○ Notify
○ Reset

Subscribe

● Synchronize
○ Ignore
○ Notify
○ Reset

Merge Authority

● Default
○ Identity Vault
○ Application
○ None

Optimize modifications to Identity Vault

● Yes
○ No

The following table 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 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.

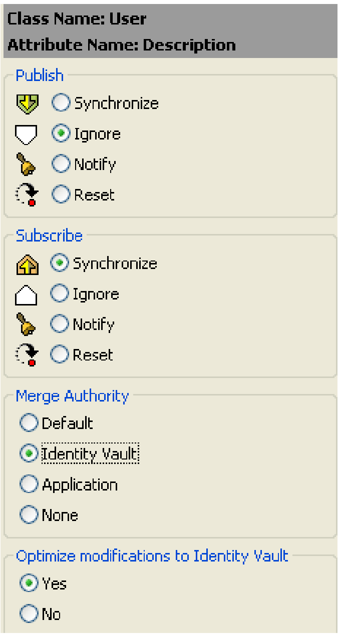
Table 7-1 *Output of Scenario One*

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

7.4.2 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 7-2 *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

The following table 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 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.

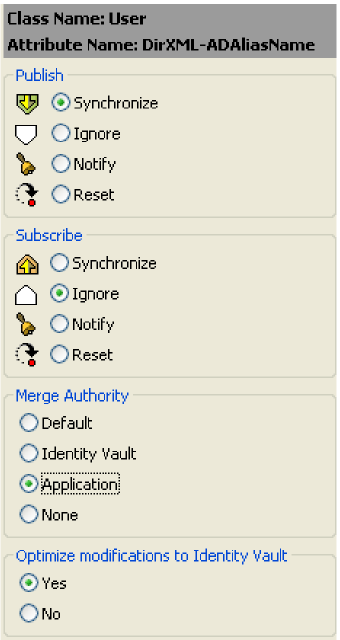
Table 7-2 *Output of Scenario Two*

	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

7.4.3 Scenario Three

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

Figure 7-3 *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

The following table 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 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.

Table 7-3 *Output of Scenario Three*

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

Troubleshooting the Driver 8

This section contains potential problems and error codes you might encounter while configuring or using the driver.

- Section 8.1, "Troubleshooting Driver Processes," on page 129
- Section 8.2, "Troubleshooting Lotus Notes-Specific Items," on page 135

8.1 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.

8.1.1 Viewing Driver Processes

In order to see the driver processes in DSTRACE, values are added to the driver set and the driver objects. You can do this in Designer and iManager.

- "Adding Trace Levels in Designer" on page 129
- "Adding Trace Levels in iManager" on page 131
- "Capturing Driver Processes to a File" on page 132

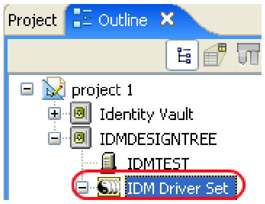
Adding Trace Levels in Designer

You can add trace levels to the driver set object or to each driver object.

- "Driver Set" on page 129
- "Driver" on page 130

Driver Set

1 In an open project in Designer, select the driver set object in the *Outline* view.



2 Right-click and 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. If you want 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 *Unlimited*, the file grows in size until there is no disk space left.

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 and 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. If you want to see password synchronization information, set the trace level to 5. if you select *Use setting from Driver Set*, 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 *Use setting from Driver Set*, the value is taken from the driver set object.

Parameter	Description
Trace file size limit	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. If you select *Use setting from Driver Set*, the value is taken from the driver set object. **NOTE:** The trace file is created in multiple files. Identity Manager automatically divides the maximum file size by ten and creates ten separate files. The combined size of these files equals the maximum trace file size.
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.

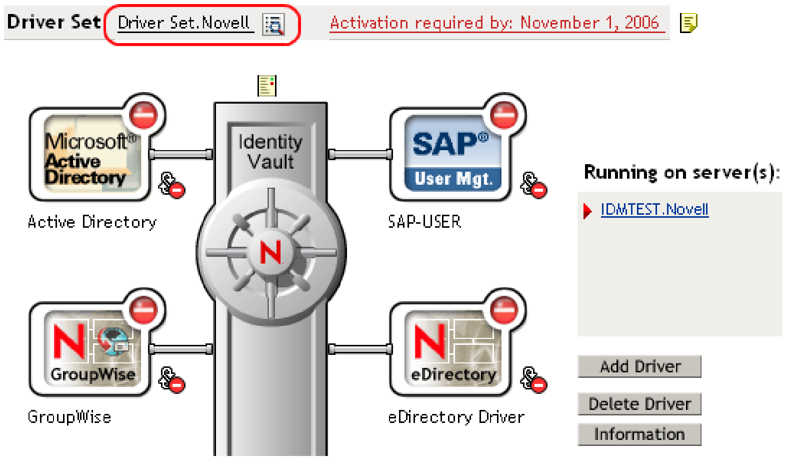
- "Driver Set" on page 131
- "Driver" on page 132

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*.

See "Misc" on page 177 for the parameters.

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*.

   See "Misc" on page 177 for the parameters.

   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 processed 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 helps you capture and save Identity Manager processes through DSTRACE on different platforms.

- "NetWare" on page 132
- "Windows" on page 133
- "UNIX" on page 133
- "iMonitor" on page 133
- "Remote Loader" on page 134

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 +dvrs` 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* > *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 their 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 8-1 *Command Line Tracing Switches*

Option	Short Name	Parameter	Description
-trace	-t	integer	Specifies the trace level. This is only used when hosting an application shim. Trace levels correspond to those used on the Identity Manager server. Example: `-trace 3` or `-t3`

Option	Short Name	Parameter	Description
-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: `-tracefile c:\temp\trace.txt` or `-tf c:\temp\trace.txt`
-tracefilemax	-tfm	size	Specifies the approximate maximum size that trace file data can occupy on disk. If you specify this option, there is a trace file with the name specified using the tracefile option and up to 9 additional “roll-over” files. The roll-over files are named using the base of the main trace filename plus “_n”, where n is 1 through 9. The size parameter is the number of bytes. Specify the size by using the suffixes K, M, or G for kilobytes, megabytes, or gigabytes. 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. Example: `-tracefilemax 1000M` or `-tfm 1000M`

8.2 Troubleshooting Lotus Notes-Specific Items

- Section 8.2.1, “Creating Lotus Notes Accounts and Groups,” on page 135
- Section 8.2.2, “Troubleshooting Installation Problems,” on page 136

8.2.1 Creating Lotus Notes Accounts and Groups

1 Create a Notes User ID to be used exclusively by the driver and give it manager-level ACL access to the target Notes database (usually `names.nsf`), the output database (`ndsrep.nsf`) created by ndsrep, and `certlog.nsf`. If you are synchronizing with the `names.nsf` database, you should select (turn on) all ACL roles (GroupCreator, GroupModifier, NetCreator, NetModifier, PolicyCreator, PolicyModifier, PolicyReader, ServerCreator, ServerModifier, UserCreator, UserModifier).

2 If a Deny Access group doesn’t already exist, create this group using the Lotus Domino Administrator tool. This group is used to hold disabled user accounts.

3 (Optional) Copy and save the Universal ID (UNID) for the Deny Access group you just created.

This string is used by the driver to uniquely identify this object, and you need to specify it when you are importing the driver configuration.

To get the string:

3a View the Document Properties of the group. (You can select the object and right-click to select Document Properties.)

3b Click the Meta tab (the fifth tab from the right).

3c Go to the end of the text in the `Identifier` field, and copy the character string from the last forward slash to the end. This will always be 32 alphanumeric characters.

For example, if text in the `Identifier` field is

```
Notes://myserver/87256E530082B5F4/
85255E01001356A8852554C200753106/
16A28402CCEB7A9C87256E9F007EDA9B
```

then the UNID would be

```
16A28402CCEB7A9C87256E9F007EDA9B
```

3d Paste this information into a file for later use when you run the Create Driver Wizard, as noted in Section 2.5, "Importing the Notes Driver Configuration File in iManager," on page 24 or Section 2.6, "Installing the Lotus Notes Driver through Designer," on page 25.

8.2.2 Troubleshooting Installation Problems

For Windows:

- The first time the driver runs, it searches for the Domino Server (specified in the driver parameters at import time), and tries to open `dsrepcfg.nsf` to write the publisher parameters that ndsrep reads. If `dsrepcfg.nsf` does not exist, the NotesDriverShim attempts to create `dsrepcfg.nsf` using the database template `dsrepcfg.ntf` that ships with the driver.

  If `dsrepcfg.nsf` is successfully created, and contains data specifying an appropriate update database file (usually named `ndsrep.nsf`), then you can load ndsrep successfully at the Domino Console.

  If `dsrepcfg.ntf` is not found, or this initial `dsrepcfg.nsf` creation process fails, then the Publisher channel shuts down, and you can't load the ndsrep task at the Domino console.

  You can use a Notes client to create the `dsrepcfg.nsf` database using the `dsrepcfg.ntf` template. After doing so, modify the ACL so that the Notes driver user has manager-level access to the database.

For AIX, Linux, and Solaris:

- If you upgrade Domino after installing the driver, you need to do one of the following:
  - Check the symbolic links, and re-create them manually if necessary. Use Table 8-2 on page 136 if your Domino server is prior to V7 running eDirectory 8.8. Use Table 8-3 on page 137 if your Domino server is V7.x running eDirectory 8.8.

Table 8-2 *Links To Check for Domino V6.x with eDirectory 8.7*

File to link	Symbolic link to create
`Notes.jar`	`/usr/lib/dirxml/classes/Notes.jar` Example: ln -s `/opt/lotus/notes/latest/your_platform/Notes.jar /user/lib/dirxml/classes/Notes.jar`

File to link	Symbolic link to create
ndsrep	/opt/lotus/notes/latest/*your_platform*/ndsrep Example: ln -s /usr/lib/dirxml/rules/notes/ndsrep /opt/lotus/notes/latest/*your_platform*/ndsrep
dsrepcfg.ntf	/opt/lotus/notes/latest/*your_platform*/dsrepcfg.ntf Example: ln -s /usr/lib/dirxml/rules/notes/dsrepcfg.ntf /opt/lotus/notes/latest/*your_platform*/dsrepcfg.ntf
libnotesdrvjni.so	/opt/lotus/notes/latest/your_platform/ libnotesdrvjni.so.1.0.0 Example on AIX/SolarisLinux: cp /usr/lib/dirxml/rules/notes/ libnotesdrvjni.so.1.0.0 /opt/lotus/notes/latest/ your_platform/libnotesdrvjni.so.1.0.0 ln -s /usr/lib/dirxml/rules/notes/ libnotesdrvjni.so.1.0.0 /usr/lib/dirxml/rules/ notes/libnotesdrvjni.so

Table 8-3 *Links To Check for Domino V7.x with eDirectory 8.8*

File to link	Symbolic link to create
Notes.jar	/opt/novell/eDirectory/lib/dirxml/classes/ Notes.jar Example: ln -s /opt/ibm/lotus/notes/latest/your_platform/jvm/ lib/ext/Notes.jar /opt/novell/eDirectory/lib/dirxml/classes/ Notes.jar
ndsrep	/opt/ibm/lotus/notes/latest/*your_platform*/ndsrep Example: ln -s /usr/lib/dirxml/rules/notes/ndsrep /opt/ibm/lotus/notes/latest/*your_platform*/ndsrep
dsrepcfg.ntf	/opt/ibm/lotus/notes/latest/*your_platform*/ dsrepcfg.ntf Example: ln -s /usr/lib/dirxml/rules/notes/dsrepcfg.ntf /opt/ibm/lotus/notes/latest/*your_platform*/ dsrepcfg.ntf

File to link	Symbolic link to create

`libnotesdrvjni.so`	`/opt/ibm/lotus/notes/latest/your_platform/libnotesdrvjni.so.1.0.0`
	Example on AIX/Solaris/Linux and eDirectory 8.8:
	`cp /opt/novell/eDirectory/lib/dirxml/rules/notes/libnotesdrvjni.so.1.0.0 /opt/ibm/lotus/notes/latest/your_platform/libnotesdrvjni.so.1.0.0`
	`ln -s /opt/novell/eDirectory/lib/dirxml/rules/notes/libnotesdrvjni.so.1.0.0 /opt/novell/eDirectory/lib/dirxml/rules/notes/libnotesdrvjni.so`

The variable *your_platform* represents the operating system. The following table shows the folder names:

Table 8-4 *Folder Names for the Different Operating Systems*

Operating System	Folder Name
AIX	`ibmpow`
Linux	`linux`
Solaris	`sunspa`

  - Back up the files listed below, and then reinstall the driver. Reinstalling the driver shim re-creates the symbolic links, but it overwrites certain files. If you have made changes to them, you need to make a backup.

    Back up the following files:

    `rdxml.startnotes`
    `rdxml.stopnotes`
    `findDomino`
    `rdxml.confignotes` (or wherever your configuration is stored)

    After reinstalling the driver shim, copy the backups to their original location.

- The sample scripts provided (`rdxml.startnotes`, `rdxml.stopnotes`, `findDomino`) produce a Remote Loader trace log for the driver that can be used for troubleshooting.

Backing Up the Driver

9

You can use Designer or iManager to create an XML file of the driver. The file contains all of the information entered into the driver during configuration. If the driver becomes corrupted, the exported file can be imported to restore the configuration information.

IMPORTANT: If the driver has been deleted, all of the associations on the objects are purged. When the XML file is imported again, new associations are created through the migration process.

Not all server-specific information stored on the driver is contained in the XML file. Make sure this information is documented through the Document Generation process in Designer. See "Generating a Document" in the *Designer 2.1 for Identity Manager 3.5.1*.

- Section 9.1, "Exporting the Driver in Designer," on page 139
- Section 9.2, "Exporting the Driver in iManager," on page 139

9.1 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.

9.2 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 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 and select a location to save the XML file, then click *Save*.

10 Click *Finish*.

10 Security: Best Practices

In order to secure the driver and the information it is synchronizing, see "Security: Best Practices" in the *Novell Identity Manager 3.5.1 Administration Guide*.

Using the Movecfg.exe Utility

A

The movecfg.exe utility is a Windows console command line utility to be utilized when upgrading the Identity Manager Driver for Lotus Notes 1.*x* to version 2.2. It is installed if you select the option to install utilities during the Identity Manager installation.

The movecfg.exe utility is used to move specific Identity Manager Driver for Lotus Notes 1.*x* parameters from the Windows registry to the Identity Manager Driver for Lotus Notes 2.2 parameters location in the Identity Vault.

If you have multiple instances of ndsrep, you must run movecfg.exe once for each one, using the -ndsrep parameter.

With version 2.2 of the Identity Manager Driver for Lotus Notes, the ndsrep Domino add-in process reads configuration parameters from a Lotus Notes database (dsrepcfg.nsf). Prior to version 2.0, these parameters were stored in the Windows registry (\HKEY_LOCAL_MACHINE\SOFTWARE\NOVELL\VRD\DOMINO).

This utility attempts to move the necessary parameters from the Windows registry to the Lotus Notes Driver object (that is being upgraded) in the Identity Vault. It also attempts to place the LastEventTimeStamp for ndsrep that is stored in the registry into a Lotus Notes database (dsrepcfg.nsf). The LastEventTimeStamp is not stored as a driver parameter in the Identity Vault. For this reason it is placed directly into the ndsrep configuration database (dsrepcfg.nsf).

You can use a batch file such as the example provided in Section A.2, "Example Batch File to Use," on page 144.

NOTE: This utility is not localized for specific languages. All parameter descriptions that are imported into a specified driver are in English.

A.1 Prerequisites

- Identity Manager utilities are installed. The movecfg.exe utility is installed if you select the Utilities option when installing Identity Manager 3. If you did not install the utilities when you installed Identity Manager, you can rerun the Identity Manager install and select to install just the utilities, or you can download movecfg.exe from the Utilities directory on the Identity_Manager_3_Linux_NW_Win.iso CD.
- Run movecfg.exe from the Domino Server machine. The movecfg.exe utility should be executed from the same Domino Server that launches ndsrep.
- Domino Server is active. The Domino Server must be up and running.
- LDAP access is required to eDirectory. The user name (in LDAP form) and password must be passed as parameters to movecfg. If a password is not passed, movecfg prompts for it. The

password is not encrypted, so clear text passwords must be accepted by the LDAP server, or the LDAP server bind will fail.

- Lotus Notes ID file password. When the movecfg utility attempts to create (or update) the ndsrep configuration database (`dsrepcfg.nsf`), it prompts for the Lotus Notes password of the Notes ID file that last accessed the Domino server (or possibly client) from this machine (this Notes ID file is referenced from the `notes.ini` file). If this password is entered correctly, the ndsrep configuration database (`dsrepcfg.nsf`) can then be appropriately updated with the LastEventTimeStamp copied from the ndsrep configuration in the registry. For `dsrepcfg.nsf` to be initially created by `movecfg.exe`, `dsrepcfg.ntf` (which is distributed with the Identity Manager Driver 2.2 for Lotus Notes) must be available to the Domino Server (at `c:\Lotus\Domino\Data\dsrepcfg.ntf`).
- Multiple Lotus Notes driver instances. If you have more than one Lotus Notes driver connected to the same Domino server, `movecfg.exe` must be run once for each instance of the Lotus Notes driver that is being converted. To convert Lotus Notes Driver parameters that are not the default driver parameters (but are the 2nd, 3rd, 4th, etc. Notes driver parameters) the -ndsrep parameter must be utilized.

A.2 Example Batch File to Use

You can run the `movecfg.exe` utility with a batch file like the following example:

```
@echo off
REM
***********************************************************************
********
REM
REM Name: MoveCfg1to2.bat
REM Description: Sample batch file to demonstrate the usage and launch
parameters
REM               of movecfg.exe
REM              See movecfg.txt for descriptions of movecfg.exe usage
parameters
REM
REM Copyright (C) 2003-2004 Novell, Inc., All Rights Reserved
REM
REM
***********************************************************************
********
setlocal

REM echo on

REM SAMPLE CALL 1
call movecfg.exe -host server.acme.com -port 389 -edir-dn
cn=admin,o=acme -edir-pwd acmePass
-driverDN cn=NotesDriver,cn=DriverSet1,o=acme -noteSvr cn=Domino1/
o=acme -timeout 15

REM SAMPLE Call 2: When converting a second or third Notes driver on
the same machine, use the -ndsrep parameter
REM call movecfg.exe -host server.acme.com -port 389 -edir-dn
cn=admin,o=acme -edir-pwd acmePass -driverDN
```

```
cn=Notes2Driver,cn=DriverSet1,o=acme -noteSvr cn=Domino1/o=acme -
timeout 15 -ndsrep Notes2Driver
```

A.3 Using the Movecfg.exe Utility

The `movecfg.exe` utility contains the following parameters:

```
movecfg -host <ldap host name/address> -port <port number> -edir-dn
<login dn> -edir-pwd <password> -driverDN <driverDN> -noteSvr
<Domino Server Name> [-ndsrep] <NDSREP instance name> [-timeout]
<timeout>
[-f] <ndsrep config db>
```

Example:

```
movecfg -host ldapsvr.mycompany.com -port 389 -edir-dn
cn=admin,o=MyOrg
-edir-pwd secret -driverDN cn=myDriver,cn=MyOrgUnit,O=MyOrg -noteSvr
CN=MyDomino/O=MyOrg
```

Table A-1 *The Movecfg Utility's Parameters with Their Descriptions*

Parameter Name	Required or Optional	Description
-host <ldap host name/address>	Required	The DNS host name or the IP address of the LDAP host of the Identity Vault Server.
-port <port number>	Optional	LDAP port of the LDAP host specified by the -host parameter. Default = 389
-edir-dn <login dn>	Required	The fully qualified LDAP distinguished name of the Identity Vault user that updates the driver configuration. It must be in LDAP form. Example: cn=DirXMLAdmin,cn=eng,o=acme
-edir-pwd <password>	Optional	The password matching the user object specified by the -edir-dn login object. If a password is not supplied, a password prompt is presented.
-driverDN <driverDN>	Required	The fully qualified LDAP distinguished name of the driver that needs its parameters updated. It must be in LDAP form. Example: cn=NotesDriver1,cn=DirXMLDriverSet,o=acme
-noteSvr <Domino Server Name>	Required	The Domino Server Name. Example cn=NoteSrv/o=acme

Parameter Name	Required or Optional	Description
[-ndsrep] <NDSREP instance name>	Optional	The name of the Driver configuration instance to be stored in the ndsrep configuration database (`dsrepcfg.nsf`). By default this is set to the relative distinguished name of the Driver (such as NotesDriver1).
[-timeout] <timeout>	Optional	The timeout value when attempting to connect and communicate with the LDAP host.
[-f] <ndsrep config db>	Optional	The name of the ndsrep configuration database. Default = `dsrepcfg.nsf`

A.4 Troubleshooting

If the `movecfg` utility is not successful in updating your outdated Lotus Notes Driver configuration, try following the manual process.

1 Shut down the Domino Server where ndsrep is launched.

2 Shut down the Identity Manager Driver for Lotus Notes that you are upgrading.

3 Copy the following text from this document, and paste it into the <publisher-options> section of the Lotus Notes Driver configuration.

```
<publisher-options>
   <polling-interval display-name="Polling Interval (in 
seconds)">30</polling-interval>
   <loop-detect-flag display-name="Enable Loop Back 
Detection">Yes</loop-detect-flag>
   <schedule-units display-name="NDSREP Schedule Units">SECONDS</
schedule-units>
   <schedule-value display-name="NDSREP Schedule Value">30</
schedule-value>
   <dn-format display-name="DNFormat">SLASH</dn-format>
   <check-attrs-flag display-name="Check Attributes?">Yes</check-
attrs-flag>
   <write-timestamps-flag display-name="Write Time Stamps?">No</
write-timestamps-flag>
</publisher-options>
```

4 Use the Regedit utility on Windows to view each ndsrep configuration value. The regedit key values are under `\HKEY_LOCAL_MACHINE\SOFTWARE\NOVELL\VRD\DOMINO`.

Update the new Lotus Notes Driver publisher-options configuration values to match the corresponding values stored within the Windows registry.

5 Start the Lotus Notes Driver and the Domino Server.

DirXML Command Line Utility

B

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

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

eDirectory 8.7.x

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

eDirectory 8.8.x

- Windows: `\Novell\Nds\dxcmd.bat`
- NetWare®: `sys:\system\dxcmd.ncf`
- UNIX: `/opt/novell/eDirectory/bin/dxcmd`

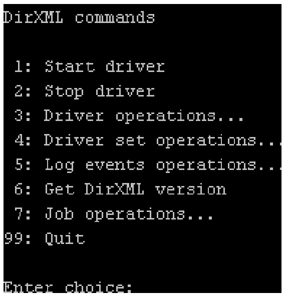
There are two different methods for using the DirXML Command Line utility:

- Section B.1, "Interactive Mode," on page 147
- Section B.2, "Command Line Mode," on page 156

B.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 you want to perform.

   Table B-1 on page 148 contains the list of options and what functionality is available.
5. Enter 99 to quit the utility.

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.

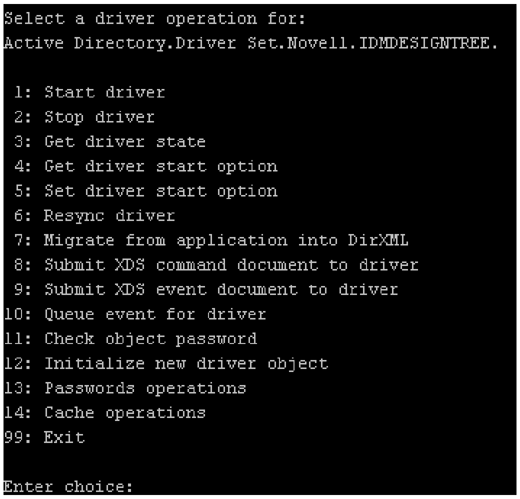
```
novell.jclient.JCException: connect (to address) 111 UNKNOWN ERROR
```

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 B-1 *Interactive Mode Options*

Option	Description
1: *Start Driver*	Starts the driver. If there is more than one driver, each driver is listed with a number. Enter the number of the driver to start the driver.
2: *Stop Driver*	Stops the driver. If there is more than one driver, each driver is listed with a number. Enter the number of the driver to stop the driver.
3: *Driver operations*	Lists the operations available for the driver. If there is more than one driver, each driver is listed with a number. Enter the number of the driver to see the operations available. See Table B-2 on page 149 for a list of operations.
4: *Driver set operations*	Lists the operations available for the driver set. ◆ 1: Associate driver set with server ◆ 2: Disassociate driver set from server ◆ 99: Exit
5: *Log events operations*	Lists the operations available for logging events through Novell® Audit. See Table B-5 on page 154 for a description of these options.
6: *Get DirXML version*	Lists the version of the Identity Manager installed.
7: *Job operations*	Manages jobs created for Identity Manager.1
99: *Quit*	Exits the DirXML Command Line utility

Figure B-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:
```

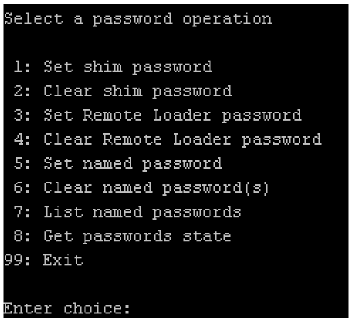
Table B-2 Driver Options

Options	Description
1: *Start driver*	Starts the driver.
2: *Stop driver*	Stops the driver.
3: *Get driver state*	Lists the state of the driver. ◆ 0 - Driver is stopped ◆ 1 - Driver is starting ◆ 2 - Driver is running ◆ 3 - Driver is stopping
4: *Get driver start option*	Lists the current driver start option. ◆ 1 - Disabled ◆ 2 - Manual ◆ 3 - Auto
5: *Set driver start option*	Changes the start option of the driver. ◆ 1 - Disabled ◆ 2 - Manual ◆ 3 - Auto ◆ 99 - Exit

Options	Description
6: *Resync driver*	Forces a resynchronization of the driver. It prompts for a time delay: *Do you want to specify a minimum time for resync? (yes/no).* If you enter Yes, specify the date and time you want the resynchronization to occur: *Enter a date/time (format 9/27/05 3:27 PM)*. If you enter No, the resynchronization occurs immediately.
7: *Migrate from application into DirXML*	Processes an XML document that contains a query command: *Enter filename of XDS query document:* Create the XML document that contains a query command by using the Novell `nds.dtd` (http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/ref/ndsdtd/query.html). Examples: NetWare: `sys:\files\query.xml` Windows: `c:\files\query.xml` Linux: `/files/query.xml`
8: *Submit XDS command document to driver*	Processes an XDS command document: *Enter filename of XDS command document:* Examples: NetWare: `sys:\files\user.xml` Windows: `c:\files\user.xml` Linux: `/files/user.xml` *Enter name of file for response:* Examples: NetWare: `sys:\files\user.log` Windows: `c:\files\user.log` Linux: `/files/user.log`
9: Submit XDS event document to driver	Processes an XDS event document: *Enter filename of XDS event document:* Examples: NetWare: `sys:\files\add.xml` Windows: `c:\files\add.xml` Linux: `/files/add.xml`

Options	Description
10: Queue event for driver	Adds and event to the driver queue *Enter filename of XDS event document:* Examples: NetWare: `sys:\files\add.xml` Windows: `c:\files\add.xml` Linux: `/files/add.xml`
11: *Check object password*	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). *Enter user name*:
12: *Initialize new driver object*	Performs an internal initialization of data on a new Driver object. This is only for testing purposes.
13: *Password operations*	There are nine Password options. See Table B-3 on page 151 for a description of these options.
14: *Cache operations*	There are five Cache operations. See Table B-4 on page 153 for a descriptions of these options.
99: *Exit*	Exits the driver options.

Figure B-2 *Password Operations*



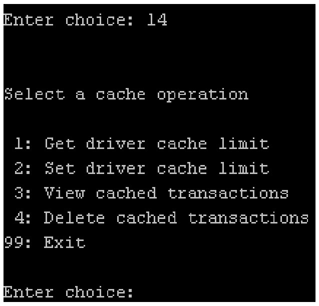
```
Select a password operation

 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 B-3 *Password Operations*

Operation	Description
1: *Set shim password*	Sets the application password. This is the password of the user account you are using to authenticate into the connected system with.
2: *Clear shim password*	Clears the application password.

Operation	Description
3: *Set Remote Loader password*	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: *Clear Remote Loader password*	Clears the Remote Loader password so no Remote Loader password is set on the Driver object.
5: *Set named password*	Allows you to store a password or other pieces of security information on the driver. See Section 4.3, “Using Named Passwords,” on page 52 for more information. There are four prompts to fill in: ◆ *Enter password name:* ◆ *Enter password description:* ◆ *Enter password:* ◆ *Confirm password:*
6: *Clear named passwords*	Clears a specified named password or all named passwords that are stored on the driver object: *Do you want to clear all named passwords? (yes/no).* 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.
7: *List named passwords*	Lists all named passwords that are stored on the driver object. It lists the password name and the password description.
8: *Get password state*	Lists if a password is set for: ◆ Driver Object password ◆ Application password ◆ Remote loader password The dxcmd utility allows you to set the Application password and the Remote Loader password. You cannot set the Driver Object password with this utility. It shows if the password has been set or not.
99: *Exit*	Exits the current menu and takes you back to the Driver options.

Figure B-3 *Cache Operations*

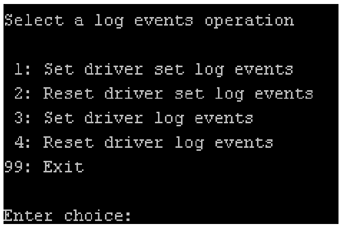


```
Enter choice: 14


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 B-4 *Cache Operations*

Operation	Description
1: *Get driver cache limit*	Displays the current cache limit that is set for the driver.
2: *Set driver cache limit*	Sets the driver cache limit in kilobytes. A value of 0 is unlimited.
3: *View cached transactions*	A text file is created with the events that are stored in cache. You can select the number of transactions to view. ◆ *Enter option token* (default=0): ◆ *Enter maximum transactions records to return* (default=1): ◆ *Enter name of file for response:*
4: *Delete cached transactions*	Deletes the transactions stored in cache. ◆ *Enter position token* (default=0): ◆ *Enter event-id value of first transaction record* to delete (optional): ◆ *Enter number of transaction records to delete* (default=1):
99: *Exit*	Exits the current menu and takes you back to the Driver options.

Figure B-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 B-5 Log Events Operations

Operation	Description
1: *Set driver set log events*	Allows you to log driver set events through Novell Audit. There are 49 items you can select to log. See Table B-6 on page 154 for a list of these options. Type the number of the item you want to log. After the items are selected, enter 99 to accept the selections.
2: *Reset driver set log events*	Resets all of the log event options.
3: *Set driver log events*	Allows you to log driver events through Novell Audit. There are 49 items to select to log. See Table B-6 on page 154 for a list of these options. Type the number of the item you want to log. After the items are selected, enter 99 to accept the selections.
4: *Reset driver log events*	Resets all of the log event options.
99: *Exit*	Exits the log events operations menu.

Table B-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

Options
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
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

Options
49: Clear SSO Credential
50: Set SSO Passphrase
51: User defined IDs
99: Accept checked items

Table B-7 *Enter Table Title Here*

Options	Description
1: *Get available job definitions*	Allows you to select an existing job. *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.

B.2 Command Line Mode

The command line mode allows you to use script or batch files. Table B-8 on page 156 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 B-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.

Option	Description
-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 `dxcmd` command to `stdout`.
-? <*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 `nds.dtd` (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>*	Processes an XDS command document. Specify the XDS command document as the input file. Examples: NetWare: `sys:\files\user.xml` Windows: `c:\files\user.xml` Linux: `/files/user.log` Specify the output filename to see the results. Examples: NetWare: `sys:\files\user.log` Windows: `c:\files\user.log` Linux: `/files/user.log`
-sendevent *<driver dn> <input filename>*	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.
-queueevent *<driver dn> <input filename>*	Submits a document to the driver's Subscriber channel by queuing the document in the driver cache. The document gets 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.
-setlogevents *<dn> <integer ...>*	Sets Novell Audit log events on the driver. The integer is the option of the item to log. See Table B-6 on page 154 for the list of the integers to enter.
-clearlogevents *<dn>*	Clears all Novell Audit log events that are set on the driver.
-setdriverset *<driver set dn>*	Associates a driver set with the server.
-cleardriverset	Clears the driver set association from the server.
-getversion	Shows the version of Identity Manager that is installed.
-initdriver object *<dn>*	Performs an internal initialization of data on a new Driver object. This is only for testing purposes.
-setnamedpassword *<driver dn> <name> <password> [description]*	Sets named passwords on the driver object. You specify the name, the password, and the description of the named password.
-clearnamedpassword *<driver dn> <name>*	Clears a specified named password.
-startjob *<job dn>*	Starts the specified job.

Option	Description
-abortjob *<job dn>*	Aborts 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.
-clearallnamedpaswords *<driver dn>*	Clears all named passwords set on a specific driver.

If a command line is executed successfully, it returns a zero. If the command line 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 B-9 on page 159 contains other values for specific command line options.

Table B-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	Return is the next scheduled time for the job in eDirectory time format (number of seconds since 00:00:00 Jan 1, 1970UTC).

Properties of the Driver

C

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

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

C.1 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 and click *Properties > Driver Configuration.*

There are different sections under *Driver Configuration.* Each section is listed in a table. The table contains a description of the fields, and the default value or an example of what value should be specified in the field.

C.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, then right-click the driver line and select *Properties > Driver Configuration.*
2. Select the *Driver Module* tab.

Option	Description
Java	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 `classes` directory as a class file, or in the `lib` directory as a `.jar` file. If this option is selected, the driver is running locally.
Native	Used to specify the name of the `.dll` file that is instantiated for the application shim component of the driver. If this option is selected, the driver is running locally.
Connect to Remote Loader	Used when the driver is connecting remotely to the connected system.
Remote Loader Client Configuration for Documentation	Includes the Remote Loader client configuration information in the driver documentation that is generated by Designer.

C.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, then right-click the driver line and click *Properties > Driver Configuration.*
2. Click *Driver Module > Connect to Remote Loader > Driver Object Password > Set Password.*

Option	Description
Driver Object Password	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 or the remote driver does not run. This password is used by the Remote Loader to authenticate itself to the remote driver shim.

C.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, then right-click the driver line and select *Properties > Driver Configuration*.
2. Click *Authentication*.

Option	Description
Authentication ID or *User ID*	Specify a user application ID. This ID is used to pass Identity Vault subscription information to the application. Example: `CN=NotesDriver/o=novell`
Authentication Context or *Connection Information*	Specify the IP address or name of the server the application shim should communicate with. The connection string uses the following format: `CN=DominoServer1/o=novell`
Remote Loader Connection Parameters or *Host name* Port *KMO* *Other parameters*	Used only if the driver is connecting to the application through the remote loader. The parameter to enter is `hostname=xxx.xxx.xxx.xxx port=xxxx kmo=certificatename`, 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 `kmo` entry is optional. It is only used when there is an SSL connection between the Remote Loader and the Metadirectory engine. Example: `hostname=10.0.0.1 port=8090 kmo=IDMCertificate`

Option	Description
Driver Cache Limit (kilobytes) or *Cache limit (KB)*	Specify the maximum event cache file size (in KB). If it is set to zero, the file size is unlimited. Click *Unlimited* to set the file size to unlimited in Designer.
Application Password or *Set Password*	Specify the password for the user object listed in the *Authentication ID* field.
Remote Loader Password or *Set Password*	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.

C.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, then right-click the driver line and select *Properties > Driver Configuration.*
2. Click *Startup Option*.

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

C.1.5 Driver Parameters

In iManager:

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

In Designer:

1. Open a project in the Modeler, then right-click the driver line and select *Properties > Driver Configuration*.
2. Click *Driver Parameters*. (See Table C-1.)

Table C-1 *Driver Parameters for the Notes Driver*

Parameter	Description
Driver Options	
Driver parameters for server: Server-name	Specify the name of the server on which the driver resides.
Edit XML	Allows you to edit the Lotus Notes driver parameters through an XML editor.
Notes Domain Name	Specify the name of the Notes Domain. An example is *NotesDomain*.
Notes Driver User ID File	Specify the full path (on the Domino Server) for the Notes User ID file associated with the Notes User this driver will use for Notes Authentication. An example is `c:\lotus\domino\data\admin.id`.
Default Certifier ID File	Specify the full path (on the Domino server) for the Default Notes Certifier ID file the driver will use as the default certifier. This is usually the root certifier, but can be any certifier with adequate access. An example is `c:\lotus\domino\data\cert.id`.
Default Notes Certifier Password	Specify the password for the Default Notes Certifier ID this driver will use when certifying new users, then reconfirm the password. You can also clear the password. This password is secured using the Named Passwords feature. See Section 4.3, "Using Named Passwords," on page 52.
Directory File	The Notes database filename the Notes Driver is synchronizing. For example, `names.nfs` is the Notes public address book.
Notes Address Book?	Select True if the Notes database that is being synchronized is a Notes Address Book. Select False if the Notes database that is being synchronized is not a Notes Address Book.

Parameter	Description
Update File	The ndsrep program creates an output database (by default, `ndsrep.nsf`), detects changes in the address book in the Domino server (or other Notes database), and copies these changes to the output database. The default filename is `ndsrep.nsf`.
Subscriber Options	
Certify (register) Notes Users	Whether the driver should certify users added to Notes on the Subscriber channel. The default is True.
Registration/Certification Log File	The Certification log file of the Domino server (`certlog.nsf`). For the Notes driver to register or certify Notes users, the Notes driver user must have rights to create entries in this database.
Default User ID File/Registration Expiration Term (in years)	Specify the expiration term (in years) for ID files created by the driver when certifying users added on the Subscriber channel. The default is *2*.
User ID File Storage Location	The full directory path of the location to create UserID files for newly created users.
Add Registered Users to Address Book	Select True if you want to add newly registered users in the Notes Address Book. Select False so newly registered users who have a UserID file created are not placed in the Notes Address Book.
Store User ID file in Address Book?	Whether Notes should store new user IDs in the address book when certifying users added to Notes on the Subscriber channel. The default is *False*.
Is Domino Server North American?	Is the Domino server this driver is binding to when certifying new users a North American Domino server? (This affects encryption levels.) Choose True for 128-bit encryption. The default is *True*.
Notes Password Strength (0 - 16)	Specify the minimum password strength for new Notes user IDs (0 - 16). The default is *5*.
Create User E-mail Box	Select the desired option. Select True if you want to create a Notes e-mail account for users. Select False if you do not want to create an e-mail account for users. The default is True.
Domino Mail Server Name	Specify the name of the Domino mail server this driver will authenticate to (in fully qualified canonical form, such as cn=MyMailServer/o=Organization).
Mail File Template	Specify the filename (on the Domino server) for the mail database template this driver will use when creating new mailfile users. The path should be relative to the Domino mail server's data directory.
User Mail File Storage Location	Specify the directory where you want to store user mail files. An example value is *mail*.
Default Notes Password:	Specify the default password for newly registered users if no other password is supplied to the Notes driver shim. An example value is *notes*.
Default Notes HTTP Password	Specify the default Notes Web HTTP password for new Notes users if no other password is supplied to the Notes driver shim. An example value is *notes*.

Parameter	Description
Publisher Options	
Polling Interval (in seconds)	Specify the polling interval (in seconds) for how often the Publisher channel checks the change log for updates. The default is *30*.
Enable Loop Back Detection	Select True to enable event loopback capability, or False to disable event loopback detection. Loopbacks cannot be used to determine object changes if they are performed by the Administration Process (AdminP). The default is True.
NDSREP Polling Units	The units of time used to specify the ndsrep polling interval. You can specify *Seconds*, *Minutes*, *Hours*, *Days*, or *Years*.The default is *Seconds*.
NDSREP Polling Interval	A positive integer specifying the polling interval for ndsrep. The default value is *30*.
DN Format	Specify the distinguished name format used by the Publisher channel. Changing this parameter might also necessitate changing the Publisher Placement policy where the Notes Source DN is often interpreted to determine Identity Vault object placement. The choices are *NOTES_TYPED, NOTES, SLASH_TYPED, SLASH, LDAP_TYPE, LDAP, DOT_TYPED,* and *DOT*. The default is *Slash*.
NDSREP Domino Console Trace Level.	Specify the Domino console trace setting. The choices are *Silent, Normal, Verbose,* and *Debug*. The default is *Normal*.
Check Attributes	Whether all attributes should be checked for each object event. Select *True* for only changed attribute values to be published from Notes to Identity Manager. Select *False* for all attributes within the Publisher channel filter to be published from Notes to Identity Manager if a changed attribute is detected. The default is *True*.
Write Time Stamps	Whether driver time stamps should be written to each synchronized object. Select *True* to attach a Notes driver-specific time stamp to each object that the Notes driver modifies. This action is done to improve the driver's ability to detect object changes from replicated databases. Select *False* to disallow any special time stamps from being attached to Notes objects. The default is *False*.

C.2 Global Configuration Values

Global configuration values (GCVs) allow 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 like 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*. (See Table C-2.)

In Designer:

1 Open a project in the Modeler, then right-click the driver line and select *Properties* > *Global Config Values*. (See Table C-2.)

For *Password Configuration*, you should only edit the first two settings listed in Table C-2 on page 168. The others are GCVs regarding Password Synchronization that are common to all drivers. They should be edited by using iManager in *Passwords* > *Password Synchronization*, not here. Some of them have dependencies on each other that are represented only in the iManager interface. They are explained in "Password Synchronization across Connected Systems" in the *Novell Identity Manager 3.5.1 Administration Guide*.

Table C-2 *Global Configuration Values > Password Configuration*

Option	Description
Subscriber and Publisher Object Placement Settings	
Subscriber: Default User sync source container in eDirectory	Specify the default user container in eDirectory where user changes are detected for synchronization in Lotus Notes. An example is `Organization\OrgUnit\Users`.
Publisher: Default User sync destination container in eDirectory	Specify the default user container in eDirectory where user changes are placed for synchronization in eDirectory. An example is `Organization\OrgUnit\Users`.
Subscriber: Default Group sync source container in eDirectory	Specify the default group container in eDirectory where group changes are detected for synchronization in Lotus Notes. An example is `Organization\OrgUnit\Groups`.
Publisher: Default Group sync destination container in eDirectory	Specify the default group container in eDirectory where group changes are placed for synchronization in eDirectory. An example is `Organization\OrgUnit\Groups`.
Lotus Notes Certifier Names and Parameter References	
Fully Qualified Default Certifier Name	Specify the default Fully Qualified (typed) Notes Certifier name as found in the Notes Address Book. The root certifier can be used (an example is `/o=acme)`.
Default Certifier Name	Specify the default Notes Certifier name as found in the Notes Address Book. The root certifier can be used (an example is `/acme)`.

Option	Description
Default Certifier Driver Parameter Key	Specify the driver parameter key name that stores the default certifier ID file name. An example is cert-id-file.
Default Certifier Password Driver Parameter Key	Specify the driver parameter key name that stores the default certifier ID password. An example is cert-id-password.
Lotus Notes Add User Policy Settings	
Add Notes User Certification Option	Select the desired Notes User Certification option. Select True to create a Notes Certification ID file for the user. Select False to not create the Notes Certification ID file. The default is True.
Add User: User ID File Creation	Select the desired Notes User ID file creation option. Select True to create an ID file when registering users. Select False to not create the ID file. The default is True.
Add User: Store User ID File in Address Book	Select the desired Notes User ID file option. Select True to place a Notes Certification ID file for the user in the Notes address book. Select False to not place the Notes Certification ID file in the address book. The default is False.
Add User: User ID Expire Term (in years)	Specify the expiration term (in years) for ID files created by the driver when certifying users who are added on the Subscriber channel. This number specifies how many years the user's Certification ID file will be valid. The default is *2*.
Add User: User ID Expiration Date	Specify an expiration date, or leave the field blank to ignore this setting. Specify the date when the user's Certification ID file will expire. This entry has priority over the Expire Term entry.
Add User: Alternate Organization Unit	Specify an alternate Organizational Unit to be used for each registered user, or leave the field blank to ignore this setting.
Add User: Alternate Organization Unit Language	Specify an alternate Organizational Unit language to be used for each registered user, or leave the field blank to ignore this setting.
Add User: Notes Explicit Policy Name To Be Attached To User	Specify the desired Notes Explicit Policy Name to be attached to each registered user. When specified, registration policies are not executed.
Add User: Synchronize User's Internet Password	Select the user's Internet password option. Select True to synchronize the user password with the Web password. Select False to not synchronize user passwords. The default is True.
Add User: Notes User Password Check Setting	Select the desired option. Select *Default* to ignore this setting. Select *Check Password* to require users to enter a password when authenticating to servers that have password checking enabled. Select *Don't Check* to not require users to enter a password when authenticating to other servers. Select *Lockout* to prevent users from accessing servers that have password checking enabled. The default is *Check Password*.
Add User: Notes User Password Change Interval (in days)	Specify the desired user password change interval in days. Specify a number to indicate the days a password is valid and before the user must supply a new password.
Add User: Notes User Password Grace Period (in days)	Specify the desired user password grace period in days. Specify a number to indicate the days the grace period is valid before the user must supply a new password.

Option	Description
Add User: Notes User's Internet Password Change Required	Select the user's Internet password change option. Select True to require users to change the password on the next login. Select False to not require users to change the password on the next login. The default is False.
Add User: Roaming Option	Select the desired Notes roaming user option. Select True to enable roaming for Notes users. Select False to disable roaming. The default is False. Selecting True brings up the next four options.
Roaming User: Roam Server Name	Specify the Domino server that will host this roaming user. An example is (`cn=ServerName/o=org`)
Roaming User: Roam Server Subdirectory	Specify the Domino server subdirectory to store roaming user data. An example is `Roaming\`
Roaming User: Cleanup Setting	Select the Notes roaming user cleanup setting. Select *Default* to do nothing. Select *Never* to never delete roaming data. Select *Every n Days* to delete roaming data by the days specified by *Roaming Cleanup Period*. Select *At Shutdown* to delete Notes data when Notes shuts down. Select *User Prompt* to clean up roaming data when the user exits Notes; the user can also decline to be prompted in the future.
Roaming User: Cleanup Period (in days)	If *Every n Days* is selected as the *Roaming User Cleanup Setting*, specify the number of days before deleting roaming user data.
Add User: Create Unique Notes ShortName Field Value	If you want the Notes driver to generate a *ShortName* field value that is unique within the Notes Address Book, select True. Select False (default) to make no such attempt. ShortNames that are generated by selecting True use a combination of the Given Name and Surname and do not exceed eight characters in length.
Add User: Create Unique Notes InternetAddress Field Value	If you want the Notes driver to generate an *Internet Address* field value that is unique within the Notes Address Book, select True. Select False (default) to make no such attempt. *Internet Address* fields that are generated by selecting True use a combination of the Given Name, Surname, and the InternetDomainName GCV. For example, if the user's name is Glen Canyon and the InternetDomainName GCV is mycompany.com, the combinatioins will be GlenCanyon1@mycompany.com, GlenCanyon2@mycompany.com..., GlenCanyon10@mycompany.com, etc.
Lotus Notes E-mail Information	
Internet Mail Domain	Specify the Internet Mail Domain to be used when generating Internet e-mail addresses. An example is *mycompany.com*.
Add User E-mail Box	Select the desired Notes user e-mail creation option. Select True to create a Notes e-mail account for a user. Select False to not create an e-mail account. The default is True.
Add User E-mail: Create Mail File in Background via AdminP	Select the desired Notes user e-mail creation option. Select True to create a mail file by issuing a request to the Domino administration process to create the mail file in the background through AdminP. Select False to create the mail file directly. AdminP support is required for this option. The default is False.

Option	Description
Add User E-mail: Inherit from Mail File Template	Select the desired Notes user e-mail database inheritance option. Select True in order for the user e-mail database to inherit changes from the specified creation template. Select False to not inherit changes. You specify the e-mail creation template through the Subscriber channel settings. The default is True.
Add User E-mail: E-mail Database ACL Setting	Select the desired Notes user e-mail database ACL option. Select *Default* to ignore this setting. Other options include *Manager, Designer, Author, Editor, Reader, Depositor*, and *No Access*. The default is *Default*.
Add User E-mail: Mail ACL Manager	Specify the Notes e-mail database Manager name. Leave this entry blank to allow e-mail access by the owner. If ACL access of the mail database is less than *Manager*, you need to specify an e-mail manager. Use the Plus icon to add names, the Minus icon to delete names, and the Pen icon to edit present entries.
Add User E-mail: Mail File Size Quota (in Kilobytes)	Specify the Notes e-mail database size quota in kilobytes. Leave blank to ignore this setting. The size specifies disk space that the server administrator allows for the e-mail database. If the Notes driver user is not a Domino server administrator, leave this value blank.
Add User E-mail: Mail FIle Size Warning Threshold (in Kilobytes)	Specify the mail file size warning threshold in kilobytes. Leave blank to ignore this setting. The size specifies disk space allowed before warning messages are sent to the database owner.
Add User E-mail: Mail File Replication	Select the desired Notes user e-mail file replication option. Select *True* to replicate the mail file of a user. Select *False* to not replicate the mail file. The default is *False*.
Lotus Notes Object Deletion Policy Settings	
Lotus Notes Deny Access Group Name	Specify a Notes Deny Access Group as a placeholder for disabled users. An example is *Deny Access*.
Remove Notes Account	Select the method to remove Notes user accounts. Select *Disable* to place user into the Notes Deny Access Group. Select *Delete* to send the delete event to the Note server for deletion and removal from all groups (except for groups that are of the type Deny List). The default is *Disable*. Selecting *Delete* brings up the next four options.
Remove User or Group Object from the Notes Address Book Immediately	Select whether to immediately delete the user or group object from the Notes address book. Select True to immediately remove the user or group from the address book. Select False to remove the user or group from the Notes address book through the background administration process. The default is True.
Add Deleted User Name To Deny Access Group	Select whether to add the deleted user name to the Deny Access Group. Select True to insert the user name into the group specified by DenyAccessGrpName. Select False to not insert the name. The default is True.
User Delete Mail File Action	Specify the mailfile deletion action when a user is deleted. Select *All* to have e-mail removed from the home mail server and all replica mail servers when the user object is deleted. Select *Home* to have e-mail removed from only the home mail server when the user object is deleted. Select *None* to preserve all e-mail when the user object is deleted. The default is *None*.

Option	Description
Delete Object: Tell AdminP Process Command	Specifies the `Tell adminp Process` command to immediately send to the Domino server after an object is deleted from the Domino Public Address Book. Options include *No Action* (default), *All, New, Daily, Delayed, Interval, People*, and *Time*.
Domino Administration Process Activation Command Settings	
Add User: Tell AdminP Process Command	Select the `AdminP process` command to add a user. This specifies the `Tell adminp Process` command to send to the Domino server immediately after the user has been added to the Domino Public Address Book. Options include *No Action* (default), *All, New, Daily, Delayed, Interval, People*, and *Time*.
Modify User: Tell AdminP Process Command	Select the `AdminP process` command to add a user. This specifies the `Tell adminp Process` command to send to the Domino server immediately after the user has been modified using AdminP methods in the Domino Public Address Book. Options include *No Action* (default), *All, New, Daily, Delayed, Interval, People*, and *Time*.
User Password Policy Settings	
Application accepts passwords from Identity Manager	If True, this option allows passwords to flow from Identity Manager to the connected system. The default is True.
Notify the user of password synchronization failure via e-mail	If True, notify the user by e-mail of any password synchronization failures. The default is False. Selecting *True* brings up the next two options.
Default E-mail Notification User	Select the default user (administrator) to receive e-mail notifications. The user should have a valid Internet EMail Address attribute specified in the Identity Vault. Password Synchronization policies are configured to send e-mail notifications to the associated user when password updates fail. The selected user receives a copy of each notification e-mail. Be sure to select a user who has proper authorization to review password update actions (such as a security administrator). If the field is left blank, password synchronization notification e-mails are only sent to the affected user.
Connected System or Driver Name	Specify the name of the connected system, application, or Identity Manager driver. This value is used by the e-mail notification templates. An example is *Notes*.

C.3 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 configured Named Passwords, see Section 4.3, “Using Named Passwords,” on page 52.

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, then right-click the driver line and select *Properties* > *Named Passwords.*

C.4 Engine Control Values

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

In iManager:

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

2 Browse to the driver, then click the upper right corner of the driver icon.

3 Click *Edit Properties* > *Engine Control Values*.

This option does not exist in Designer at this time.

Table C-3 *Engine Control Values*

Option	Description
Subscriber channel retry interval in seconds	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.
Qualified form for DN-syntax attribute values	The qualified specification for DN-syntax attribute values controls whether values for DN-syntax attribute values are presented in unqualified slash form or qualified slash form. A True setting means the values are presented in qualified form.
Qualified form from rename events	The qualified form for rename events controls whether the new-name portion of rename events coming from the Identity Vault are presented to the Subscriber channel with type qualifiers. For example, CN=. A True setting means the names are presented in qualified form.

Option	Description
Maximum eDirectory replication wait time in seconds	The maximum eDirectory™ replication wait time 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.)
Use non-compliant backwards-compatible mode for XSLT	This control sets the XSLT processor used by the Metadirectory engine to a backwards-compatible mode. The backwards-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 backwards-compatibility with existing DirXML® style sheets that depend on the non-standard behaviors. 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 backwards-compatibility with existing DirXML style sheets.
Maximum application objects to migrate at once	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. 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. **NOTE:** This control does not limit the number of application objects that can be migrated; it merely limits the batch size.
Set creatorsName on objects created in Identity Vault	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. 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.
Write pending associations	This control determines whether the Identity Manager engine writes a pending association on an object during Subscriber channel processing. 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.

Option	Description
Use password event values	This control determines the source of the value reported for the nspmDistributionPassword attribute for Subscriber channel Add and Modify events. The default Notes driver configuration sets the value to *True*. Setting the control to False means that the current value of the nspmDistributionPassword is obtained and reported as the value of the attribute event. This means that only the current password value is available. Setting the control to True means that the value recorded with the eDirectory event is decrypted and is reported as the value of the attribute event. This means that both the old password value (if it exists) and the replacement password value at the time of the event are available. This is useful for synchronizing passwords to certain applications such as Lotus Notes that require the old password to enable setting a new password.
Enable password synchronization status reporting	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.

C.5 Log Level

Every driver set and 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). Change the log level if you want to track additional message types.

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

In iManager:

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

2 Browse to the driver, then click the upper right corner of the driver icon.

3 Click *Edit Properties > Log Level*.

In Designer:

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

Option	Description
Use log settings from the DriverSet	If this is selected, the driver logs events as the options are set on the Driver Set object.
Log errors	Logs just errors

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

C.6 Driver Image

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

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

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

In iManager:

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

In Designer:

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

C.7 Security Equals

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

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

In iManager:

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

2 Browse to the driver, then click the upper right corner of the driver icon.

3 Click *Edit Properties* > *Security Equals*.

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

C.8 Filter

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

In iManager:

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

2 Browse to the driver, then click the upper right corner of the driver icon.

3 Click *Edit Properties* > *Filter*.

The filter editor is accessed through the outline view in Designer.

1 In an open project, click the *Outline* tab.

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.

C.9 Edit Filter XML

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

In iManager:

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

2 Browse to the driver, then click the upper right corner of the driver icon.

3 Click *Edit Properties* > *Filter*.

You can edit the Filter in XML through the Filter Editor.

1 In an open project, click the *Outline* tab.

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 and to launch the Filter Editor, then click *XML Source* at the bottom of the Filter Editor.

C.10 Misc

Allows you to add a trace level to your driver. With the trace level set, DSTRACE displays the Identity Manager events as the Metadirectory engine processes the events. The trace level only affects 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, then right-click the driver line and select *Properties* > *Trace.*

Option	Description
Trace level	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.
Trace file	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.
Trace file size limit	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. **NOTE:** The trace file is created in multiple files. Identity Manager automatically divides the maximum file size by ten and creates ten separate files. The combined size of these files equals the maximum trace file size.
Trace name	Driver trace messages are prepended with the value entered in this field.
Use setting from Driver Set	This option is only available in Designer. It allows the driver to use the same setting that is set on the Driver Set object.

C.11 Excluded Users

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

In iManager:

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

Designer does not list the excluded users.

C.12 Driver Manifest

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

In iManager:

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

In Designer:

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

C.13 Driver Inspector

The Driver Inspector page displays information about all of the 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:** Select this option to re-read all of the objects associated with the driver and refresh the displayed information.
- **Actions:** Allows 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 objects associated with the driver that have a Disabled state.
    - **Filter for Manual Associations:** Displays all objects associated with the driver that have a Manual state.
    - **Filter for Migrate Associations:** Displays all objects associated with the driver that have a Migrate state.
    - **Filter for Pending Associations:** Displays all objects associated with the driver that have a Pending state.
    - **Filter for Processed Associations:** Displays all objects associated with the driver that have a Processed state.
    - **Filter for Undefined Associations:** Displays all objects associated with the driver 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.

C.14 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:** Allows you to edit the properties of the currently selected Server object.
- **Delete:** Deletes the selected items from the cache file.
- **Refresh:** Select this option 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: Allows you to specify a desired number.
- **Actions:** Allows 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:** Allows you to access a specified entry in the cache file. Specify the entry number, then click *OK*.
    - **Cache Summary:** Summarizes all of the events stored in the cache file.

C.15 Inspector

The Inspector displays information about the connected system without directly accessing the system. Designer does not have this option.

C.16 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
Identity Manager accepts password (Publisher Channel)	If this option is enabled, Identity Manager allows passwords to flow from the connected system driver into the Identity Vault data store. 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. 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.

Option	Description
Use Distribution Password for password synchronization	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. 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. 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. 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 *Synchronize Universal Password with Distribution Password*. This use of Identity Manager Password Synchronization is also referred to as “tunneling.”
Accept password only if it complies with user’s Password Policy	To use this setting, users must have a Password policy assigned that has Universal Password enabled, and Advanced Password Rules enabled and configured. 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. 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.

Option	Description
If password does not comply, ignore Password Policy on the connected system by resetting user's password to the Distribution Password	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. Keep in mind that if you do not select this option, user passwords can become out-of-sync on connected systems. 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. **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.
Always accept password; ignore Password Policies	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.
Application accepts passwords (Subscriber Channel)	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. 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. 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 *Synchronize Universal Password with Distribution Password*. This use of Password Synchronization is also referred to as "tunneling."
Notify the user of password synchronization failure via-email	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. **NOTE:** To set up e-mail notification, select *Passwords > Edit EMail Templates*.

Sample XDS Documents for Custom Parameters

D

The driver supports using AdminP process such as Delete, Move, and Rename. These features require you to use Notes 6.0.3 or later, turn on AdminP support for the driver (see Allow Domino AdminP Support in Section 4.4.2, “Subscriber Options,” on page 55), and make changes to your driver policy.

The driver also supports sending commands to the Domino server console.

This section provides examples of the event produced by the Metadirectory engine, and the command that must be given to the driver shim. Policy samples are not provided, but the example shows how the event must be transformed and shows attributes that need to be provided by policies.

Refer to Section 4.9, “Move/Rename,” on page 99 and Section 4.10, “TELL AdminP Commands,” on page 102 for more information.

- Section D.1, “Sample of Adding a User,” on page 185
- Section D.2, “Sample of Renaming: Modifying a User Last Name,” on page 187
- Section D.3, “Sample of Moving a User,” on page 188
- Section D.4, “Sample of Deleting a User,” on page 190
- Section D.5, “Samples of Sending a Command to the Domino Server Console,” on page 191
- Section D.6, “Replication (Rep) Attribute Tags,” on page 192
- Section D.7, “Sample ACL Entry Tags,” on page 195
- Section D.8, “Setting and Modifying Lotus Notes Field Flags,” on page 199

D.1 Sample of Adding a User

This section shows a sample of the events when creating user John Doe in the Identity Vault.

- Section D.1.1, “Add Event Produced by the Metadirectory Engine,” on page 185
- Section D.1.2, “Add Event Received by the Notes Driver Shim,” on page 186

D.1.1 Add Event Produced by the Metadirectory Engine

```
<nds dtdversion="2.0" ndsversion="8.x">
  <source>
    <product version="2.0.5.38 ">DirXML</product>
    <contact>Novell, Inc.</contact>
  </source>
  <input>
    <add  class-name="User"
            event-id="MYSERVER-NDS#20040603175534#1#1"
            qualified-src-
dn="O=DirXML\OU=Notes\OU=Users\OU=sales\CN=John Doe"
            src-dn="\mytree\DirXML\Notes\Users\sales\John Doe"
            src-entry-id="38727">
      <association state="pending"></association>
      <add-attr attr-name="CN">
        <value naming="true" timestamp="1086285300#20"
type="string">John Doe</value>
      </add-attr>
      <add-attr attr-name="Surname">
        <value timestamp="1086285300#3" type="string">Doe</value>
      </add-attr>
      <add-attr attr-name="Given Name">
        <value timestamp="1086285334#1" type="string">John</value>
      </add-attr>
    </add>
  </input>
</nds>
```

D.1.2 Add Event Received by the Notes Driver Shim

```
<nds dtdversion="2.0" ndsversion="8.x">
  <source>
    <product version="2.0.5.38">Identity Manager</product>
    <contact>Novell, Inc.</contact>
  </source>
  <input>
    <add    expire-term="5"             certify-user="Yes"
            class-name="Person"
            create-mail="Yes"
            dest-dn="cn=John Doe/ou=sales/o=dirxml"
            drv-param-cert-id="sales-cert-id-file"             drv-
param-cert-pwd="sales-cert-id-password"
            enforce-unique-short-name="No"
            event-id="MYSERVER-NDS#20040603175534#1#1"
            internet-password-force-change="Yes"             mail-acl-
level="MANAGER"             mail-acl-manager-name="CN=Notes Driver/
O=dirxml"            mail-file-quota="120000"             mail-quota-
warning-threshold="100000"             notes-password-change-
interval="100"             notes-password-check-
setting="PWD_CHK_CHECKPASSWORD"            notes-password-grace-
period="5"             notes-policy-name="/EmployeePolicy"
            qualified-src-
dn="O=DirXML\OU=Notes\OU=Users\OU=sales\CN=John Doe"
            roaming-cleanup-period="90"             roaming-cleanup-
setting="REG_ROAMING_CLEANUP_EVERY_NDAYS"             roaming-
server="cn=myserver/o=dirxml"             roaming-
subdir="Roaming\JohnDoe"             roaming-user="Yes"
            src-dn="\mytree\DirXML\Notes\Users\sales\John Doe"
            src-entry-id="38727"
            sync-internet-password="Yes">
      <add-attr attr-name="FullName">
        <value naming="true" timestamp="1086285300#20"
type="string">John Doe</value>
      </add-attr>
      <add-attr attr-name="LastName">
```

```
        <value timestamp="1086285300#3" type="string">Doe</value>
      </add-attr>
      <add-attr attr-name="FirstName">
        <value timestamp="1086285334#1" type="string">John</value>
      </add-attr>
      <add-attr attr-name="InternetAddress">
        <value>John Doe@dirxml.com</value>
      </add-attr>
    </add>
  </input>
</nds>
```

D.2 Sample of Renaming: Modifying a User Last Name

This section shows a sample of the events when changing a last name from Doe to Doerr in the Identity Vault. Refer to Section 4.9, “Move/Rename,” on page 99 for more information.

- Section D.2.1, “Modify Event Produced by the Metadirectory Engine,” on page 187
- Section D.2.2, “Modify Event Received by the Notes Driver Shim,” on page 188

D.2.1 Modify Event Produced by the Metadirectory Engine

```
<nds dtdversion="2.0" ndsversion="8.x">
  <source>
    <product version="2.0.5.38 ">Identity Manager</product>
    <contact>Novell, Inc.</contact>
  </source>
  <input>
    <modify class-name="User"
            event-id="MYSERVER-NDS#20040603175500#1#3"
            qualified-src-
dn="O=DirXML\OU=Notes\OU=Users\OU=sales\CN=John Doe"
            src-dn="\mytree\DirXML\Notes\Users\sales\John Doe"
            src-entry-id="38727"
            timestamp="1086291578#2">
      <association
state="associated">BB888BB0C35D13EC87256EA8006296CE</association>
      <modify-attr attr-name="Surname">
        <remove-value>
          <value timestamp="1086285300#3" type="string">Doe</value>
        </remove-value>
        <add-value>
          <value timestamp="1086291578#2" type="string">Doerr</value>
        </add-value>
      </modify-attr>
    </modify>
  </input>
</nds>
```

D.2.2 Modify Event Received by the Notes Driver Shim

```
<nds dtdversion="2.0" ndsversion="8.x">
  <source>
    <product version="2.0.5.38 ">Identity Manager</product>
    <contact>Novell, Inc.</contact>
  </source>
  <input>
    <modify class-name="Person"
                drv-param-cert-id="sales-cert-id-file"
drv-param-cert-pwd="sales-cert-id-password"
                event-id="MYSERVER-NDS#20040603175500#1#3"
                qualified-src-
dn="O=DirXML\OU=Notes\OU=Users\OU=sales\CN=John Doe"
                src-dn="\mytree\DirXML\Notes\Users\sales\John Doe"
                src-entry-id="38727"
                tell-adminp-process="tell adminp process all"
                timestamp="1086291578#2">
      <association
state="associated">BB888BB0C35D13EC87256EA8006296CE</association>
      <modify-attr attr-name="LastName">
        <remove-value>
          <value timestamp="1086285300#3" type="string">Doe</value>
        </remove-value>
        <add-value>
          <value timestamp="1086291578#2" type="string">Doerr</value>
        </add-value>
      </modify-attr>
    </modify>
  </input>
</nds>
```

D.3 Sample of Moving a User

This section shows a sample of the events when moving John Doerr from the OU=sales to OU=mktg in eDirectory™. Refer to Section 4.9, "Move/Rename," on page 99 for more information.

- Section D.3.1, "Move Event Produced by the Metadirectory Engine," on page 188
- Section D.3.2, "Move Event Received by the Notes Driver Shim," on page 189

D.3.1 Move Event Produced by the Metadirectory Engine

```
<nds dtdversion="2.0" ndsversion="8.x">
  <source>
    <product version="2.0.5.38 ">Identity Manager</product>
    <contact>Novell, Inc.</contact>
  </source>
  <input>
    <move    class-name="User"
            event-id="MYSERVER-NDS#20040603175500#1#1"
            old-src-dn="\mytree\DirXML\Notes\Users\sales\John Doerr"
            qualified-old-src-
dn="O=DirXML\OU=Notes\OU=Users\OU=sales\CN=John Doerr"

            qualified-src-
dn="O=DirXML\OU=Notes\OU=Users\OU=mktg\CN=John Doerr"
            src-dn="\mytree\DirXML\Notes\Users\mktg\John Doerr"
            src-entry-id="38727"
            timestamp="1086285300#1">
      <association
state="associated">BB888BB0C35D13EC87256EA8006296CE</association>
      <parent qualified-src-dn="O=DirXML\OU=Notes\OU=Users\OU=mktg"
              src-dn="\mytree\DirXML\Notes\Users\mktg" src-entry-
id="36691"/>
    </move>
  </input>
</nds>
```

D.3.2 Move Event Received by the Notes Driver Shim

```
<nds dtdversion="2.0" ndsversion="8.x">
  <source>
    <product version="2.0.5.38 ">Identity Manager</product>
    <contact>Novell, Inc.</contact>
  </source>
  <input>
    <move   certifier-name="/mktg/dirxml"
            class-name="Person"
           drv-param-cert-id="mktg-cert-id-file"            drv-param-
cert-pwd="mktg-cert-id-password"            drv-param-old-cert-
id="sales-cert-id-file"            drv-param-old-cert-pwd="sales-cert-
id-password"
            event-id="MYSERVER-NDS#20040603175500#1#1"
            old-src-dn="\mytree\DirXML\Notes\Users\sales\John Doerr"
            qualified-old-src-
dn="O=DirXML\OU=Notes\OU=Users\OU=sales\CN=John Doerr"
            qualified-src-
dn="O=DirXML\OU=Notes\OU=Users\OU=mktg\CN=John Doerr"
            src-dn="\mytree\DirXML\Notes\Users\mktg\John Doerr"
            src-entry-id="38727"
            tell-adminp-process="tell adminp process all"
            timestamp="1086285300#1">
      <association
state="associated">BB888BB0C35D13EC87256EA8006296CE</association>
      <parent qualified-src-dn="O=DirXML\OU=Notes\OU=Users\OU=mktg"
              src-dn="\mytree\DirXML\Notes\Users\mktg" src-entry-
id="36691"/>
    </move>
  </input>
</nds>
```

D.4 Sample of Deleting a User

This section shows a sample of the events when deleting John Doerr from eDirectory.

- Section D.4.1, “Delete Event Produced by the Metadirectory Engine,” on page 190
- Section D.4.2, “Delete Event Received by the Notes Driver Shim,” on page 190

D.4.1 Delete Event Produced by the Metadirectory Engine

```
<nds dtdversion="2.0" ndsversion="8.x">
  <source>
    <product version="2.0.5.38 ">Identity Manager</product>
    <contact>Novell, Inc.</contact>
  </source>
  <input>
    <delete class-name="User"
            event-id="MYSERVER-NDS#20040603195215#1#6"
            qualified-src-
dn="O=DirXML\OU=Notes\OU=Users\OU=mktg\CN=John Doerr"
            src-dn="\mytree\DirXML\Notes\Users\mktg\John Doerr"
            src-entry-id="38727"
            timestamp="1086292335#6">
      <association
state="associated">BB888BB0C35D13EC87256EA8006296CE</association>
    </delete>
  </input>
</nds>
```

D.4.2 Delete Event Received by the Notes Driver Shim

```
<nds dtdversion="2.0" ndsversion="8.x">
  <source>
    <product version="2.0.5.38 ">Identity Manager</product>
    <contact>Novell, Inc.</contact>
  </source>
  <input>
    <delete class-name="Person"
            delete-windows-user="false" deny-access-group-
id="7EFB951A3574521F87256E540001F140"
            event-id="MYSERVER-NDS#20040603195215#1#6"
            immediate="true"             mail-file-
action="MAILFILE_DELETE_ALL"             qualified-src-
dn="O=DirXML\OU=Notes\OU=Users\OU=mktg\CN=John Doerr"
            src-dn="\mytree\DirXML\Notes\Users\mktg\John Doerr"
            src-entry-id="38727"
            tell-adminp-process="tell adminp process all"
timestamp="1086292335#6">
      <association
state="associated">BB888BB0C35D13EC87256EA8006296CE</association>
    </delete>
  </input>
</nds>
```

D.5 Samples of Sending a Command to the Domino Server Console

This section shows an example of using the driver's ability to send a command to the Domino server console and receive a response.

- Section D.5.1, “Domino Console Command as Received by the Driver Shim,” on page 191
- Section D.5.2, “Command Response Returned by the Notes Driver Shim,” on page 191

D.5.1 Domino Console Command as Received by the Driver Shim

```
<nds dtdversion="1.0" ndsversion="8.5" xmlns:notes="http://
www.novell.com/dirxml/notesdriver">
  <input>
    <notes:domino-console-command event-id="0">show server -xml</
notes:domino-console-command>
  </input>
</nds>
```

D.5.2 Command Response Returned by the Notes Driver Shim

Responses are truncated after 32000 characters.

```
<nds dtdversion="2.0" ndsversion="8.x" xmlns:notes="http://
www.novell.com/dirxml/notesdriver">
      <source>
        <product build="20040602_1644" instance="NotesDriver"
version="2.1">Identity Manager Driver for Lotus Notes</product>
        <contact>Novell, Inc.</contact>
      </source>
      <output>
        <notes:domino-console-response event-id="0">
          <server platform="Windows/32" time="20040603T141140,48-06"
version="Release 6.5">
            <name>myserver/dirxml</name>
            <title>MyServer Domino Server</title>
            <directory>C:\Lotus\Domino\Data</directory>
            <partition>C.Lotus.Domino.Data</partition>
            <uptime days="6" hours="1" minutes="52" seconds="38"/>
            <transactions hour="80" minute="2" peak="3614"/>
           <sessions peaknumber="5" peaktime="20040528T130914,23-06"/>
            <transactions count="35797" maxconcurrent="20"/>
            <threadpool threads="40"/>
            <availability index="100" state="AVAILABLE"/>
            <mailtracking enabled="0" state="Not Enabled"/>
            <mailjournalling enabled="0" state="Not Enabled"/>
            <sharedmail enabled="0" state="Not Enabled"/>
            <mailboxes number="1"/>
            <mail dead="0" pending="0"/>
            <tasks waiting="0"/>
            <transactionlogging enabled="0"/>
            <hosting enabled="0"/>
            <faultrecovery enabled="0" state="Not Enabled"/>
            <activitylogging enabled="0" state="Not Enabled"/>
            <controller enabled="0" state="Not Enabled"/>

<diagnosticdirectory>C:\Lotus\Domino\Data\IBM_TECHNICAL_SUPPORT</
diagnosticdirectory>
            <consolelogging enabled="0" state="Not Enabled"/>

<consolelogfile>C:\Lotus\Domino\Data\IBM_TECHNICAL_SUPPORT\console.log
</consolelogfile>
          </server>
        </notes:domino-console-response>
        <status event-id="0" level="success"/>
      </output>
</nds>
```

D.6 Replication (Rep) Attribute Tags

- Section D.6.1, “The ADD Event Policy Rule For Database Replication,” on page 192
- Section D.6.2, “Mailfile Database Replication Attribute Tags As They Are Submitted To the Shim,” on page 192
- Section D.6.3, “Sample Modify Event Policy Rule,” on page 193
- Section D.6.4, “Modify Event Attribute Tags As They Are Submitted To the Shim,” on page 194

D.6.1 The ADD Event Policy Rule For Database Replication

To view a sample ADD Event policy rule to submit database replication parameters on behalf of a newly created mailfile, click setmailfilerep.xml (../samples/setmailfilerep.xml). You can use or modify the sample code for your own XML solutions.

D.6.2 Mailfile Database Replication Attribute Tags As They Are Submitted To the Shim

Below is a sample ADD Event containing mailfile database replication attribute tags as they are submitted to the Notes Driver shim. This sample modifies the replication settings of the newly created mailfile of the new Notes user and also creates a replica on server CN=server1/O=novell.

```
<nds dtdversion="2.0" ndsversion="8.x">
   <source>
      <product version="2.0.8.20050127 ">Identity Manager</product>
      <contact>Novell, Inc.</contact>
   </source>
   <input>
      <add  certify-user="true"
            class-name="Person"
            create-mail="true"
            dest-dn="CN=DaffyDuck/OU=eng/O=novell"
            drv-param-cert-id="eng-cert-id-file"
            drv-param-cert-pwd="eng-cert-id-password"
            event-id="BLACKCAP-NDS#20050331215122#1#1"
            mail-acl-manager-name="CN=Notes Driver/O=novell"
            mailfile-rep-abstract="false"
            mailfile-rep-clear-history="false"
            mailfile-rep-cutoff-delete="false"
            mailfile-rep-cutoff-interval="240"
            mailfile-rep-dest-server="CN=server1/O=novell"
            mailfile-rep-disabled="false"
            mailfile-rep-dont-send-local-security-updates="false"
            mailfile-rep-entry-remove="false"
            mailfile-rep-ignore-deletes="false"
            mailfile-rep-ignore-dest-deletes="false"
            mailfile-rep-immediate="CN=server1/O=novell"
            mailfile-rep-include-acl="true"
            mailfile-rep-include-agents="true"
            mailfile-rep-include-documents="true"
            mailfile-rep-include-forms="true"
            mailfile-rep-include-formulas="true"
            mailfile-rep-new-db-name="mail/daffyduck_rep1.nsf"
            mailfile-rep-new-server="CN=server1/O=novell"
            mailfile-rep-priority="LOW"
            mailfile-rep-src-server="CN=server2/O=novell"
            mailfile-rep-view-list="Inbox;Sent;Calendar;Meetings"
            qualified-src-
dn="O=DirXML\OU=Notes\OU=Users\OU=eng\CN=DaffyDuck"
            src-dn="\novell_tree\DirXML\Notes\Users\eng\DaffyDuck"
            src-entry-id="40729"
            timestamp="1112305882#22">
         <add-attr attr-name="FullName">
            <value timestamp="1112305882#22" type="string">DaffyDuck</
value>
         </add-attr>
         <add-attr attr-name="LastName">
            <value timestamp="1112305882#7" type="string">Duck</value>
         </add-attr>
         <add-attr attr-name="FirstName">
            <value timestamp="1112305882#5" type="string">Daffy</
value>
         </add-attr>
         <add-attr attr-name="InternetAddress">
            <value>DaffyDuck@novell.com</value>
         </add-attr>
      </add>
   </input>
</nds>
```

D.6.3 Sample Modify Event Policy Rule

To view sample code to Modify Event policy rule to submit database replication parameters, see setsyncdbrep.xml (../samples/setsyncdbrep.xml). You can use or modify the code sample in your own XML solutions.

D.6.4 Modify Event Attribute Tags As They Are Submitted To the Shim

Sample Modify Event containing database replication attribute tags as submitted to the Notes Driver shim. This sample modifies the synchronized .nsf database (in this case `names.nsf`):

```
<nds dtdversion="2.0" ndsversion="8.x">
   <source>
      <product version="2.0.8.20050127 ">Identity Manager</product>
      <contact>Novell, Inc.</contact>
   </source>
   <input>
      <modify     class-name="Group"
                  event-id="BLACKCAP-NDS#20050401191642#1#1"
                  qualified-src-
dn="O=DirXML\OU=Notes\OU=Groups\CN=Engineering"
                  rep-abstract="false"
                  rep-clear-history="false"
                  rep-cutoff-delete="false"
                  rep-cutoff-interval="240"
                  rep-dest-server="CN=server1/O=novell"
                  rep-disabled="false"
                  rep-dont-send-local-security-updates="true"
                  rep-entry-remove="false"
                  rep-ignore-deletes="false"
                  rep-ignore-dest-deletes="false"
                  rep-immediate="CN=server1/O=novell"
                  rep-include-acl="true"
                  rep-include-agents="true"
                  rep-include-documents="true"
                  rep-include-forms="true"
                  rep-include-formulas="true"
                  rep-priority="HIGH"
                  rep-src-server="CN=server2/O=novell"
                  rep-view-list="People;People By
Category;Groups;Groups By Category"
                  src-
dn="\novell_tree\DirXML\Notes\Groups\Engineering"
                  src-entry-id="40743"
                  timestamp="1112383002#1">
            <association
state="associated">3EEB6FC36CBE4D3687256FD60069C721</association>
            <modify-attr attr-name="ListDescription">
               <add-value>
                  <value timestamp="1112383002#1"
type="string">Software Engineering Group</value>
               </add-value>
            </modify-attr>
      </modify>
   </input>
</nds>
```

D.7 Sample ACL Entry Tags

The acl-entry-enable-role and acl-entry-disable-role tag values require a list of the roles that are defined in the ACL record. They also accept the [[ALL]] tag, which indicates using all of the roles defined in the ACL record.

You can select all roles with the string acl-entry-enable-role=“[[ALL]]”. This is equivalent to acl-entry-enable-role=“[GroupCreator] [GroupModifier] [NetCreator] [NetModifier] [PolicyCreator] [PolicyModifier] [PolicyReader] [ServerCreator] [ServerModifier] [UserCreator] [UserModifier]” for `names.nsf`.

You can deselect all roles with the string acl-entry-disable-role=“[[ALL]]”. This is equivalent to acl-entry-disable-role=“[GroupCreator] [GroupModifier] [NetCreator] [NetModifier] [PolicyCreator] [PolicyModifier] [PolicyReader] [ServerCreator] [ServerModifier] [UserCreator] [UserModifier]” for `names.nsf`.

- Section D.7.1, “ADD Event Policy Rule To Submit ACLEntry Parameters,” on page 195
- Section D.7.2, “The Add Event ACLEntry Tags That Are Submitted To the Notes Driver Shim,” on page 195
- Section D.7.3, “Sample Modify Event Policy Rule,” on page 196
- Section D.7.4, “Modify Event As Submitted To the Notes Driver Shim,” on page 198

D.7.1 ADD Event Policy Rule To Submit ACLEntry Parameters

To view sample code to ADD Event policy rule to submit ACLEntry parameters, see setsyncdbacladd.xml (../samples/setsyncdbacladd.xml). You can use or modify the code sample for your own XML solutions.

D.7.2 The Add Event ACLEntry Tags That Are Submitted To the Notes Driver Shim

Sample Add Event containing ACLEntry tags as they are submitted to the Notes Driver shim:

```
<nds dtdversion="2.0" ndsversion="8.x">
   <source>
      <product version="2.0.5.51 ">Identity Manager</product>
      <contact>Novell, Inc.</contact>
   </source>
   <input>
      <add acl-entry-can-create-documents="true"
       acl-entry-can-create-ls-or-java-agent="true"
       acl-entry-can-create-personal-agent="true"
       acl-entry-can-create-personal-folder="true"
       acl-entry-can-create-shared-folder="true"
       acl-entry-can-delete-documents="true"
       acl-entry-can-replicate-or-copy-documents="true"
       acl-entry-enable-role="[[ALL]]"
       acl-entry-level="MANAGER"
       acl-entry-public-reader="true"
       acl-entry-public-writer="true"
       acl-entry-user-type="PERSON"
       certify-user="true"
       class-name="Person"
       create-mail="true"
       dest-dn="CN=DaffyDuck/OU=sales/O=novell"
       drv-param-cert-id="sales-cert-id-file"
       drv-param-cert-pwd="sales-cert-id-password"
       event-id="MYSERVER-NDS#20040920214955#1#1"
       expire-term="2"
       mail-acl-manager-name="CN=Notes Driver/O=novell"
       qualified-src-
dn="O=DirXML\OU=Notes\OU=Users\OU=sales\CN=DaffyDuck"
       src-dn="\mytree\DirXML\Notes\Users\sales\DaffyDuck"
       src-entry-id="39862">
         <add-attr attr-name="FullName">
            <value naming="true" timestamp="1095716982#20"
             type="string">DaffyDuck</value>
         </add-attr>
         <add-attr attr-name="LastName">
            <value timestamp="1095716982#3" type="string">Duck</value>
         </add-attr>
         <add-attr attr-name="FirstName">
            <value timestamp="1095716995#1" type="string">Daffy</
value>
         </add-attr>
         <add-attr attr-name="InternetAddress">
            <value>DaffyDuck@novell.com</value>
         </add-attr>
      </add>
   </input>
</nds>
```

D.7.3 Sample Modify Event Policy Rule

Below is a sample Modify Event policy rule to submit ACLEntry parameters to the Notes Driver shim:

```
<rule>
   <description>Apply ACL entry attributes to MODIFY events</
description>
   <conditions>
      <or disabled="true">
         <if-operation op="equal">modify</if-operation>
      </or>
   </conditions>
   <actions>
      <do-set-xml-attr expression="../modify" name="acl-entry-public-
reader">
         <arg-string>
            <token-text>true</token-text>
         </arg-string>
      </do-set-xml-attr>
      <do-set-xml-attr expression="../modify" name="acl-entry-public-
writer">
         <arg-string>
            <token-text>true</token-text>
         </arg-string>
      </do-set-xml-attr>
      <do-set-xml-attr expression="../modify" name="acl-entry-level">
         <arg-string>
            <token-text>MANAGER</token-text>
         </arg-string>
      </do-set-xml-attr>
      <do-set-xml-attr expression="../modify" name="acl-entry-user-
type">
         <arg-string>
            <token-text>PERSON</token-text>
         </arg-string>
      </do-set-xml-attr>
      <do-set-xml-attr expression="../modify" name="acl-entry-enable-
role">
         <arg-string>
            <token-text>[[ALL]]</token-text>
         </arg-string>
      </do-set-xml-attr>
      <do-set-xml-attr expression="../modify" name="acl-entry-disable-
role">
         <arg-string>
            <token-text xml:space="preserve">[NetCreator]
[NetModifier]</token-text>
         </arg-string>
      </do-set-xml-attr>
      <do-set-xml-attr expression="../modify" name="acl-entry-can-
create-documents">
         <arg-string>
            <token-text>true</token-text>
         </arg-string>
      </do-set-xml-attr>
      <do-set-xml-attr expression="../modify" name="acl-entry-can-
create-ls-or-java-agent">
         <arg-string>
            <token-text>true</token-text>
         </arg-string>
      </do-set-xml-attr>
      <do-set-xml-attr expression="../modify" name="acl-entry-can-
create-personal-agent">
         <arg-string>
            <token-text>true</token-text>
         </arg-string>
      </do-set-xml-attr>
      <do-set-xml-attr expression="../modify" name="acl-entry-can-
create-personal-folder">
          <arg-string>
            <token-text>true</token-text>
         </arg-string>
      </do-set-xml-attr>
      <do-set-xml-attr expression="../modify" name="acl-entry-can-
create-shared-folder">
         <arg-string>
            <token-text>true</token-text>
         </arg-string>
      </do-set-xml-attr>
      <do-set-xml-attr expression="../modify" name="acl-entry-can-
delete-documents">
         <arg-string>
            <token-text>true</token-text>
         </arg-string>
      </do-set-xml-attr>
      <do-set-xml-attr expression="../modify" name="acl-entry-can-
replicate-or-copy-documents">
         <arg-string>
            <token-text>true</token-text>
         </arg-string>
      </do-set-xml-attr>
   </actions>
</rule>
```

D.7.4 Modify Event As Submitted To the Notes Driver Shim

Below shows the Modify Event containing ACLEntry tags as they are submitted to the Notes Driver shim:

```
<nds dtdversion="2.0" ndsversion="8.x">
   <source>
      <product version="2.0.5.51 ">Identity Manager</product>
      <contact>Novell, Inc.</contact>
   </source>
   <input>
      <modify acl-entry-can-create-documents="true"
       acl-entry-can-create-ls-or-java-agent="true"
       acl-entry-can-create-personal-agent="true"
       acl-entry-can-create-personal-folder="true"
       acl-entry-can-create-shared-folder="true"
       acl-entry-can-delete-documents="true"
       acl-entry-can-replicate-or-copy-documents="true"
       acl-entry-disable-role="[NetCreator] [NetModifier]"
       acl-entry-enable-role="[[ALL]]"
       acl-entry-level="MANAGER"
       acl-entry-public-reader="true"
       acl-entry-public-writer="true"
       acl-entry-user-type="PERSON" class-name="Person"
       event-id="MYSERVER-NDS#20040920215410#1#1"
       qualified-src-
dn="O=DirXML\OU=Notes\OU=Users\OU=sales\CN=DaffyDuck"
       src-dn="\mytree\DirXML\Notes\Users\sales\DaffyDuck"
       src-entry-id="39862" timestamp="1095717426#2">
         <association
state="associated">BE64D2CAAB6EADD987256F150077EF7B</association>
         <modify-attr attr-name="OfficePhoneNumber">
            <remove-value>
               <value timestamp="1095717250#1" type="teleNumber">444-
4444</value>
            </remove-value>
            <add-value>
```

```
               <value timestamp="1095717426#2" type="teleNumber">555-
1212</value>
            </add-value>
         </modify-attr>
      </modify>
   </input>
</nds>
```

D.8 Setting and Modifying Lotus Notes Field Flags

The Notes Driver v2.1.1 and above can set (add) or modify Lotus Notes field flags on documents (records) in a Lotus Notes database (.nsf) that is being synchronized (Subscriber channel only). Available Lotus Notes field flags which the driver can appropriately set are *read-access*, *read/write-access*, *names*, *protected*, and *summary*. The *seal* and *sign* flags can also be enabled or disabled, but without the expected supporting functionality.

Each Notes field flag can be set enabled or disabled using a corresponding XML tag. The following table represents this mapping.

Table D-1 *Driver Attributes Tags and Their Corresponding Notes Field Flag*

driver-attr-flag	Notes Field Flag
authors-flag	READ/WRITE-ACCESS
encrypted-flag	SEAL
names-flag	NAMES
protected-flag	PROTECTED
readers-flag	READ-ACCESS
signed-flag	SIGN
summary-flag	SUMMARY

You can insert these XML tags (driver-attr-flags) into XDS documents of the Subscriber channel as attributes of the <add-attr> or <modify-attr> elements (siblings to the attr-name attribute). The field flags must be used in the appropriate manner, according to Lotus Notes database schema and design rules. Inappropriate flags (or flag combinations) on a field can cause unexpected results for that document (record).

For example, it is possible for uncertified users to be removed from the view of the Notes Address book (rendering them inaccessible) when certain attribute flags (readers-flag, authors-flag) are set inappropriately on a field that cannot handle the flag.

The encrypted-flag can be set for a field, even though the driver does not take the necessary steps to call the encryption methods with appropriate certificates. So it is possible to set a field as *SEAL* but some other code will need to call the appropriate methods to encrypt the field in the document, or the field will not truly be sealed (encrypted).

- Section D.8.1, “Sample Creation Policy Rules,” on page 200

- Section D.8.2, “A Sample Modify Policy Rule,” on page 200
- Section D.8.3, “Example Add XDS Doc,” on page 200
- Section D.8.4, “Example Modify XDS Doc,” on page 201

D.8.1 Sample Creation Policy Rules

To view an example of using the field flags in creation policy rules, see setfieldflagsadd.xml (../samples/setfieldflagsadd.xml). You can use or modify the code sample for your own XML solutions.

D.8.2 A Sample Modify Policy Rule

To view an example of a Modify policy rule in the command transform, see setfieldflagsmodify.xml (../samples/setfieldflagsmodify.xml). You can use or modify the code sample for your own XML solutions.

D.8.3 Example Add XDS Doc

Below is an example Add XDS doc before it is submitted to the Notes Driver shim:

```
<nds dtdversion="2.0" ndsversion="8.x">
   <source>
      <product version="2.0.5.51 ">Identity Manager</product>
      <contact>Novell, Inc.</contact>
   </source>
   <input>
      <add certify-user="true"
       class-name="Person"
       create-mail="true"
       dest-dn="CN=ErnieEngineer/OU=eng/O=novell"
       drv-param-cert-id="eng-cert-id-file"
       drv-param-cert-pwd="eng-cert-id-password"
       event-id="BLACKCAP-NDS#20040915163542#1#1"
       expire-term="22"
       internet-password-force-change="false"
       mail-acl-manager-name="CN=Notes Driver/O=novell"
       mail-file-inherit-flag="true"
       no-id-file="false"
       notes-password-change-interval="0"
       notes-password-check-setting="PWD_CHK_CHECKPASSWORD"
       notes-password-grace-period="0"
       qualified-src-
dn="O=DirXML\OU=Notes\OU=Users\OU=eng\CN=ErnieEngineer"
       roaming-cleanup-period="90"
       roaming-cleanup-setting="REG_ROAMING_CLEANUP_EVERY_NDAYS"
       roaming-server="CN=blackcap/O=novell"
       roaming-subdir="Roaming\ErnieEngineer"
       roaming-user="false" src-
dn="\raspberry\DirXML\Notes\Users\eng\ErnieEngineer"
       src-entry-id="39853"
       store-useridfile-in-ab="true"
       sync-internet-password="true">
         <add-attr attr-name="FullName">
```

```
            <value naming="true" timestamp="1095266118#20"
type="string">ErnieEngineer</value>
         </add-attr>
         <add-attr attr-name="LastName">
            <value timestamp="1095266118#3" type="string">Engineer</
value>
         </add-attr>
         <add-attr attr-name="FirstName">
            <value timestamp="1095266142#1" type="string">Ernie</
value>
         </add-attr>
         <add-attr attr-name="InternetAddress">
            <value>ErnieEngineer@novell.com</value>
         </add-attr>
         <add-attr attr-name="ShoeSize" authors-flag="false"
encrypted-flag="false" names-flag="false" protected-flag="true"
readers-flag="false" signed-flag="false" summary-flag="true">
            <value type="string">9.5</value>
         </add-attr>
      </add>
   </input>
</nds>
```

D.8.4 Example Modify XDS Doc

Below is an example Modify XDS doc before it is submitted to the Notes Driver shim:

```
<nds dtdversion="2.0" ndsversion="8.x">
   <source>
      <product version="2.0.5.51 ">Identity Manager</product>
      <contact>Novell, Inc.</contact>
   </source>
   <input>
      <modify class-name="Person"
       event-id="BLACKCAP-NDS#20040915164613#1#1"
       qualified-src-
dn="O=DirXML\OU=Notes\OU=Users\OU=eng\CN=ErnieEngineer"
       src-dn="\raspberry\DirXML\Notes\Users\eng\ErnieEngineer"
       src-entry-id="39853" tell-adminp-process="tell adminp process
all" timestamp="1095267005#2">
         <association
state="associated">A4C23EE8273577AF87256F10005B2BF9</association>
         <modify-attr attr-name="OfficePhoneNumber"
          authors-flag="false"
          encrypted-flag="false"
          names-flag="false"
          protected-flag="true"
          readers-flag="false"
          signed-flag="false"
          summary-flag="true">
            <remove-value>
               <value timestamp="1095266773#1" type="teleNumber">222-
2222</value>
            </remove-value>
            <add-value>
               <value timestamp="1095267005#2" type="teleNumber">222-
2221</value>
            </add-value>
         </modify-attr>
      </modify>
   </input>
</nds>
```