Driver for Lotus Notes* Implementation Guide

Novell®
Identity Manager

3.6

July 23, 2008

www.novell.com

Novell Trademarks

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

Third-Party Materials

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

Contents

About This Guide

The Identity Manager Driver for Lotus Notes is designed to 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 http://www.novell.com/documentation/feedback.html and enter your comments there.

Documentation Updates

For the most recent version of this document, see the Identity Manager 3.6 Drivers documentation Web site (http://www.novell.com/documentation/idm36drivers/index.html).

Additional Documentation

For documentation on using Identity Manager and the other drivers, see the Identity Manager 3.6 Documentation Web site (http://www.novell.com/documentation/idm36drivers/index.html).

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

<div style="text-align: right; font-size: 3em;">1</div>

The Identity Manager Driver for Lotus Notes (Lotus Notes driver) lets you synchronize data in the Identity Vault with data stored in a Domino Directory or another Notes database.

The Lotus Notes driver 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, "Supported Lotus Notes Versions," on page 11
- Section 1.2, "Lotus Notes Driver Concepts," on page 11
- Section 1.3, "Support for Standard Driver Features," on page 13

1.1 Supported Lotus Notes Versions

You can use the Lotus Notes driver with Lotus Notes v6.x, v7.x, and v8.0.

1.2 Lotus Notes Driver Concepts

The following sections explain concepts you should understand before implementing the Notes driver:

- Section 1.2.1, "Default Data Flow," on page 11
- Section 1.2.2, "Policies," on page 12
- Section 1.2.3, "Driver Components," on page 13

1.2.1 Default Data Flow

A channel is a combination of rules, policies, and filters that is used to synchronize data between two systems. The Subscriber and Publisher channels describe the direction in which the data flows. The Subscriber and Publisher channels act independently; actions in one channel are not affected by what happens in the other.

- "Subscriber Channel" on page 11
- "Publisher Channel" on page 12

Subscriber Channel

The Subscriber channel is the channel of communication from the Identity Vault to Lotus Notes. The channel takes events generated in the Identity Vault and sends them to the Lotus Notes system. The following illustration shows this data flow:

Figure 1-1 *Data Flow Through the Subscriber Channel*
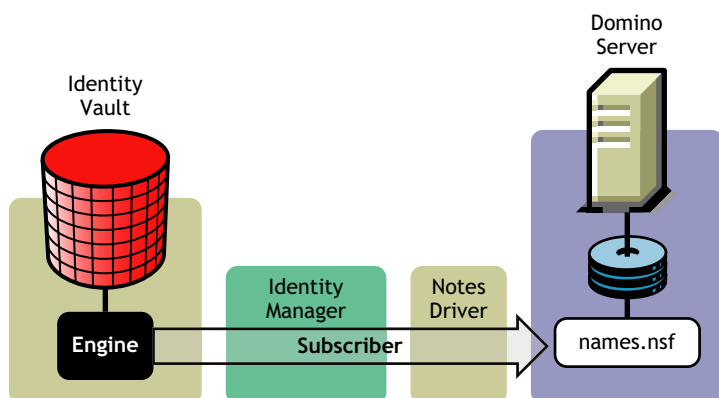
The driver can be configured to work with Notes databases other than the Public Address Book, `names.nsf`.

Publisher Channel

The Publisher channel represents the channel of communication from Lotus Notes to the Identity Vault. The channel takes event generated in Lotus Notes and sends them to the Identity Vault. The following illustration shows how this data is published:

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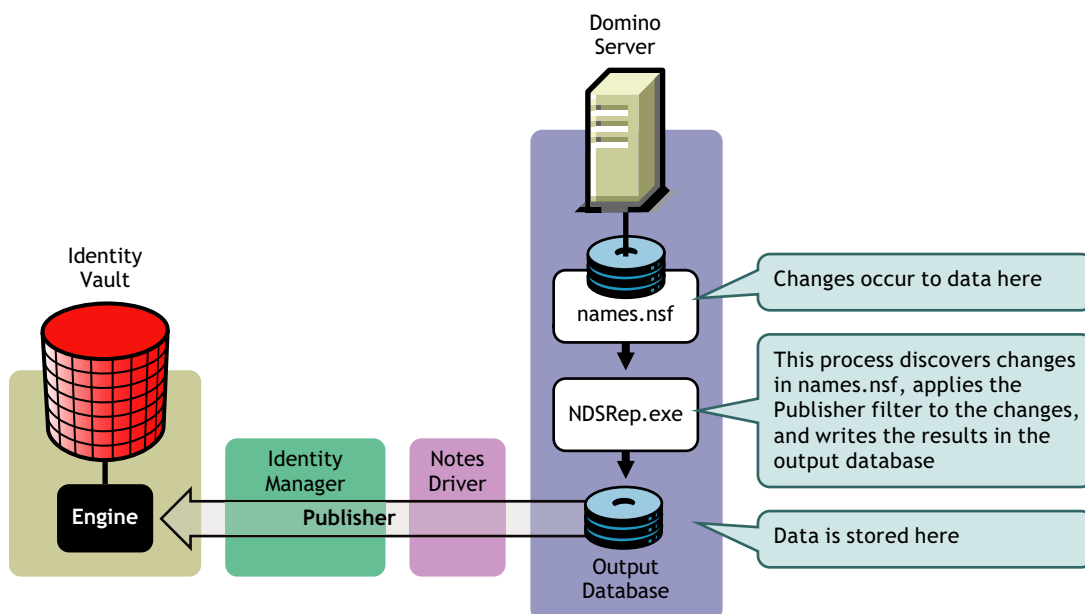
1.2.2 Policies

Policies are used to control the synchronization of data between the Identity Vault and Lotus Notes. 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 Identity Vault User object is considered to be the same object in Notes when Given name and Surname match in both directories. An Identity Vault Group object is considered to be the same object in Notes when the CN is the same in both directories. You should 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. You should modify or add additional Placement policies and policy rules to meet your business needs.

1.2.3 Driver Components

The driver contains the following components:

- **Default Driver Configuration File:** 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_6-V3.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:** 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`l, which is used by ndsrep to determine the Publisher filter and other driver publication settings.

1.3 Support for Standard Driver Features

The following sections provide information about how the Lotus Notes driver supports these standard driver features:

1.3.1 Local Platforms

A local installation is an installation of the driver on the Metadirectory server. The Lotus Notes driver can be installed on the operating systems supported for the Metadirectory server.

For information about the operating systems supported for the Metadirectory server, see "Metadirectory Server" in "System Requirements" in the *Identity Manager 3.6 Installation Guide*.

1.3.2 Remote Platforms

The Lotus Notes driver can use the Remote Loader service to run on a server other than the Metadirectory server. For example, you might not want to install the Metadirectory server (Metadirectory engine and Identity Vault) on the same server as Lotus Notes. In this case, you install the Remote Loader and driver on the Lotus Notes server and the Remote Loader enables the driver to communicate with the Metadirectory server.

For information about the operating systems supported for the Remote Loader, see "Remote Loader" in "System Requirements" in the *Identity Manager 3.6 Installation Guide*.

1.3.3 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 or role-based provisioning or the Entitlements Service driver, to manage entitlement functionality.

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

Installing the Driver Files

2

There are several installation scenarios you can use to best meet the needs of your environment. The following sections explain the scenarios and provide instructions for installing the files based upon the scenario you've chosen.

2.1 Where to Install the Lotus Notes Driver

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

2.1.1 Local Installation

In a local installation, the Lotus Notes driver is on the same server as the Metadirectory engine, Identity Vault, and Lotus Domino server.

Figure 2-1 *Local Installation*

Remote Loader
Identity Vault
Metadirectory Engine
Identity Manager Driver
  for Lotus Notes
Lotus Domino
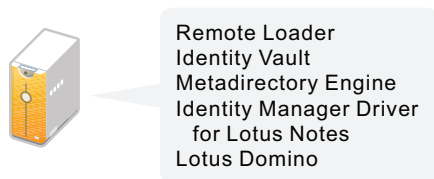
You should always use the Remote Loader with the Lotus Notes driver, even in a local installation like this one. Using the Remote Loader has the following advantages:

- An exception in the driver caused by the Lotus Notes APIs doesn't cause the Identity Vault (eDirectory™) to shut down.
- The driver has its own Java memory space, which helps avoid heap issues.

2.1.2 Local Installation on a Notes Client Workstation

In a local installation on a Lotus Notes client workstation, the Lotus Notes driver is on the same workstation as the Metadirectory engine, Identity Vault, and Lotus Notes client. The Lotus Domino server is a separate server. The Lotus Notes driver communicates with the Domino server through the Lotus Notes client. This option is the recommended installation practice.

Figure 2-2   *Local Installation on a Notes Client Workstation*
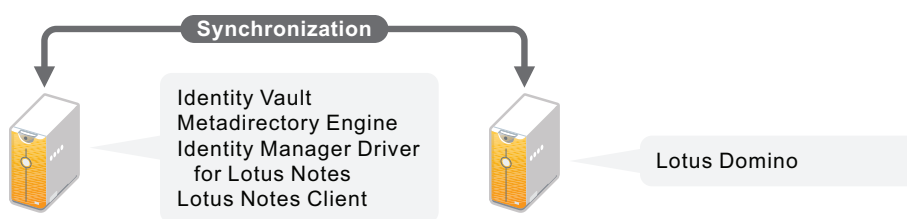
2.1.3 Remote Installation

In a remote installation, the Lotus Notes driver is on the Domino server and the Metadirectory engine and Identity Vault are on a separate Metadirectory server. The driver uses the Remote Loader, also installed on the Domino server, to communicate with the Metadirectory engine.

Figure 2-3   *Remote Installation*
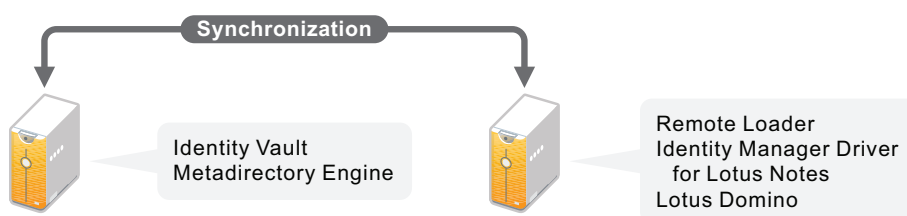
2.2 Installing the Driver Files

The following sections correspond to the scenarios in Section 2.1, "Where to Install the Lotus Notes Driver," on page 15. Complete the steps for the scenario you've chosen:

- Section 2.2.1, "Local Installation," on page 16
- Section 2.2.2, "Local Installation on a Notes Client Workstation," on page 17
- Section 2.2.3, "Remote Installation on a Lotus Domino Server," on page 17

2.2.1 Local Installation

Complete the following steps for a local installation. In this scenario, the Metadirectory engine, Identity Vault, Lotus Notes driver, and Lotus Domino are all on the same server.

1 Install the Metadirectory server (Metadirectory engine and drivers) on the Lotus Domino server. For instructions, see "Installing the Metadirectory Server" in the *Identity Manager 3.6 Installation Guide*.

2 If you want to use the Remote Loader with the driver, install the Remote Loader on the Lotus Domino server. For instructions, see "Installing the Remote Loader" in the *Identity Manager 3.6 Installation Guide*.

3 Continue with Section 2.3, "Copying Ndsrep and Notes.jar to the Required Locations," on page 17 to complete installation of the files.

2.2.2 Local Installation on a Notes Client Workstation

Complete the following steps for a local installation on a Notes client workstation. In this scenario, the Metadirectory engine, Identity Vault, Lotus Notes driver, and Notes client are all on the same workstation while the Lotus Domino server is a separate server.

1 Install the Metadirectory server (Metadirectory engine and drivers) on the Notes client workstation. For instructions, see "Installing the Metadirectory Server" in the *Identity Manager 3.6 Installation Guide*.

2 If you want to use the Remote Loader with the driver, install the Remote Loader on the Lotus Domino server. For instructions, see "Installing the Remote Loader" in the *Identity Manager 3.6 Installation Guide*.

3 Continue with Section 2.3, "Copying Ndsrep and Notes.jar to the Required Locations," on page 17 to complete installation of the files.

2.2.3 Remote Installation on a Lotus Domino Server

Complete the following steps for a remote installation on a Lotus Domino server. In this scenario, the Metadirectory engine and Identity Vault are on one server and the Remote Loader and Lotus Notes driver are on a separate Notes Domino server.

1 If you have not already done so, install a Metadirectory server. For instructions, see "Installing the Metadirectory Server" in the *Identity Manager 3.6 Installation Guide*.

2 Install the Remote Loader on the Lotus Domino server. For instructions, see "Installing the Remote Loader" in the *Identity Manager 3.6 Installation Guide*.

You install the Lotus Notes driver as part of the Remote Loader installation.

3 Continue with Section 2.3, "Copying Ndsrep and Notes.jar to the Required Locations," on page 17 to complete installation of the files.

2.3 Copying Ndsrep and Notes.jar to the Required Locations

After the Lotus Notes driver files are installed, you need to copy ndsrep (and its support files) to the Lotus Notes directory and copy the Lotus Notes JAR file to the driver directory.

For information about ndsrep, see "Publisher Channel" on page 12.

* Section 2.3.1, "Windows Domino Server," on page 17
* Section 2.3.2, "Notes Client Workstation," on page 18
* Section 2.3.3, "Linux, AIX, or Solaris," on page 19

2.3.1 Windows Domino Server

1 Manually copy the following files:

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`	The Domino Server executable folder:
	
	`\Lotus\Domino`
		The Remote Loader:
		
		`\Novell\RemoteLoader\lib`
		
		or
		
		The Metadirectory server (without a Remote Loader):
		
		`\Novell\NDS\lib`

2 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`.

3 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.

2.3.2 Notes Client Workstation

1 Manually copy the following files:

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`

Filename	Copy from	Copy to
Notes.jar	The Notes Client workstation:	The Remote Loader:
	\Lotus\Domino	\Novell\RemoteLoader\lib
	or	or
	\Lotus\Domino\jvm\lib\ext	The Metadirectory server (without a Remote Loader):
		\Novell\NDS\lib

2 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.

3 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.

2.3.3 Linux, AIX, or Solaris

1 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 or newer, 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`

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

3 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.

4 Modify the scripts and configuration file to fit to your environment, as described in the table in Step 3.

5 Make sure that the three scripts noted in Step 3 have file access for execution (for example, rwxr-xr-x).

For troubleshooting tips, see "Troubleshooting Installation Problems" on page 92.

Creating a New Driver

<div style="text-align: right; font-size: large;">3</div>

After the Lotus Notes driver files are installed on the server where you want to run the driver (see Chapter 2, "Installing the Driver Files," on page 15), you can create the driver in the Identity Vault. You do so by importing the basic driver configuration file and then modifying the driver configuration to suit your environment. The following sections provide instructions:

- Section 3.1, "Creating Lotus Notes Accounts and Groups," on page 21
- Section 3.2, "Creating the Driver in Designer," on page 21
- Section 3.3, "Creating the Driver in iManager," on page 25
- Section 3.4, "Activating the Driver," on page 30

3.1 Creating Lotus Notes Accounts and Groups

The driver requires an account to access the Lotus Notes system. It also requires a Deny Access group to exist in the Lotus Notes system.

1. Create a Lotus Notes user account to be used exclusively by the driver. Do the following when you create the account:

    - Assign 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 information from the Identity Vault to the Notes database (`names.nsf`), give the user all ACL roles (GroupCreator, GroupModifier, NetCreator, NetModifier, PolicyCreator, PolicyModifier, PolicyReader, ServerCreator, ServerModifier, UserCreator, UserModifier).
    - Assign access rights to the user account's certifier ID file.
    - Assign access rights to the certifier ID files for the Notes certifiers that you want the driver to create users for.
    - (Optional) If you want the driver to be able to create new user certifier ID files, assign file access rights to the location where you want the driver to create the files.

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.

3. The installation procedure should take care of Universal ID issues. If you are having problems with Universal IDs, see Section 7.2.1, "Creating Lotus Notes Accounts and Groups," on page 91.

3.2 Creating the Driver in Designer

You create the Lotus Notes driver by importing the driver's basic configuration file and then modifying the configuration to suit your environment. After you've created and configured the driver, you need to deploy it to the Identity Vault and start it.

- Section 3.2.1, "Importing the Driver Configuration File," on page 22
- Section 3.2.2, "Configuring the Driver," on page 22

3.2.1 Importing the Driver Configuration File

1 In Designer, open your project.

2 In the Modeler, right-click the driver set where you want to create the driver, then select *New > Driver* to display the Driver Configuration Wizard.

3 In the Driver Configuration list, select *Notes*, then click *Run*.

4 On the Import Information Requested page, fill in the following fields:

Driver Name: Specify a name that is unique within the driver set.

Driver is Local/Remote: Select *Local* if this driver will run on the Metadirectory server without using the Remote Loader service. Select *Remote* if you want the driver to use the Remote Loader service, either locally on the Metadirectory server or remotely on another server.

5 (Conditional) If you chose to run the driver remotely, click *Next*, then fill in the fields listed below. Otherwise, skip to Step 6.

Remote Host Name and Port: Specify the host name or IP address of the server where the driver's Remote Loader service is running.

Driver Password: Specify the driver object password that is defined in the Remote Loader service. The Remote Loader requires this password to authenticate to the Metadirectory server.

Remote Password: Specify the Remote Loader's password (as defined on the Remote Loader service). The Metadirectory engine (or Remote Loader shim) requires this password to authenticate to the Remote Loader

6 Click *Next* to import the driver configuration.

At this point, the driver is created from the basic configuration file. To ensure that the driver works the way you want it to for your environment, you must review and modify (if necessary) the driver's default configuration settings.

7 To review or modify the default configuration settings, click *Configure*, then continue with the next section, Configuring the Driver.

or

To skip the configuration settings at this time, click *Close*. When you are ready to configure the settings, continue with Configuring the Driver.

3.2.2 Configuring the Driver

After importing the driver configuration file, you need to configure the driver before it can run. You should complete the following tasks to configure the driver:

◆ **Ensure that the driver can authenticate to Lotus Notes:** Make sure that you've established a Lotus Notes user account for the driver (see Section 3.1, "Creating Lotus Notes Accounts and Groups," on page 21) and the correct authentication information is defined for the driver parameters (see Section A.1.3, "Authentication," on page 96).

- **Configure the driver parameters:** There are many settings that can help you customize and optimize the driver. The settings are divided into categories such as Driver Configuration, Engine Control Values, and Global Configuration Values (GCVs). Although it is important for you to understand all of the settings, your first priority should be to configure the driver parameters located on the Driver Configuration page. The driver parameters control general settings such as the certifier file and Lotus Notes database accessed by the driver, Subscriber channel settings, and Publisher channel settings. For information about the driver parameters, see Section A.1.5, "Driver Parameters," on page 98. For additional information, see Section 5.4, "Using Driver Parameters," on page 36.

- **Configure the driver filter:** Modify the driver filter to include the object classes and attributes you want synchronized between the Identity Vault and Lotus Notes.

- **Configure policies:** Modify the policies as needed. For information about the default configuration policies, see "Policies" on page 12.

- **Configure password synchronization:** The basic driver configuration is set up to support password synchronization through Universal Password. If you don't want this setup, see "Configuring Password Flow " in the *Identity Manager 3.6 Password Management Guide*.

After completing the configuration tasks, continue with the next section, Deploying the Driver.

3.2.3 Deploying the Driver

After a driver is created in Designer, it must be deployed into the Identity Vault.

1 In Designer, open your project.

2 In the Modeler, right-click the driver icon or the driver line, then select *Live > Deploy*.

3 If you are authenticated to the Identity Vault, skip to Step 5; otherwise, specify the following information:

- **Host:** Specify the IP address or DNS name of the server hosting the Identity Vault.

- **Username:** Specify the DN of the user object used to authenticate to the Identity Vault.

- **Password:** Specify the user's password.

4 Click *OK*.

5 Read the deployment summary, then click *Deploy*.

6 Read the message, then click *OK*.

7 Click *Define Security Equivalence* to assign rights to the driver.

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

7a Click *Add*, then browse to and select the object with the correct rights.

7b Click *OK* twice.

8 Click *Exclude Administrative Roles* to exclude users that should not be synchronized.

You should exclude any administrative User objects (for example, Admin and DriversUser) from synchronization.

8a Click *Add*, then browse to and select the user object you want to exclude.

8b Click *OK*.

8c Repeat Step 8a and Step 8b for each object you want to exclude.

8d Click *OK*.

9 Click *OK*.

3.2.4 Starting the Driver

When a driver is created, it is stopped by default. To make the driver work, you must start the driver and cause events to occur. Identity Manager is an event-driven system, so after the driver is started, it won't do anything until an event occurs.

To start the driver:

1 If you are using the Remote Loader with the driver, make sure the Remote Loader driver instance is running:

* **Windows:** Open the Remote Loader console on the driver's server, select the driver in the list, then click *Start*.

* **Linux, AIX, or Solaris:** At the server command line, add the driver object password and remote loader password to the Notes driver startup script. For example:

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

These passwords must be the same as the Driver Object Password and Remote Loader Password you specified when configuring the driver. The `driver_script_directory` should be the directory where you placed the files (see Section 2.3.3, "Linux, AIX, or Solaris," on page 19).

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

For example,

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

2 In Designer, open your project.

3 In the Modeler, right-click the driver icon or the driver line, then select *Live > Start Driver*.

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

* Searches for the Domino Server (specified in the driver parameters).

* 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.

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.

4 Continue with the next section, Starting Ndsrep.

3.2.5 Starting Ndsrep

The ndsrep program is used to publish information from the Notes database to the Identity Vault (Publisher channel). It is not required if you are only sending information from the Identity Vault to the Notes database (Subscriber channel).

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.

You should have already copied ndsrep to the Lotus Domino server (see Section 2.3, "Copying Ndsrep and Notes.jar to the Required Locations," on page 17).

1 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.

2 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 additional information about using ndsrep, see Appendix C, "Ndsrep," on page 109.

3.3 Creating the Driver in iManager

You create the Notes driver by importing the driver's basic configuration file and then modifying the configuration to suit your environment. After you've created and configured the driver, you need to start it.

- Section 3.3.1, "Importing the Driver Configuration File," on page 26
- Section 3.3.2, "Configuring the Driver," on page 27
- Section 3.3.3, "Starting the Driver," on page 28
- Section 3.3.4, "Starting Ndsrep," on page 29

3.3.1 Importing the Driver Configuration File

1 In iManager, click to display the Identity Manager Administration page.

2 In the Administration list, click *Import Configuration* to launch the Import Configuration wizard.

3 Follow the wizard prompts, filling in the requested information (described below) until you reach the Summary page.

Prompt	Description
Where do you want to place the new driver?	You can add the driver to an existing driver set, or you can create a new driver set and add the driver to the new set. If you choose to create a new driver set, you are prompted to specify the name, context, and server for the driver set.
Import a configuration into this driver set	Use the default option, *Import a configuration from the server (.XML file)*. In the *Show* field, select *Identity Manager 3.6 configurations*. In the *Configurations* field, select the Notes file.
Driver name	Type a name for the driver. The name must be unique within the driver set.
Driver is Local/Remote	Select *Local* if this driver will run on the Metadirectory server without using the Remote Loader service. Select *Remote* if you want the driver to use the Remote Loader service, either locally on the Metadirectory server or remotely on another server.
Remote Host Name and Port	This applies only if the driver is running remotely. Specify the host name or IP address of the server where the driver's Remote Loader service is running.
Driver Password	This applies only if the driver is running remotely. Specify the driver object password that is defined in the Remote Loader service. The Remote Loader requires this password to authenticate to the Metadirectory server.
Remote Password	This applies only if the driver is running remotely. Specify the Remote Loader's password (as defined on the Remote Loader service). The Metadirectory engine (or Remote Loader shim) requires this password to authenticate to the Remote Loader
Define Security Equivalences	The driver requires rights to objects within the Identity Vault. The Admin user object is most often used to supply these rights. However, you might want to create a DriversUser (for example) and assign security equivalence to that user. Whatever rights that the driver needs to have on the server, the DriversUser object must have the same security rights.
Exclude Administrative Roles	You should exclude any administrative User objects (for example, Admin and DriversUser) from synchronization.

When you finish providing the information required by the wizard, a Summary page similar to the following is displayed.
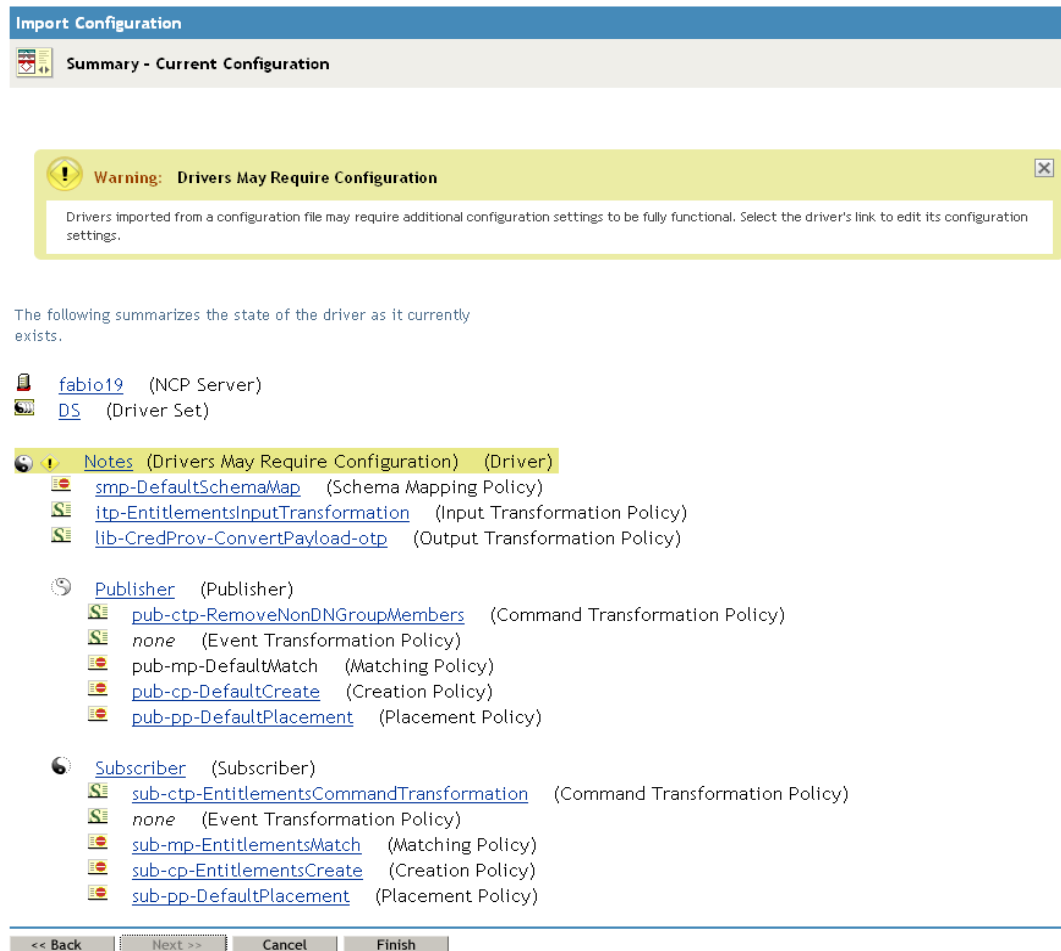
At this point, the driver is created from the basic configuration file. To ensure that the driver works the way you want it to for your environment, you must review and modify the driver's default configuration settings.

4 To modify the default configuration settings, click the linked driver name, then continue with the next section, Configuring the Driver.

or

To skip the configuration settings at this time, click *Finish*. When you are ready to configure the settings, continue with Configuring the Driver.

3.3.2 Configuring the Driver

After importing the driver configuration file, you need to configure the driver before it can run. You should complete the following tasks to configure the driver:

◆ **Ensure that the driver can authenticate to Lotus Notes:** Make sure that you've established a Lotus Notes user account for the driver (see Section 3.1, "Creating Lotus Notes Accounts and Groups," on page 21) and the correct authentication information is defined for the driver parameters (see Section A.1.3, "Authentication," on page 96).

- **Configure the driver parameters:** There are many settings that can help you customize and optimize the driver. The settings are divided into categories such as Driver Configuration, Engine Control Values, and Global Configuration Values (GCVs). Although it is important for you to understand all of the settings, your first priority should be to configure the driver parameters located on the Driver Configuration page. The driver parameters control general settings such as the certifier file and Lotus Notes database accessed by the driver, Subscriber channel settings, and Publisher channel settings. For information about the driver parameters, see Section A.1.5, "Driver Parameters," on page 98. For additional information, see Section 5.4, "Using Driver Parameters," on page 36.

- **Configure the driver filter:** Modify the driver filter to include the object classes and attributes you want synchronized between the Identity Vault and Lotus Notes.

- **Configure policies:** Modify the policies as needed. For information about the default configuration policies, see "Policies" on page 12.

- **Configure password synchronization:** The basic driver configuration is set up to support password synchronization through Universal Password. If you don't want this setup, see "Configuring Password Flow " in the *Identity Manager 3.6 Password Management Guide*.

After completing the configuration tasks, continue with the next section, Starting the Driver.

3.3.3 Starting the Driver

When a driver is created, it is stopped by default. To make the driver work, you must start the driver and cause events to occur. Identity Manager is an event-driven system, so after the driver is started, it won't do anything until an event occurs.

To start the driver:

1 If you are using the Remote Loader with the driver, make sure the Remote Loader driver instance is running:

- **Windows:** Open the Remote Loader console on the driver's server, select the driver in the list, then click *Start*.

- **Linux, AIX, or Solaris:** At the server command line, add the driver object password and remote loader password to the Notes driver startup script. For example:

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

These passwords must be the same as the Driver Object Password and Remote Loader Password you specified when configuring the driver. The *driver_script_directory* should be the directory where you placed the files (see Section 2.3.3, "Linux, AIX, or Solaris," on page 19).

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

For example,

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

2 In iManager, click to display the Identity Manager Administration page.

3 Click *Identity Manager Overview*.

4 Browse to and select the driver set object that contains the driver you want to start.

5 Click the driver set name to access the Driver Set Overview page.

6 Click the upper right corner of the driver, then click *Start driver*.

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

- ◆ Searches for the Domino Server (specified in the driver parameters).
- ◆ 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.

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 system path to find the appropriate `notes.ini` file.

7 Continue with the next section, Starting Ndsrep.

3.3.4 Starting Ndsrep

The ndsrep program is used to publish information from the Notes database to the Identity Vault (Publisher channel). It is not required if you are only sending information from the Identity Vault to the Notes database (Subscriber channel).

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.

You should have already copied ndsrep to the Lotus Domino server (see Section 2.3, "Copying Ndsrep and Notes.jar to the Required Locations," on page 17).

1 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.

2 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 additional information about using ndsrep, see .

3.4 Activating the Driver

If you created the driver in a driver set where you've already activated the Metadirectory engine and service drivers, the driver inherits the activation. If you created the driver in a driver set that has not been activated, you must activate the driver within 90 days. Otherwise, the driver stops working.

For information on activation, refer to "Activating Novell Identity Manager Products" in the *Identity Manager 3.6 Installation Guide*.

Upgrading an Existing Driver 4

The following sections provide information to help you upgrade an existing driver to version 3.6:

4.1 Supported Upgrade Paths

You can upgrade from any 3.*x* version of the Lotus Notes driver. Upgrading a pre-3.*x* version of the driver directly to version 3.6 is not supported.

4.2 What's New in Version 3.6

Version 3.6 of the driver includes support for Lotus Domino 8.

4.3 Upgrade Procedure

The process for upgrading the Lotus Notes driver is the same as for other Identity Manager drivers. For detailed instructions, see the *Identity Manager 3.6 Installation Guide*.

Customizing the Driver

5

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.

5.1 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.
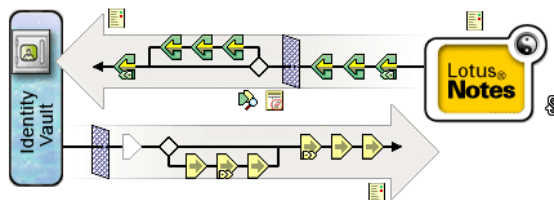
  9c Click the icon for the policy set on the Publisher Channel.

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 .

5.2 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 `windows\setup\drivers\lotusNotes\rules` directory on the Identity Manager DVD or the Identity Manager Windows 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 on Linux and Unix.)

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 5.5, "Custom Driver Parameters," on page 49 for more information about these parameters.

5.3 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 stowith 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 5.6, "Example Files," on page 82.

5.4 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 5.4.1, "Driver Options," on page 36
- Section 5.4.2, "Subscriber Options," on page 39
- Section 5.4.3, "Publisher Options," on page 45

5.4.1 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 5-1 *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* check box 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 by 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>`

Parameter Name and Common Description	Description	XML to Define Driver Parameters

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>`

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>`

Parameter Name and Common Description	Description	XML to Define Driver Parameters

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>`

5.4.2 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 5-2 *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 Allow AdminP Support attribute described in Table 5-4 on page 50.
		`<definition display-name="Allow Domino AdminP Support" name="allow-adminp-support" type="boolean">` `<description> </description>`
		
		`<value>`**true**`</value> </definition>`

Parameter Name and Common Description	Description	XML to Define Driver Parameters

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>`

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>`

Parameter Name and Common Description	Description	XML to Define Driver Parameters

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>`

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, the NotesDriverShim attempts to create `JohnDoe1.nsf`. If this value is 0, 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>`

Parameter Name and Common Description	Description	XML to Define Driver Parameters

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>```

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 the Note at the top of Subscriber Options (page 39), 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>```

Parameter Name and Common Description	Description	XML to Define Driver Parameters

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>`

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 the Note at the top of .

Parameter Name and Common Description	Description	XML to Define Driver Parameters

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 the Note at the top of Subscriber Options (page 39).

notes-v1-schema-format
 	
Use NotesDriver v1 Schema Format
	Specify 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 the Note at the top of Subscriber Options (page 39).

Parameter Name and Common Description	Description	XML to Define Driver Parameters

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>`

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 `update-addressbook` attribute.
		`<definition display-name="Add RegisteUsers to Address Book" name="update-ab-flag" type="boolean">` `<description> </description>`
		
		`<value>`**`true`**`</value> </definition>`

5.4.3 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 correct XML text for such parameters can be found in the file drvparamdefs.xml (../samples/drvparamdefs.xml).

Table 5-3 *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 it 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 it 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>`

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>`

Parameter Name and Common Description	Description	XML to Define Driver Parameters

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>`

loop-detect-flag
 	
Enable Loop Back Detection
	Loopback detection parameter. Set it to True to enable loopback detection. Set it 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 the Note at the top of Publisher Options (page 45), then use the values under the Description heading for this parameter.

Parameter Name and Common Description	Description	XML to Define Driver Parameters

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 the Note at the top of 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>`

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), 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>`

Parameter Name and Common Description	Description	XML to Define Driver Parameters

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 a special driver time stamp on synchronized Notes parameter. Set it 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 it 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>`

5.5 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 5.2, "Automatically Determining Which Certifier to Use," on page 35. 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 5-4 on page 50.

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 5-5 on page 66.

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.

Table 5-4 *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 attempts 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
`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

Parameter Name and Common Description	Description	XML to Define Driver Parameters
`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 it is 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 5.4.2, "Subscriber Options," on page 39, 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 by 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
`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

Parameter Name and Common Description	Description	XML to Define Driver Parameters
`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
`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 `cert-id-password` parameter in the driver configuration. Alternates are `named-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
`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
`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

Parameter Name and Common Description	Description	XML to Define Driver Parameters
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 - '=', '<', '>', '<=', '>=', '!='.
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 The 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

Parameter Name and Common Description	Description	XML to Define Driver Parameters
`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
`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 it is set to True, the user is forced to change his or her password on next login. If it is 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 The 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 date when an old user name expires 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 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 The 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 to 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 5.10, "TELL AdminP Commands," on page 86.	As an attribute to an `add`, `modify`, `move`, `delete` command element. String
`update-addressbook` RegisteUsers 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 The 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

5.5.1 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 5-5 *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 (such as "`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 it is 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 SharedFolder	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 5-6   *ACL Description and Its Java ACL Constant*

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

5.5.2 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 5-7 *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	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	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 not 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 not have any roles defined).	As an attribute to an `add` command element. String
`mailfile-acl-entry-level` MailFile ACL Entry Level	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 5-6 on page 68. 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 it is 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 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 it to True to truncate large documents and remove attachments, and to 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 it is 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 it to True to clear the replication history from the replication object, and set it to False affects nothing.	As an attribute to an `add` command element. Boolean: true | false
`mailfile-rep-cutoff-delete` MailFile 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 (`mailfile-rep-cutoff-interval`). Set it to True to automatically delete documents that are older than the cutoff date, and to 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 it to True to disable replication, and to 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). 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 it to True to remove the current replication entry from the replication object, and set it to False to affect 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 it toTrue to include the ACL, and to 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 it 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 it to True to include documents, and to 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 it to True to include forms, and to 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 it to True to include formulas, and to 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 it to True to not replicate outgoing deletions, and to 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 it to True to not replicate incoming deletions, and to 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 by 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 The efault is *False*.

5.5.3 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 5-8 *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 it to True to truncate large documents and remove attachments and to 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 whether creating the synchronized database replica is performed via AdminP (in the background). When it is 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 it to True to clear the replication history from the replication object, and set it 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 it to True to automatically delete documents that are older than the cutoff date, and to 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 it to True to disable replication, and to 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 it to True to not send local security updates, and to 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 it to True to remove the current replication entry from the replication object, and set it 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 it to True to include the ACL, or to 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 it to True to include agents, or to 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 it to True to include documents, or to 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 it to True to include forms, or to 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 it to True to include formulas, or to 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 it to True to not replicate outgoing deletions, or to False to replicate outgoing 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-ignore-dest-deletes` Rep Ignore Destination Deletes	Specifies whether incoming deletions should not be replicated to the current database. Set it to True to not replicate incoming deletions, or to False to replicate incoming deletions.	As an attribute to an `add`, `modify`, or `delete` command element. Boolean: true | false
`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, 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 might 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. The 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 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 `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

5.6 Example Files

You can configure and create rules, policies, and style sheets by 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 5.1, "Determining eDirectory Object Placement When a Notes Object is Moved," on page 33.

- **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.

- **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 5.2, "Automatically Determining Which Certifier to Use," on page 35 for more information.

- **Override.xsl:** Shows an example of how to use attributes to override parameters. See the list in Table 5-4 on page 50.

- **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 5.1, "Determining eDirectory Object Placement When a Notes Object is Moved," on page 33 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 5.2, "Automatically Determining Which Certifier to Use," on page 35 and Section 5.3, "Using Named Passwords," on page 36.

NOTE: Most of these files are located in the product distribution in windows/setup/drivers/lotusNotes/rules. Some of them are used in the sample driver configuration.

5.7 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.

5.8 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.

5.9 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.

5.9.1 Subscriber Channel

- "Moving a User" on page 84
- "Modifying a User Name in eDirectory (a Rename Event in Notes)" on page 84
- "Renaming a Group" on page 85

Moving a User

1 Make sure you are using Notes 6.0.3 or later and have reviewed Section 5.9.3, "Considerations for Using AdminP," on page 85.

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 5.4.2, "Subscriber Options," on page 39.

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 for Moving a User," on page 116.

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 5.9.3, "Considerations for Using AdminP," on page 85.

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 5.4.2, "Subscriber Options," on page 39.

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 for Renaming: Modifying a User Last Name," on page 115.

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 84.)

1 Make sure you are using Notes 6.0.3 or later and have reviewed Section 5.9.3, "Considerations for Using AdminP," on page 85.

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 5.4.2, "Subscriber Options," on page 39.

5.9.2 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 5.1, "Determining eDirectory Object Placement When a Notes Object is Moved," on page 33.

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.

5.9.3 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 5.10, "TELL AdminP Commands," on page 86, and Domino Console Command in Section 5.5, "Custom Driver Parameters," on page 49.

- ◆ You can enable or disable AdminP support for an individual command. See Allow AdminP Support in Section 5.5, "Custom Driver Parameters," on page 49.

5.10 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 5.5, "Custom Driver Parameters," on page 49.

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 5.4.2, "Subscriber Options," on page 39.

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 113.

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 fails when the AdminP process attempts it.

5.11 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, 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, 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 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.

Managing the Driver

6

As you work with the Lotus Notes driver, there are a variety of management tasks you might need to perform, including the following:

- Starting, stopping, and restarting the driver
- Viewing driver version information
- Using Named Passwords to securely store passwords associated with the driver
- Monitoring the driver's health status
- Backing up the driver
- Inspecting the driver's cache files
- Viewing the driver's statistics
- Using the DirXML® Command Line utility to perform management tasks through scripts
- Securing the driver and its information
- Synchronizing objects
- Migrating and resynchronizing data
- Activating the driver

Because these tasks, as well as several others, are common to all Identity Manager drivers, they are included in one reference, the *Identity Manager 3.6 Common Driver Administration Guide*.

Troubleshooting the Driver

<div style="text-align: right; font-size: 4em;">7</div>

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

- Section 7.1, "Troubleshooting Driver Processes," on page 91
- Section 7.2, "Troubleshooting Lotus Notes-Specific Items," on page 91

7.1 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. For more information, see "Viewing Identity Manager Processes" in the *Identity Manager 3.6 Common Driver Administration Guide*.

7.2 Troubleshooting Lotus Notes-Specific Items

- Section 7.2.1, "Creating Lotus Notes Accounts and Groups," on page 91
- Section 7.2.2, "Troubleshooting Installation Problems," on page 92

7.2.1 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 by 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 is always 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 3.2, "Creating the Driver in Designer," on page 21 or Section 3.3, "Creating the Driver in iManager," on page 25.

7.2.2 Troubleshooting Installation Problems

- "Windows" on page 92
- "AIX, Linux, and Solaris:" on page 92

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.

AIX, Linux, and Solaris:

If you upgrade Domino after installing the driver, you need to do one of the following:

Check and Recreate Links

Check the symbolic links, and re-create them manually if necessary. Use Table 7-1 on page 92 if your Domino server is prior to V7 running eDirectory 8.8. Use Table 7-2 on page 93 if your Domino server is V7.x running eDirectory 8.8.

Table 7-1 *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/Solaris/Linux:
	
	`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 7-2   *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 7-3 *Folder Names for the Different Operating Systems*

Operating System	Folder Name
AIX	ibmpow
Linux	linux
Solaris	sunspa

Reinstall the Driver

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.

Check the Trace Log

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

Driver Properties

A

This section provides information about the Driver Configuration and Global Configuration Values properties for the Lotus Notes driver. These are the only unique properties for drivers. All other driver properties (Named Password, Engine Control Values, Log Level, and so forth) are common to all drivers. Refer to "Driver Properties" in the *Identity Manager 3.6 Common Driver Administration Guide* for information about the common properties.

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

- Section A.1, "Driver Configuration," on page 95
- Section A.2, "Global Configuration Values," on page 100

A.1 Driver Configuration

In iManager:

1 Click 🔵 to display the Identity Manager Administration page.

2 Open the driver set that contains the driver whose properties you want to edit:

    2a In the *Administration* list, click *Identity Manager Overview*.

    2b If the driver set is not listed on the *Driver Sets* tab, use the *Search In* field to search for and display the driver set.

    2c Click the driver set to open the Driver Set Overview page.

3 Locate the driver icon, then click the upper right corner of the driver icon to display the *Actions* menu.

4 Click *Edit Properties* to display the driver's properties page.

By default, the Driver Configuration page is displayed.

In Designer:

1 Open a project in the Modeler.

2 Right-click the driver icon or line, then select click *Properties > Driver Configuration*.

The Driver Configuration options are divided into the following sections:

- Section A.1.1, "Driver Module," on page 95
- Section A.1.2, "Driver Object Password (iManager Only)," on page 96
- Section A.1.3, "Authentication," on page 96
- Section A.1.4, "Startup Option," on page 97
- Section A.1.5, "Driver Parameters," on page 98

A.1.1 Driver Module

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

Table A-1 *Driver Module*

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.
	The java class name is:
	`com.novell.nds.dirxml.driver.notes.NotesDriverShim`
Native	This option is not used with the Lotus Notes driver.
Connect to Remote Loader	Used when the driver is connecting remotely to the connected system. Designer includes two suboptions:
	◆ *Driver Object Password*: Specifies a password for the Driver object. If you are using the Remote Loader, you must enter a password on this page. Otherwise, the remote driver does not run. The Remote Loader uses this password to authenticate itself to the remote driver shim.
	◆ *Remote Loader Client Configuration for Documentation*: Includes information on the Remote Loader client configuration when Designer generates documentation for the driver.

A.1.2 Driver Object Password (iManager Only)

Table A-2 *Driver Object 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.

A.1.3 Authentication

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

Table A-3 *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`

Option	Description
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 hostname 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`
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.

A.1.4 Startup Option

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

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.

A.1.5 Driver Parameters

The Driver Parameters section lets you configure the driver-specific parameters. When you change driver parameters, you tune driver behavior to align with your network environment.

The parameters are presented by category:

Table A-4 *Driver Settings*

Option	Description
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 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 by using the Named Passwords feature. See Section 5.3, "Using Named Passwords," on page 36.
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.
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`.

Table A-5 *Subscriber Settings*

Option	Description
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.

Option	Description
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. The name of the template corresponds to the Domino version: `mail6.ntf`, `mail7.ntf`, or `mail8.ntf`.
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*.

Table A-6   *Publisher Settings*

Parameter	Description
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*.

Parameter	Description
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*.
Publisher heartbeat interval	Allows the driver to send a periodic status message on the Publisher channel when there has been no Publisher channel traffic for the given number of seconds. The default value is 1 second.

A.2 Global Configuration Values

Global configuration values (GCVs) are values that can be used by the driver to control functionality. GCVs are defined on the driver or on the driver set. Driver set GCVs can be used by all drivers in the driver set. Driver GCVs can be used only by the driver on which they are defined.

The Lotus Notes driver includes many GCVs. You can also add your own if you discover you need additional ones as you implement policies in the driver.

To access the driver's GCVs in iManager:

1 Click to display the Identity Manager Administration page.

2 Open the driver set that contains the driver whose properties you want to edit.

   2a In the *Administration* list, click *Identity Manager Overview*.

2b If the driver set is not listed on the *Driver Sets* tab, use the *Search In* field to search for and display the driver set.

2c Click the driver set to open the Driver Set Overview page.

3 Locate the driver icon, click the upper right corner of the driver icon to display the *Actions* menu, then click *Edit Properties*.

or

To add a GCV to the driver set, click *Driver Set*, then click *Edit Driver Set properties*.

To access the driver's GCVs in Designer:

1 Open a project in the Modeler.

2 Right-click the driver icon ![icon] or line, then select *Properties > Global Configuration Values*.

or

To add a GCV to the driver set, right-click the driver set icon ![icon], then click *Properties > GCVs*.

The global configuration values are organized as follows:

Table A-7 *Lotus Notes Certifier Names and Parameters References*

Option	Description
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)`.
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.

Table A-8 *Lotus Notes ADD User Policy Settings*

Option	Description
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.
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.

Option	Description
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.

Table A-9 *Lotus Notes E-Mail Information*

Option	Description
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.
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.

Option	Description
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*.

Table A-10 *Lotus Notes Object Deletion Policy Settings*

Option	Description
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*.
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*.

Table A-11 *Domino Administration Process Activation Command Settings*

Option	Description
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*.

Table A-12 *User Password Policy Settings*

Option	Description
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*.

Table A-13 *Credential Provisioning*

Option	Description
Enable Credential Provisioning Policies	Select *True* to enable the driver's policies for provisioning credentials to Novell® SecureLogin.
On user creation	Select *True* to provision new users with credentials
On user enable/disable	Select *True* to provision credentials to user accounts that have just been enabled and to deprovision credentials from user accounts that have been disabled.

Option	Description
On password changes	Select *True* to reprovision credentials when Identity Vault passwords change.
Application Credential ID	Specify the ID that SecureLogin uses to identify the login. This login is linked with the application in the SecureLogin client.
Application User ID Attribute	Specify the name of the attribute from which to retrieve the application user I.
Provision to Novell SecretStore	Select *True* if Novell SecretStore® is to be used by the credential provisioning policies.
SecretStore Shared Secret Type	Select the shared secret type that Novell SecretStore is using.
Use Enhanced Protection Password	Select *True* if the Novell SecretStore Enhanced Protection Password is to be used. If true is selected then the named password 'secretstore-enhanced-proctection-password' must be appropriately set.
Provision to Novell SecureLogin Repository	Select *True* if the Novell SecureLogin repository is to be used by the credential provisioning policies.
Set Novell SecureLogin Passphrase	Select *True* to enable the SecureLogin passphrase to be set.
SecureLogin Passphrase Question	If you enabled the passphrase to be set, specify the passphrase question. The question needs to be one that can be verified against an Identity Vault attribute.
SecureLogin Passphrase Answer Value Attribute	If you enabled the passphrase to be set, specify the Identity Vault attribute used to verify the user's response to the passphrase question.

Trace Levels

The driver supports the following trace levels:

Table B-1 *Supported Trace Levels*

Level	Description
0	Status messages (success/failure/warning)
1	Informational messages about what Identity Manager is doing
2	Adds dumps of the XML that is passed to/from the driver
3	Adds XML dumps after a policy is applied and more verbose output during policy evaluation
4	Informational messages about the application
5	Debugging messages and messages about application progress and password synchronization information

For information about setting driver trace levels, see to "Viewing Identity Manager Processes" in the *Identity Manager 3.6 Common Driver Administration Guide*.

Ndsrep

<div style="text-align: right; font-size: 3em;">C</div>

The ndsrep program is used to publish information from the Notes database to the Identity Vault (Publisher channel). It is not required if you are only sending information from the Identity Vault to the Notes database (Subscriber channel).

The following sections provide information about using ndsrep.

C.1 Setting Up Ndsrep

1 Review the information about ndsrep in "Publisher Channel" on page 12.

2 Make sure you have copied the necessary files for your platform, as described in Section 2.3, "Copying Ndsrep and Notes.jar to the Required Locations," on page 17.

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.

C.2 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.

C.2.1 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
    ```

C.2.2 Automated Method 2

1. Create a Program document to run ndsrep. Refer to your Domino Administrator documentation for information about how to create a Program document.

    The Program document causes ndsrep to start each time the Domino server starts.

2. When creating the Program document, use the following information:

    ◆ Program name: `ndsrep`.
    ◆ Command: `NotesDrv1`

C.2.3 Manual Method

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

    ```
    load ndsrep instance
    ```

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

C.3 Controlling Ndsrep

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

Table C-1 *ndsrep TELL commands*

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

C.4 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, 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 by 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 Section C.2.2, "Automated Method 2," on page 110).

- 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 5, "Customizing the Driver," on page 33

Sample XDS Documents for Custom Parameters

D

The driver supports using AdminP processes 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 5.4.2, "Subscriber Options," on page 39), and make changes to your driver policy.

The driver also supports sending commands to the Domino server console.

This section provides samples 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 event samples show how the event must be transformed and show attributes that need to be provided by policies.

Refer to Section 5.9, "Move/Rename," on page 83 and Section 5.10, "TELL AdminP Commands," on page 86 for more information.

D.1 Sample for Adding a User

This section shows a sample of the events when creating user John Doe in the Identity Vault.

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 for 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 for more information.

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 for 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 for more information.

-
-

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 for Deleting a User

This section shows a sample of the events when deleting John Doerr from eDirectory.

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 for Sending a Command to the Domino Server Console

This section shows a sample of using the driver's ability to send a command to the Domino server console and receive a response.

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

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 for a 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

A 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`.

D.7.1 ADD Event Policy Rule To Submit ACLEntry Parameters

To view sample code for the 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

Below is a sample showing the 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 that 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 by using a corresponding XML tag.

Table D-1 *Driver Attributes Tags and 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. It is possible to set a field as *SEAL*, but some other code needs to call the appropriate methods to encrypt the field in the document, or the field is not actually sealed (encrypted).

D.8.1 Sample Creation Policy Rules

To view a sample 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 Sample Modify Policy Rule

To view a sample 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 Sample Add XDS Doc

Below is a sample 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 Sample Modify XDS Doc

Below is a sample 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>
```