Novell DirXML® Driver for Delimited Text

1.1.2

IMPLEMENTATION GUIDE

March 31, 2004

www.novell.com

Novell®

Legal Notices

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

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

You may not export or re-export this product in violation of any applicable laws or regulations including, without limitation, U.S. export regulations or the laws of the country in which you reside.

Patents Pending.

Novell, Inc.
1800 South Novell Place
Provo, UT 84606
U.S.A.

www.novell.com

DirXML Driver for Delimited Text Implementation Guide

March 31, 2004

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

Novell Trademarks

Third-Party Trademarks

Contents

About This Guide

This guide explains how to install and configure the DirXML® Driver for Delimited Text.

The guide contains the following sections:

Additional Documentation

For information on Nsure™ Identity Manager, see the Identity Manager Documentation Web site (http://www.novell.com/documentation/lg/dirxml20/index.html).

For information on other DirXML drivers, see Driver Implementation Guides (http://www.novell.com/documentation/lg/dirxmldrivers/index.html).

Documentation Updates

For the most recent version of this document, see *DirXML Driver for Delimited Text* in the Driver Implementation Guides (http://www.novell.com/documentation/lg/dirxmldrivers/index.html) section.

Documentation Conventions

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

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

User Comments

We want to hear your comments and suggestions about this manual and the other documentation included with Novell DirXML. Send e-mail to proddoc@novell.com.

1 Introducing the DirXML Driver for Delimited Text

This section covers the following topics:

- "Overview" on page 9
- "Four Java Interfaces to the Driver" on page 9
- "Driver Heartbeat" on page 10
- "Default Driver Configuration" on page 10

For information on new features in Nsure™ Identity Manager, see "What's New in Identity Manager 2?" in the Novell Nsure Identity Manager 2 Administration Guide.

Overview

The DirXML® Driver for Delimited Text is designed to synchronize user data between Novell® eDirectory™ and XML files or delimited text files that contain comma-separated values.

This driver differs from other DirXML drivers in that it does not interface directly with any particular application. Text files take the place of the application.

Using your own style sheets, you can enable the driver to work with virtually any text-based file that contains predictably repeatable patterns.

Four Java Interfaces to the Driver

The Driver for Delimited Text includes four Java* interfaces:

- InputSorter
- InputSource
- PreProcessor
- PostProcessor.

These interfaces enable you to add extensions, which are optional. The driver continues to function as before without extensions. However, if you want to directly modify the behavior of the driver, but have been unable to make these modifications from a style sheet or DirXML script, extending the Driver for Delimited Text can be useful.

By using Java classes that you write, you can use the interfaces to customize the publish and subscribe processes in the following ways:

Process	Interface	Description
Publish	InputSorter	Defines the processing order of multiple input file. The system where your driver is installed determines the default processing order. For example, files on an NT system are processed in alphabetical order. You can use the InputSorter to impose the processing order that you require.
Publish	InputSource	Provides data other than the files in the default location for the driver to process. For example, you could check an FTP server for input files and then transfer the files to the local file system for processing.
Publish	PreProcessor	Ties data manipulation required to prepare input files for driver processing directly to the driver. Before this interface was available, preprocessing was independent of the driver. You could create a separate application that would monitor another directory for input files, modify the files in some way, and then copy the files to the input directory of the driver. By creating a class that implements the PreProcessor, you can do this type of preprocessing more directly.
Subscribe	PostProcessor	Ties data manipulation required by the application consuming eDirectory output directly to the driver.

These enhancements to the driver require Java programming. For more information, see "Using Java Interfaces to Customize File Processing" on page 38.

Driver Heartbeat

The DirXML engine accepts heartbeat documents from the Driver for Delimited Text. You can configure the driver to send the documents.

Default Driver Configuration

This section provides information on the following:

- "Data Flow" on page 10
- "Supported File Types" on page 12

For information on Identity Manager fundamentals, see the *Nsure Identity Manager Administration Guide* (http://www.novell.com/documentation/lg/dirxml20/index.html).

Data Flow

Publisher and Subscriber Channels

The Driver for Delimited Text supports Publisher and Subscriber channels:

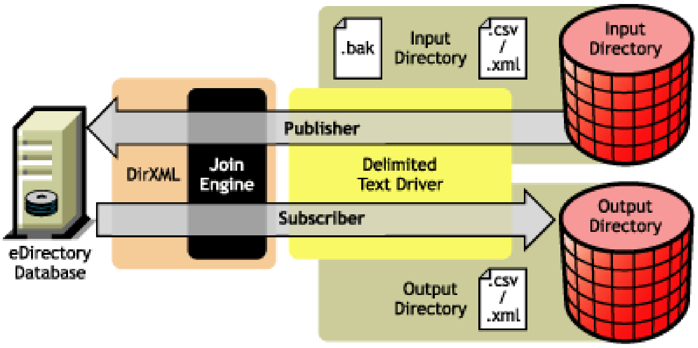
- The Publisher reads information from input text files on your local file system and submits that information to eDirectory via the DirXML engine.

By default, the Publisher does the following:

1. Checks the input directory every 10 seconds.
2. Processes any files that have a .csv extension.
3. Changes .csv extensions of processed files to .bak.
4. Cycles through this process until you stop the driver.

- The Subscriber watches for additions and modifications to eDirectory objects and creates output files on your local file system that reflect those changes.

  By default, the Subscriber keeps an output file open until either 200 transactions have been logged or 30 seconds have elapsed. When either of these thresholds is reached, the output file is saved with a *number*.csv filename and a new output file is opened.



The sample configuration that ships with this driver includes both Subscriber and Publisher channels. However, in many configurations, only one-way data flow is required. In those configurations, only a Publisher or Subscriber channel is used. The other channel is disabled.

Policies

Policies are used to control data synchronization between the driver and eDirectory. The Driver for Delimited Text comes with the set of preconfigured policies detailed in the following table. You can customize these policies by using Novell iManager, as explained in Chapter 4, "Customizing the DirXML Driver for Delimited Text," on page 29.

Policy	Description
Schema Map	Configured on the driver object. Maps eDirectory User properties to application attributes as follows: Surname > LastName Given Name > FirstName Title > Title Internet EMail Address > Email Telephone Number > WorkPhone Facsimile Telephone Number > Fax mobile > WirelessPhone Description > Description The application attributes correspond to the sequence of values in the file or, if present, to the attributes associated with unnamed XDS <field> elements.

Policy	Description
Input Transform	Configured on the driver object. If the input document is an XML document, no transformations are made. If the document is a delimited text file, each record is transformed into an XDS add element for User objects with attributes defined by the schema map. The user CN is formed by concatenating the values of first name and last name. Associations are defined by value the user's e-mail attribute.
Output Transform	Configured on the driver object. Specifies that a comma is used as the delimiter character for output files and that the file format is Comma Separated Values (CSV).
Create	Configured on the Publisher channel. Specifies that in order for a User to be created in eDirectory, the Given Name and Internet EMail Address attributes must be defined.
Matching	Configured on the Publisher channel. Specifies that a user in eDirectory is the same user specified in the input file when the value of Internet Email Address is the same in both places. In the case of a match, only changed attributes are updated in eDirectory.
Placement	Configured on the Publisher channel. Specifies that a new user is placed in the Users\Active container and named with the CN created by the Input Transform rule. You need to create a Users\Active container at the root of your tree before you start the driver.
Event Transform	Configured on the Subscriber channel. If eDirectory reports a Modify or Sync event, those events are changed to an instance element that can be used to create a complete output record.

Supported File Types

The sample configuration currently supports two types of files:

- Comma-Separated Values Files
- XML Files in XDS Format

Comma-Separated Values Files

Comma-separated values (CSV) files are text files that contain data divided into fields and records. Fields are delimited by commas, and records are delimited by a hard return.

If you need a comma or hard return within the value of a particular field, the entire field value should be enclosed in quotes.

Because the meaning of each field in a CSV file is derived from its position, each record in a CSV file should have the same number of fields. Field values can be left blank, but each record should have the same number of delimiter characters.

XML Files in XDS Format

XDS format is the defined Novell subset of possible XML formats. This is the initial format for data coming from eDirectory. By modifying default rules and changing the style sheets, the Delimited Text driver could be configured to work with any XML format.

For detailed information on XDS format, refer to NDS DTD Commands and Events (http://developer.novell.com/ndk/doc/dirxml/index.html?dirxmlbk/data/a5323rs.html).

For information on configuring the driver to use XML files in the XDS format, refer to "Configuring for XDS XML Files" on page 36.

2 Upgrading the Delimited Text Driver

You can upgrade the DirXML® Driver for Delimited Text at the same time that you install the engine or you can upgrade after the engine is installed.

To upgrade your DirXML Driver for Delimited Text after the engine is installed:

1. Run the Identity Manager 2 installation program from the Identity Manager 2.0 CD or download image.

   For Windows* NT*, refer to "Installing to Windows" on page 17.

   For NetWare®, refer to "Installing to NetWare" on page 19.

   For Solaris* or Linux*, refer to "Installing to Linux or Solaris" on page 20.

2. Select the DirXML Driver for Delimited Text.

   The new driver replaces the previous driver. Because your previous driver configuration is preserved, no post-installation configuration is required.

3. Restart the driver.

IMPORTANT: The sample configuration file for the updated driver has changed for this release of Identity Manager. If your current configuration meets your requirements, you don't need to import this sample configuration. If you *do* import the new sample configuration, you will see an additional driver for Delimited Text with a new name, a new eDirectory container specified in the placement rule, and new rule names.

3 Installing the DirXML Driver for Delimited Text

This section provides information on the following:

- "Where to Install the Driver" on page 17
- "Prerequisites" on page 17
- "Installing the Driver" on page 17
- "Setting Up the Driver" on page 23

Where to Install the Driver

A DirXML® Driver for Delimited Text can be installed on the same computer where Novell® eDirectory™ and the DirXML engine are installed. This installation is referred to as a local configuration. If platform or policy constraints make a local configuration difficult or impossible, you can install the driver on the computer hosting the target application. This installation is referred to as a remote configuration.

Unless you extend this driver's functionality with the new Java interfaces, it is capable of only reading input files from the local file system of the computer where the driver is running.

Prerequisites

❑ Novell Nsure™ Identity Manager 2

Installing the Driver

You can install the Driver for Delimited Text (along with other DirXML drivers) at the same time that the DirXML engine is installed. See "Installation" in the Novell Nsure Identity Manager 2 Administration Guide.

As the following sections explain, you can also install the driver separately, after the DirXML engine is installed..

Installing to Windows

1 Run the installation program from the Identity Manager 2.0 CD or the download image.

  If the installation program doesn't autolaunch, you can run \nt\install.exe.

2 In the Welcome dialog box, click Next, then accept the license agreement.

3 In the first DirXML Overview dialog box, review information, then click Next.

  The dialog box provides information on the following:

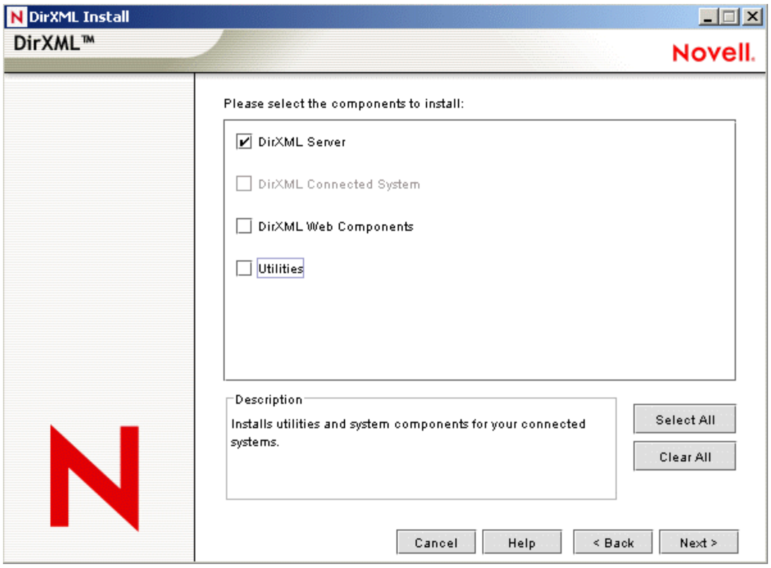
- A DirXML server
- A DirXML connected server system

4 In the second DirXML Overview dialog box, review information, then click Next.

The dialog box provides information on the following:

- A Web-based administration server
- DirXML utilities

5 In the Please Select the Components to Install dialog box, select only DirXML Server, then click Next.



Please select the components to install:

☑ DirXML Server

□ DirXML Connected System

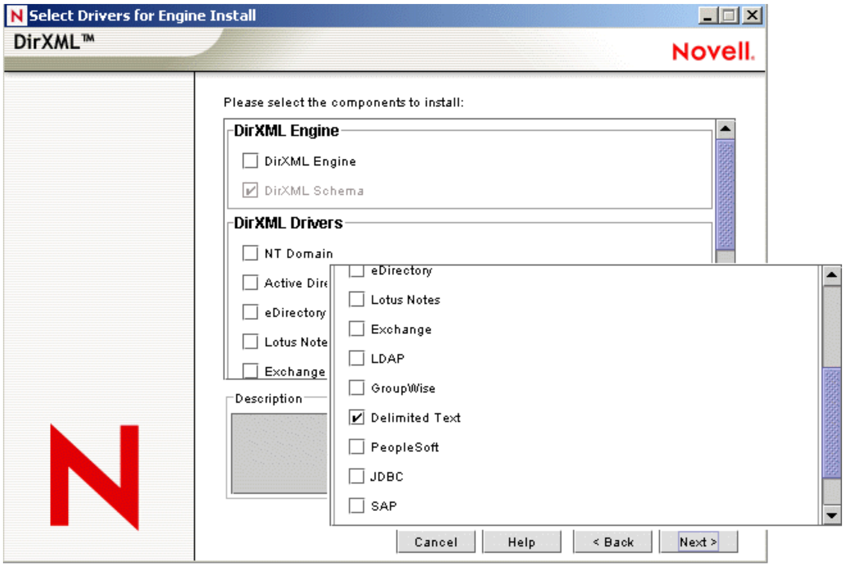
□ DirXML Web Components

□ Utilities

Description

Installs utilities and system components for your connected systems.

6 In the Select Drivers for Engine Install dialog box, select only Delimited Text, then click Next.



Please select the components to install:

DirXML Engine

□ DirXML Engine

☑ DirXML Schema

DirXML Drivers

□ NT Domain

□ Active Dire

□ eDirectory

□ Lotus Note

□ Exchange

□ eDirectory

□ Lotus Notes

□ Exchange

□ LDAP

□ GroupWise

☑ Delimited Text

□ PeopleSoft

□ JDBC

□ SAP

You can't deselect DirXML Schema, which is dimmed. Later, the installation program will extend the schema to enable the newly installed driver to function.

7 In the DirXML Upgrade Warning dialog box, click OK.

8 In the Schema Extension dialog box, type a username and password, then click Next.

9 In the Summary dialog box, review the selected options, then click Finish.

10 In the Installation Complete dialog box, click Close.

After installation you must configure the driver as explained in "Setting Up the Driver" on page 23.

Installing to NetWare

1 At the NetWare® server, insert the Identity Manager 2.0 CD and mount the CD as a volume.

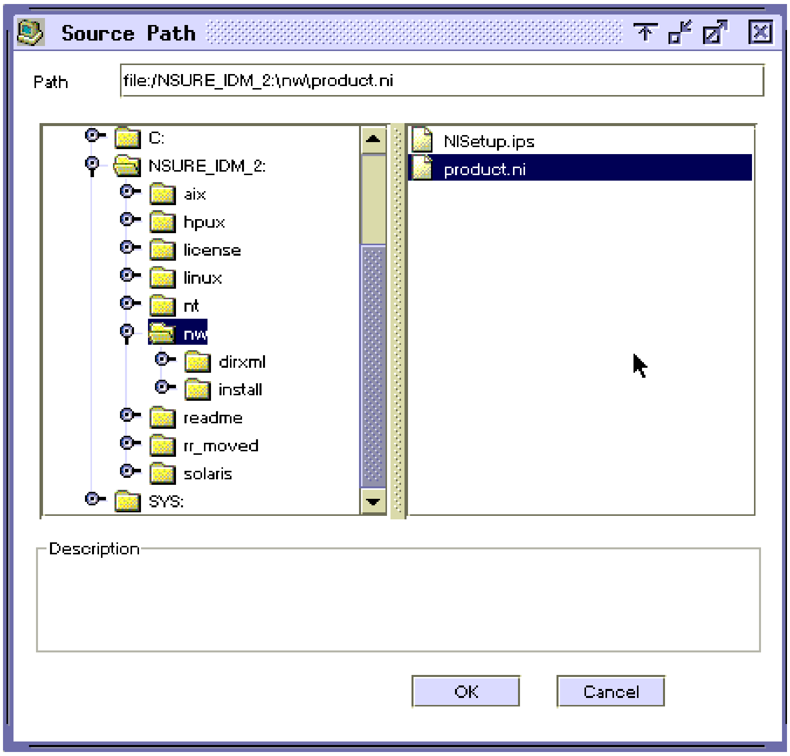
To mount the CD, enter `m cdrom`.

2 (Conditional) If the graphical utility isn't loaded, load it by entering **`startx`**.

3 In the graphical utility, click the Novell icon, then click Install.

4 In the Installed Products dialog box, click Add.

5 In the Source Path dialog box, browse to and select the product.ni file.



Path file:/NSURE_IDM_2:\nw\product.ni

5a Browse to and expand the CD volume (NSURE_IDM_2) that you mounted earlier.

5b Expand the nw directory, select product.ni, then click OK twice.

6 In the Welcome dialog box, click Next, then accept the license agreement.

7 In the DirXML Install dialog box, select only DirXML Server, then click Next.

Deselect the following:

- DirXML Web Components
- Utilities

8 In the Select Drivers for Engine Install dialog box, select only Delimited Text.

Deselect the following:

- DirXML Server
- All drivers except Delimited Text

9 In the DirXML Upgrade Warning dialog box, click OK.

The dialog box advises you to activate a license for the driver within 90 days.

10 In the Schema Extension dialog box, type a username and password, then click Next.

11 In the Summary page, review the selected options, then click Finish.

12 Click Close.

After installation you must configure the driver as explained in "Setting Up the Driver" on page 23.

Installing to Linux or Solaris

By default, the DirXML Driver for Delimited Text is installed when you install the DirXML engine. In case the driver wasn't installed at that time, this section can help you install it.

As you move through the installation program, you can return to a previous section (screen) by entering `previous`.

1 In a terminal session, log in as root.

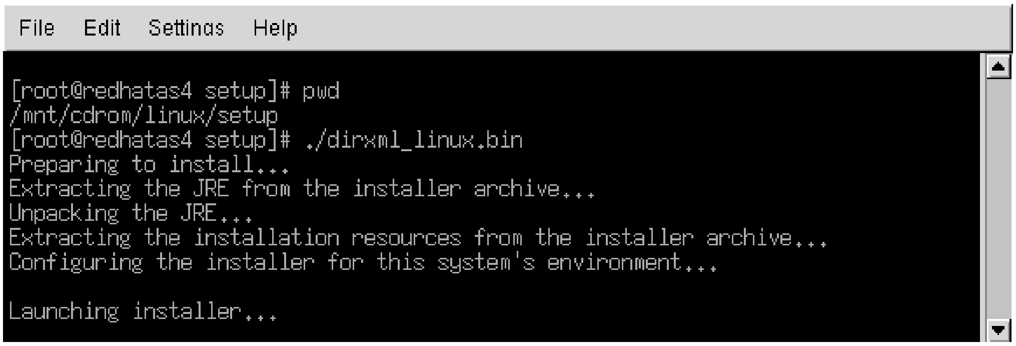
2 Insert the Identity Manager 2.0 CD and mount it.

Typically, the CD is automatically mounted. To manually mount the CD:

Platform	What to Type
Red Hat*	**`mount`** `/mnt/cdrom`, then press Enter
SUSE®	**`mount`** `/media/cdrom`, then press Enter
Solaris*	**`mount`** `/cdrom`, then press Enter

3 Change to the setup directory.

Platform	Path
Red Hat	/mnt/cdrom/linux/setup/
SUSE	/media/cdrom/linux/setup/
Solaris	/cdrom/solaris/nsure_idm_2/setup/



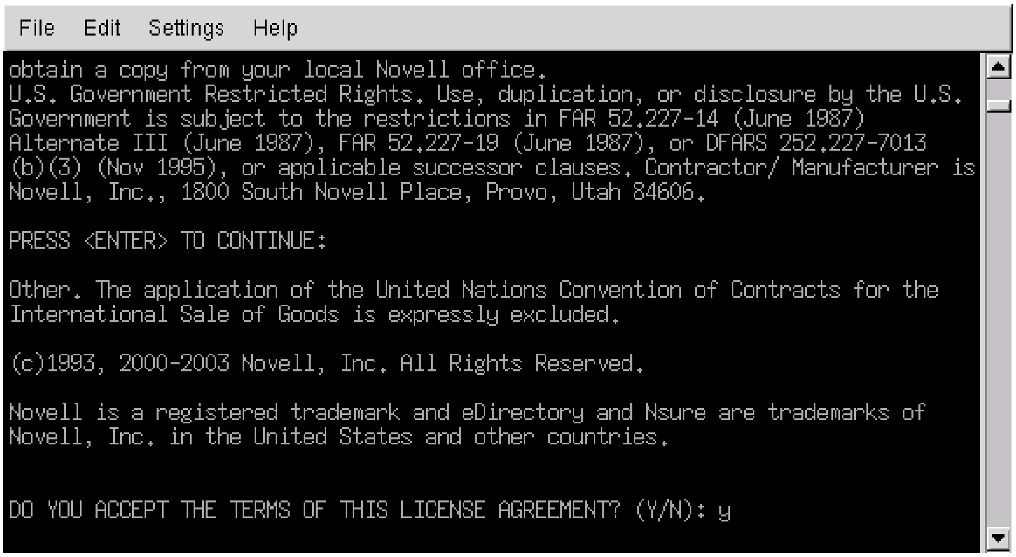
```
[root@redhatas4 setup]# pwd
/mnt/cdrom/linux/setup
[root@redhatas4 setup]# ./dirxml_linux.bin
Preparing to install...
Extracting the JRE from the installer archive...
Unpacking the JRE...
Extracting the installation resources from the installer archive...
Configuring the installer for this system's environment...

Launching installer...
```

4 Run the installation program by typing `./dirxml_linux.bin`.

5 In the Introduction section, press Enter.

6 Accept the license agreement.

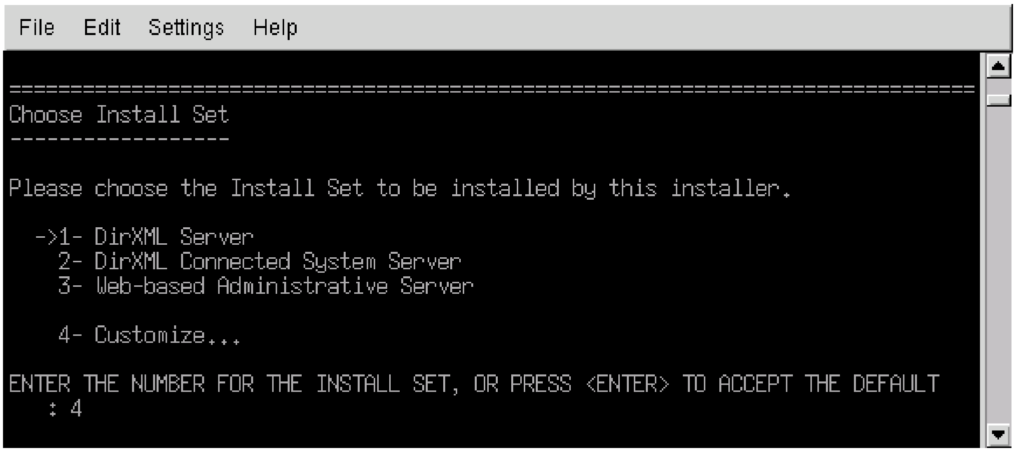
Press Enter until you reach the Do You Accept the Terms of This License Agreement prompt, type `y`, then press Enter.



```
obtain a copy from your local Novell office.
U.S. Government Restricted Rights. Use, duplication, or disclosure by the U.S.
Government is subject to the restrictions in FAR 52.227-14 (June 1987)
Alternate III (June 1987), FAR 52.227-19 (June 1987), or DFARS 252.227-7013
(b)(3) (Nov 1995), or applicable successor clauses. Contractor/ Manufacturer is
Novell, Inc., 1800 South Novell Place, Provo, Utah 84606.

PRESS <ENTER> TO CONTINUE:

Other. The application of the United Nations Convention of Contracts for the
International Sale of Goods is expressly excluded.

(c)1993, 2000-2003 Novell, Inc. All Rights Reserved.

Novell is a registered trademark and eDirectory and Nsure are trademarks of
Novell, Inc. in the United States and other countries.


DO YOU ACCEPT THE TERMS OF THIS LICENSE AGREEMENT? (Y/N): y
```

7 In the Choose Install Set section, select the Customize option.

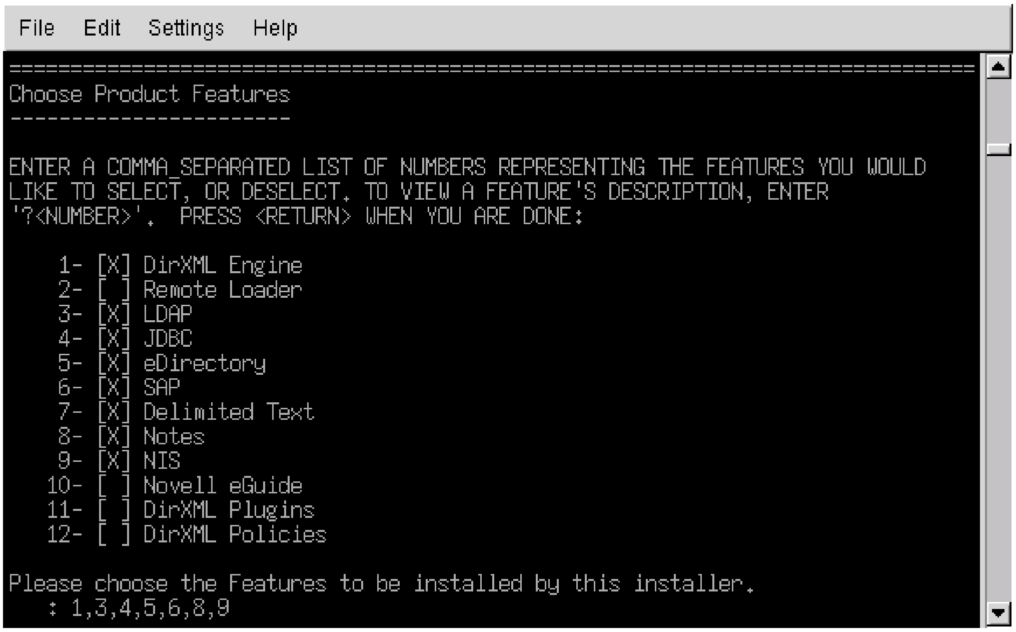
Type 4, then press Enter.



```
===============================================================================
Choose Install Set
------------------

Please choose the Install Set to be installed by this installer.

  ->1- DirXML Server
    2- DirXML Connected System Server
    3- Web-based Administrative Server

    4- Customize...

ENTER THE NUMBER FOR THE INSTALL SET, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
   : 4
```

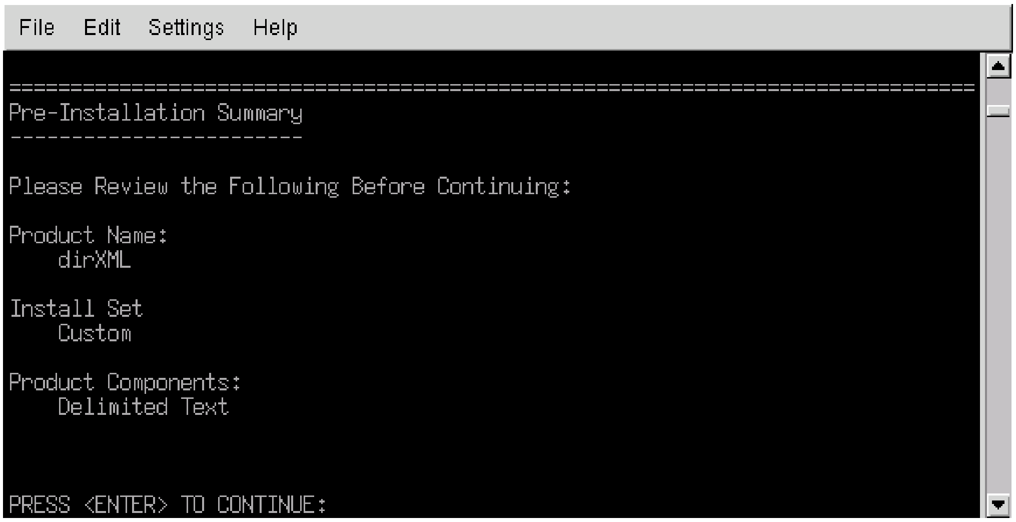
8 At the Choose Product Features section, deselect all features except Delimited Text, then press Enter.

To deselect a feature, type its number. Type a comma between additional features that you deselect.



```
===============================================================================
Choose Product Features
-----------------------

ENTER A COMMA_SEPARATED LIST OF NUMBERS REPRESENTING THE FEATURES YOU WOULD
LIKE TO SELECT, OR DESELECT. TO VIEW A FEATURE'S DESCRIPTION, ENTER
'?<NUMBER>'.  PRESS <RETURN> WHEN YOU ARE DONE:

    1- [X] DirXML Engine
    2- [ ] Remote Loader
    3- [X] LDAP
    4- [X] JDBC
    5- [X] eDirectory
    6- [X] SAP
    7- [X] Delimited Text
    8- [X] Notes
    9- [X] NIS
   10- [ ] Novell eGuide
   11- [ ] DirXML Plugins
   12- [ ] DirXML Policies

Please choose the Features to be installed by this installer.
   : 1,3,4,5,6,8,9
```

9 In the Pre-Installation Summary section, review options.



```
===============================================================================
Pre-Installation Summary
------------------------

Please Review the Following Before Continuing:

Product Name:
    dirXML

Install Set
    Custom

Product Components:
    Delimited Text



PRESS <ENTER> TO CONTINUE:
```

To return to a previous section, type `previous`, then press Enter.

To continue, press Enter.

10 After the installation is complete, exit the installation by pressing Enter.

After installation you must configure the driver as explained in "Setting Up the Driver" on page 23.

Setting Up the Driver

If you are upgrading an existing driver, setup is not required.

If this is the first time the Delimited Text driver has been used, complete the post-installation tasks in the following sections:

- "Configuring the Driver" on page 23
- "Preparing Data Locations" on page 25
- "Starting the Driver" on page 25
- "Migrating and Resynchronizing Data" on page 25
- "Activating the Driver" on page 27

Configuring the Driver

In iManager, complete the following steps:

1 Select DirXML Utilities > Create Driver.

2 Select a driver set.

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

3 Mark Import a Preconfigured Driver from the Server and select the DelimitedTextCSVSample.xml file.

The driver configuration file is installed on the Web server when you set up iManager.

4 Click Next.

You are prompted for the following information:

Field	Description
Driver Name	The eDirectory object name to be assigned to this driver.
Container	The container in eDirectory where new users should be created. If this container doesn't exist, you must create it before you start the driver.
Output File Path	Specify the platform-specific path to the local directory where output files should be created. (For example, on Windows the path is c:\csvsample\output.)
Output File Extension	Specify the extension to append to output files when those files are created.
Input File Path	Specify the platform-specific path to the local directory where input files exist.
Input File Extension	Specify the extension used to designate input files.
Rename File Extension	Specify the extension that an input file will be renamed with after the file has been processed. Leave this field blank if you want the file deleted.
New User Container	Specify the DN of the container where new users are to be placed.
Configure Data Flow	You can configure dataflow to one of the following options: ◆ Bidirectional: Delimited Text and eDirectory are both authoritative sources of the data synchronized between them. ◆ DT-to-eDirectory: Delimited Text is the authoritative source. ◆ eDirectory-to-DT: eDirectory is the authoritative source.
Install Driver as Remote/ Local	Configure the driver for use with the Remote Loader service by selecting the Remote option, or select Local to configure the driver for local use. If you select Local, you can skip the remaining parameters.
Remote Host Name and Port	Specify the host name or IP address and port number for where the Remote Loader service has been installed and is running for this driver. The default port is 8090.
Driver Password	The driver object password is used by the Remote Loader to authenticate itself to the DirXML server. It must be the same password that is specified as the driver object password on the DirXML Remote Loader.
Remote Password	The Remote Loader password is used to control access to the Remote Loader instance. It must be the same password that is specified as the Remote Loader password on the DirXML Remote Loader.

5 Click Finish.

NOTE: You can configure and modify the driver's other parameters. For more information, refer to "Configuring Driver Parameters" on page 29.

Preparing Data Locations

If you use all of the defaults provided by the sample configuration, prepare locations for DirXML data:.

1. Add the containers Users\Active at the root level of your eDirectory tree.
2. On Windows, create the following two directories on your local file system:
   - c:\cvsample\input
   - c:\cvsample\output

   On Solaris, Linux, or NetWare, create input and output directories wherever you'd like. Then update the driver configuration with the correct platform-specific paths. For more information, refer to Source File Path in the Publisher Settings section. (See "Publisher Settings" on page 34).

Starting the Driver

If you changed default data locations during configuration, ensure that the new locations exist before you start the driver.

1. In iManager, select DirXML Management > Overview.
2. Locate the driver in its driver set.
3. Click the driver status indicator in the upper right corner of the driver icon, then click Start Driver.

   Synchronization takes place on an object-by-object basis as changes are made to individual objects. If you want to have an immediate synchronization, you must initiate that process as explained in "Migrating and Resynchronizing Data" on page 25.

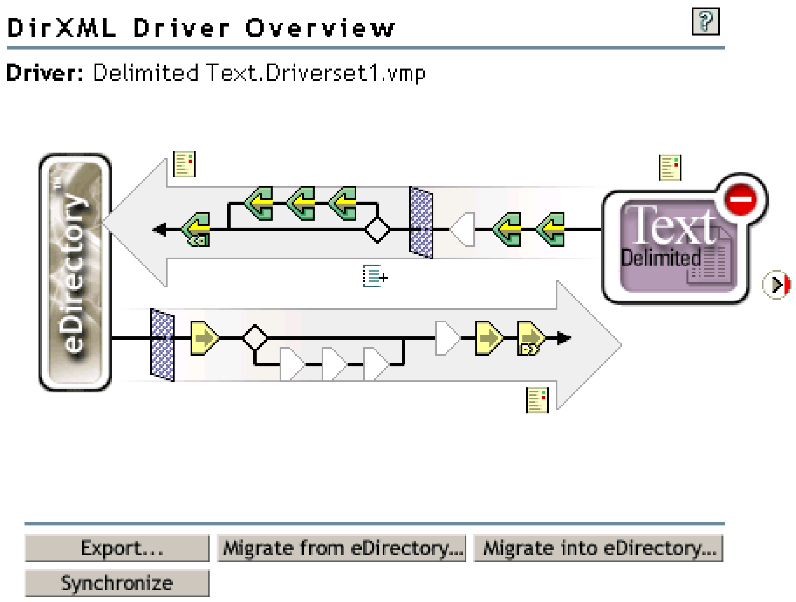
Migrating and Resynchronizing Data

Identity Manager synchronizes data as the data changes. If you want to synchronize all data immediately, you can do one of the following:

- Migrate new data into or from eDirectory.
- Resynchronize existing data by using the driver set Properties page.

The Migrate from eDirectory and Synchronize options usually behave independently of any driver. However, some issues exist with these options and the DirXML Driver for Delimited Text.

The following figure illustrates the buttons that select the Migrate from eDirectory and Synchronize options:



DirXML Driver Overview

Driver: Delimited Text.Driverset1.vmp

Migrating Data from eDirectory

To find eDirectory entries that do not have a DirXML association for the driver, click Migrate from eDirectory. Non-associated entries that are not vetoed by rules processing are written to the output file and will receive an association.

NOTE: Entries that already have up-to-date associations aren't written to the output file. To output all associated entries, click Synchronize.

Migrating Data into eDirectory

With most DirXML drivers, migrating into eDirectory causes the DirXML engine to query the application through its driver. The query finds entries that can be synchronized from the application into eDirectory. These synchronized entries are created in eDirectory. The driver assigns an association value to each new entry.

The Migrate into eDirectory option has no function when used with the DirXML Driver for Delimited Text. This driver gets all of its input as files are placed in the input directory. Because the driver is always up-to-date concerning the files it processes, you don't need to do any additional processing or querying. If you do click Migrate into eDirectory, nothing happens because queries are disabled with this driver.

Synchronizing Data

When entries in eDirectory have an association with the selected driver, the Synchronize option causes those entries to synchronize again with the associated entries in the application. If the driver has a Subscriber channel, eDirectory is considered to be the authoritative source for the attributes in the Subscriber filter.

The DirXML Driver for Delimited Text can't query the application directly to determine what is out of sync. Therefore, the DirXML engine sends a Modify event to the driver for each associated entry. This event causes each of these entries to be written to an output file whenever you click Synchronize. If no Subscriber is configured, Synchronize will have no effect with the driver.

To use Migrating from eDirectory, Migrating into eDirectory, or Synchronize:

1. In iManager, select DirXML Management > Overview.
2. Locate the driver set containing the Delimited Text driver, then double-click the driver icon.
3. Click the appropriate migration button.

Activating the Driver

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

For information on activation, refer to "Activating Novell Identity Manager Products" in the Novell Nsure Identity Manager 2 Administration Guide.

4 Customizing the DirXML Driver for Delimited Text

The DirXML® Driver for Delimited Text includes a sample configuration that you can use as a starting point for your deployment.

Most deployments require you to change the sample configuration. For example, if you need only one-way data synchronization, or if the attributes you're synchronizing are different from the eight provided in the sample, you need to customize the driver.

This section covers the following customization topics:

- "Configuring Driver Parameters" on page 29
- "Configuring Data Synchronization" on page 37
- "Using Java Interfaces to Customize File Processing" on page 38

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.

Configuring Driver Parameters

When you change driver parameters, you tune driver behavior to align with your network environment. For example, you might find the default publisher polling interval to be shorter than your synchronization requires. Making the interval longer could improve network performance while still maintaining appropriate synchronization.

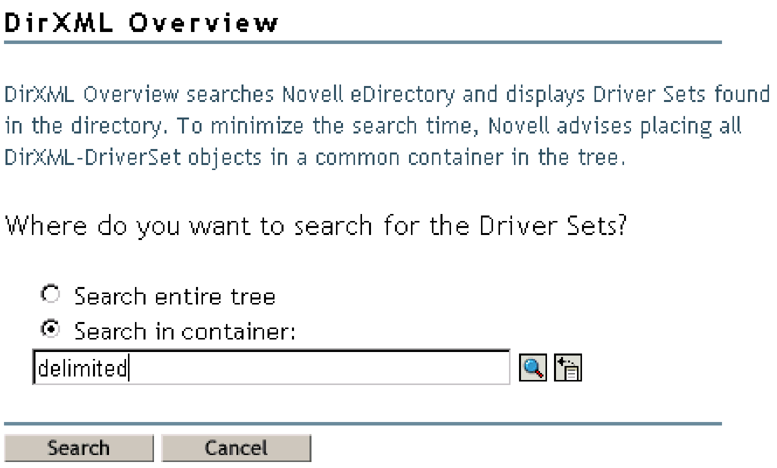
Driver parameters are divided into the following settings:

- Driver (See "Driver Settings" on page 31.)
- Subscriber (See "Subscriber Settings" on page 32.)
- Publisher (See "Publisher Settings" on page 34.)

To configure the driver parameters:

1. In iManager, select DirXML Management > Overview.
2. Using the Find, Browse, or Search feature, locate the DirXML Driver for Delimited Text set.

   The following figure illustrates using the Search feature.



DirXML Overview

DirXML Overview searches Novell eDirectory and displays Driver Sets found in the directory. To minimize the search time, Novell advises placing all DirXML-DriverSet objects in a common container in the tree.

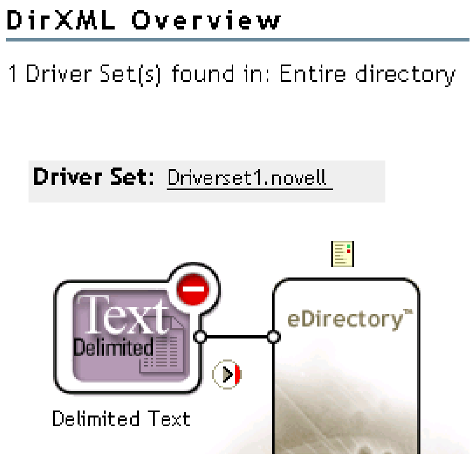
Where do you want to search for the Driver Sets?

○ Search entire tree

● Search in container:

delimited

3 View the driver overview by clicking the driver icon (Delimited Text).

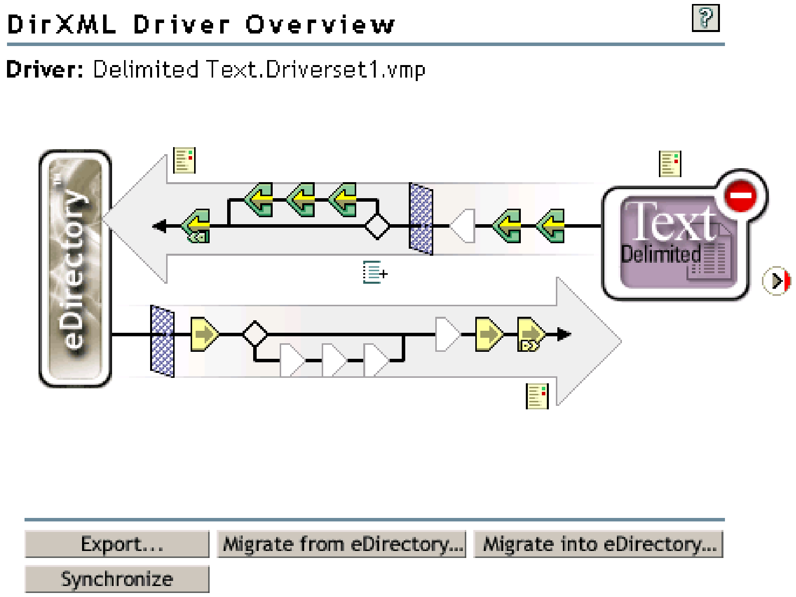


DirXML Overview

1 Driver Set(s) found in: Entire directory

Driver Set: Driverset1.novell

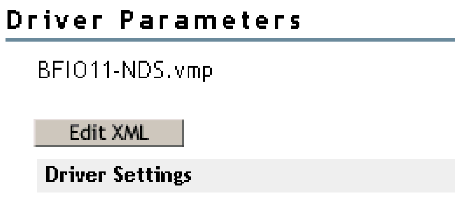
4 Access the Driver Configuration page by clicking the driver icon again.



DirXML Driver Overview

Driver: Delimited Text.Driverset1.vmp

5 Scroll to the Driver Parameters section, make changes, then click OK.



Driver Parameters

BFIO11-NDS.vmp

Edit XML

Driver Settings

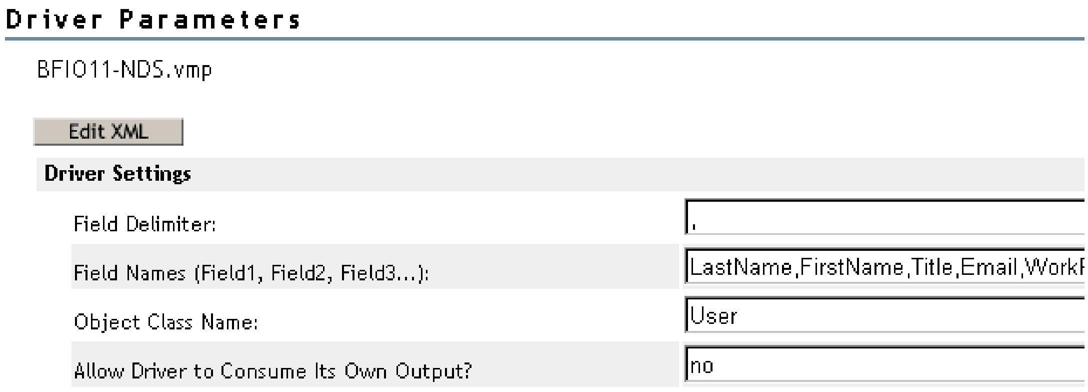
To make changes to the Driver Settings section, see "Driver Settings" on page 31.

To make changes to the Subscriber Settings section, see "Subscriber Settings" on page 32.

To make changes to the Publisher Settings section, see "Publisher Settings" on page 34.

Driver Settings

The following figure illustrates the driver settings and their default values in the sample configuration.



Driver Parameters

BFIO11-NDS.vmp

Edit XML

Driver Settings

Field Delimiter:	,
Field Names (Field1, Field2, Field3...):	LastName,FirstName,Title,Email,WorkF
Object Class Name:	User
Allow Driver to Consume Its Own Output?	no

Field Delimiter

Field Delimiter indicates the character that is used to delimit field values in the input files. It must be one character.

If the values of any of the input fields contain this character, enclose the entire value in quotes to prevent it from being seen as a delimiter.

NOTE: Changing this delimiter parameter to something other than a comma does not automatically change the delimiter character used in the output files when a Subscriber is used. To change the delimiter character in the output files, edit the Output Transform style sheet. The delimiter character is assigned to a variable near the top of that style sheet.

Field Names

Field Names is a comma-separated list of attribute names that can be referred to in the Schema Mapping rule. In the input files, the fields of the records must correspond to the order and positioning of the names in this list.

For example, if you list eight field names in this parameter, each record of the input files should have eight fields separated by the field delimiter character. On NetWare® and Windows, see sample.csv in the delimitedtext/samples directory for an example. On Solaris and Linux, sample.csv is located in the /usr/lib/dirxml/rules/delim directory.

The following table lists the default values:

Parameter	Sample Configuration Value
Field Names (Field 1, Field 2, Field 3...)	LastName,FirstName,Title,Email,WorkPhone,Fax, WirelessPhone,Description

Object Class Name

Object Class Name is the Novell® eDirectory™ class name that should be used when creating new objects to correspond to input files.

Allow Driver to Consume Its Own Output

This parameter prevents you from inadvertently creating a situation in which the driver writes output files that are immediately read in again as input of the same driver.

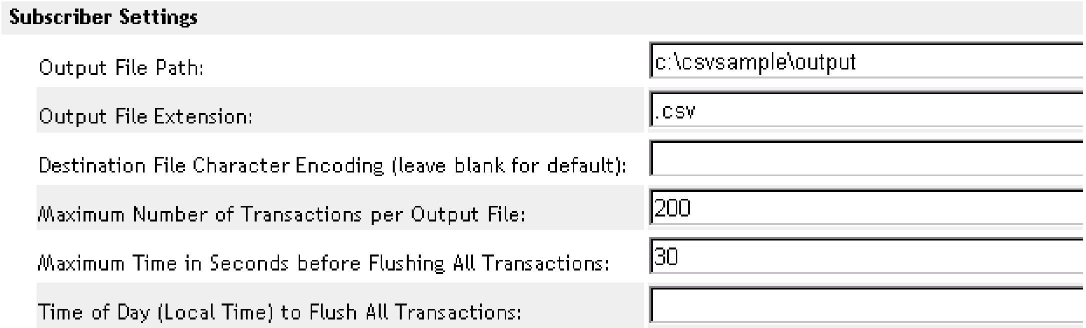
The default is No. By default, the driver won't load if all the following conditions occur:

- You have both a Subscriber channel and a Publisher channel.
- The input and output directories are the same.
- The input and output file extensions are the same.

If you want to feed the output of the Subscriber channel into the input of the Subscriber channel as a way to detect eDirectory events to trigger other changes in eDirectory, set this parameter to Yes. For example, if you want the Full Name attribute updated when the Given Name, Surname, or Initials attributes are updated, set this parameter to Yes.

Subscriber Settings

The following figure illustrates the Subscriber settings and their default values in the sample configuration.



Subscriber Settings

Output File Path:	c:\csvsample\output
Output File Extension:	.csv
Destination File Character Encoding (leave blank for default):	
Maximum Number of Transactions per Output File:	200
Maximum Time in Seconds before Flushing All Transactions:	30
Time of Day (Local Time) to Flush All Transactions:	

Output File Path

Output File Path is the directory on the local file system where output files will be created. An error occurs if this directory doesn't exist.

Platform	Sample Configuration Value
Windows	c:\csvsample\output

Platform	Sample Configuration Value
Solaris or Linux	/csvsample/output
NetWare	Specify the volume (for example, sys:csvsample\output)

Output File Extension

Output files have a unique name that ends with the characters in the Destination File Extension parameter. If the output files from a Subscriber are used as input files for the Publisher of another Delimited Text DirXML Driver, the destination file extension must match the source file extension parameter of the second driver.

Destination File Character Encoding

When the Destination File Character Encoding parameter contains no value, the default Java character encoding for your locale are used.

To use an encoding other than the default for your locale, enter one of the canonical names from the Supported Encodings table (http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html).

NOTE: The Publisher and Subscriber can use different character encodings.

Maximum Number of Transactions Per Output File

This parameter determines the maximum number of transactions that are written to a single output file. When the file transaction limit is reached, the file closes, and a new file is created for subsequent transactions. If you don't want to limit the number of transactions that can be written to a single file, leave this parameter blank or set it to zero. For more information, refer to Maximum Time in Seconds Before Flushing All Transactions.

Maximum Time in Seconds Before Flushing All Transactions

If no new transactions have been written to the output file in the amount of time specified in this parameter, the file is closed. When new transactions need to be written, a new output file is created. If you don't want to limit the time that can pass before the output file is closed, leave this parameter blank or set it to zero.

Time of Day (Local Time) to Flush All Transactions

If a value is supplied for this parameter, the current output file is closed at the specified time each day. Subsequent transactions are written to a new file. This parameter does not prevent the Maximum Number of Transactions or the Maximum Time in Seconds parameters from also acting as output file thresholds. If you use this parameter and only want one file per day, set the other two parameters to zero.

The format of this parameter can be HH:MM:SS (using the 24-hour clock) or H:MM:SS AM/PM. An hour is required, but the minutes and seconds are optional. Because the parameter assumes local time, any time zone information included in the value is ignored.

NOTE: The previous three parameters (Maximum Number of Transactions Per Output File, Maximum Time in Seconds Before Flushing All Transactions, Time of Day to Flush All Transactions) are all capable of acting as a threshold for the transaction size a file is able to grow to, or for the time that it will remain open to accept new transactions.

As long as an output file is still open for writing by the Delimited Text driver, it shouldn't be considered as finalized, and you should avoid opening the file in any other process until the driver closes it. For this reason, one of the three previous parameters must be set to assure that output files don't remain open indefinitely. To avoid this condition, if the driver detects that all three parameters are blank (or zero) it automatically sets the Maximum Number of Transactions per Output File to the value of 1.

Publisher Settings

The following table lists the Publisher settings and their default values in the sample configuration.

Parameter	Sample Configuration Value
Source File Path	On Windows: c:\csvsample\input On Solaris and Linux: /usr/lib/dirxml/rules/delim For NetWare, you need to specify the volume (for example, sys:csvsample\input)
Source File Extension	.csv
Source File Character Encoding (leave blank for default)	[blank]
Source File Rename Extension (leave blank to delete a file)	.bak
Polling Rate (in seconds)	10

Source File Path

The Publisher looks for new input files in the Source File Path, which is a directory on the local file system.

Source File Extension

The Publisher uses only files that have the extension specified in this parameter. After the files have been processed, the value of the Source File Rename Extension parameter is appended to the filename, so the Publisher won't try to process the same file again. If the value of the Source File Rename Extension parameter is left blank, the source file is deleted after it is processed.

Source File Character Encoding

When the Source File Character Encoding parameter contains no value, the default Java character encoding for your locale is used.

To use an encoding other than the default for your locale, enter one of the canonical names from the Supported Encodings table (http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html).

If the Source File Extension parameter is .xml, the Source File Character Encoding can be indicated in one of two ways:

- If a value is indicated in the Source File Character Encoding parameter, it is used.
- If the parameter is blank, and if the XML document specifies an Encoding Declaration as described in the W3C XML Recommendation (http://www.w3.org/TR/REC-xml#charencoding) in paragraph 4.3.3, the Encoding Declaration is handled by the XML parser in the DirXML engine.

  The DirXML XML parser handles the following character encodings:

- UTF-8
- UTF-16
- ISO-8859-1
- US-ASCII

NOTE: The Publisher and Subscriber can use different character encodings.

Source File Rename Extension

For information on Source File Rename Extension, see Source File Extension in this section ("Publisher Settings" on page 34).

IMPORTANT: If you change the default, use only characters that are valid in filenames on your platform. Invalid characters cause the rename to fail and the driver to reprocess the same file repeatedly.

Polling Rate

When the Publisher has finished processing all source files, it waits the number of seconds specified in this parameter before checking for new source files to process.

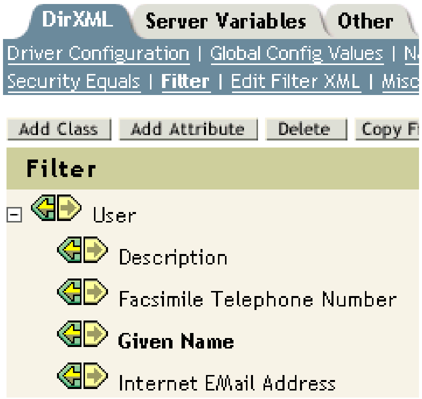
Setting Up One-Way Synchronization

If your data synchronization goes only one way, disable the channel that you won't use. To disable one of the channels, clear the filters on the channel you don't need and don't specify a path for the input or output directory, depending on the channel.

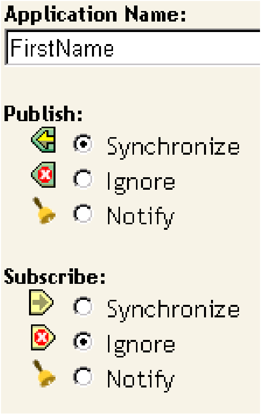
For example, if you only need a Publisher channel:

1 On the Filter editor in iManager, clear the filters on the Subscriber object.

1a For example, select the Given Name filter.


DirXML
Server Variables
Other
Driver Configuration | Global Config Values | N
Security Equals | Filter | Edit Filter XML | Misc
Add Class
Add Attribute
Delete
Copy F
Filter
User
Description
Facsimile Telephone Number
Given Name
Internet EMail Address

1b Select Ignore in the Subscribe section.



Application Name:

FirstName

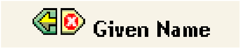
Publish:

● Synchronize
○ Ignore
○ Notify

Subscribe:

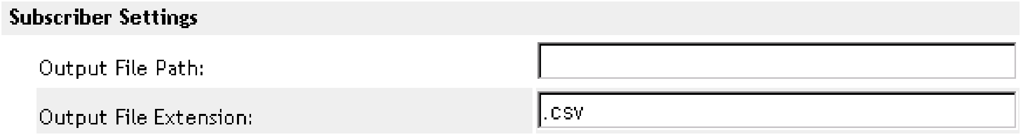
○ Synchronize
● Ignore
○ Notify

As the following figure illustrates, the filter’s Subscribe channel is disabled.



Given Name

2. Save the changes by clicking OK.
3. In the Driver Parameters section, scroll to Subscriber Settings and remove the path specified for the Output File Path.



Subscriber Settings

Output File Path:	
Output File Extension:	.csv

If you only need a Subscriber channel, clear the filters on the Publisher object and remove the path specified for the Source File Path in the Driver Parameters section.

Configuring for XDS XML Files

You can use XML files in XDS format instead of comma-separated value (CSV) files with the driver.

Because you generally will want to use this driver only with a Publisher or Subscriber channel, perform only the steps from the section that you need.

Using the Publisher Channel

To have the driver accept input in XML format, change the input file extension to .xml.

Using the Subscriber Channel

To have the driver send output in XDS format, remove the Event Transform and Output Transform style sheets from the Subscriber channel.

1. In iManager, select eDirectory Administration > Delete Object.
2. Browse to and select the SubscriberEventTransformSS object for the driver.
3. Click OK.
4. Click Repeat Task.

5 Browse to and select the OutputTransformSS object for the driver.

6 Click OK twice.

Configuring Data Synchronization

The real power of Nsure™ Identity Manager is in managing the shared data itself. This section covers some common customizations for the DirXML Driver for Delimited Text.

The sample configuration available with the driver uses comma-separated value files. However, you can use the driver in many ways. It is designed to be as flexible as possible. The driver passes the text-based files largely unchanged to the style sheets. The style sheets do most of the work. You can write new style sheets that will allow the driver to work with almost any text-based file that contains predictably repeatable patterns.

The basis for this exchange is the <delimited-text> XML element. For example, to design a Publisher that reads information from a text file, create an Input Transform style sheet that receives the contents of the file and converts it into a <delimited-text> element.

The following is an example of a <delimited-text> element:

```
<delimited-text>
     <record>
          <field>John</field>
          <field>Maxfield</field>
          <field>555-1212</field>
     </record>
     <record>
          <field>Sarah</field>
          <field>Lopez</field>
          <field>555-3434</field>
     </record>
</delimited-text>
```

When field elements appear like this without an identifying name attribute, the driver uses the field position and matches it with the position of the Field Name driver parameter.

You can provide the field name within the XML:

```
<delimited-text>
     <record>
          <field name="FirstName">John</field>
          <field name="LastName">Maxfield</field>
          <field name="Phone">555-1212</field>
     </record>
     <record>
          <field name="FirstName">Sarah</field>
          <field name="LastName">Lopez</field>
          <field name="Phone">555-3434</field>
     </record>
</delimited-text>
```

For detailed information on writing style sheets to handle other document types, refer to the sample style sheets that come with this driver. If you create the driver using the sample configuration, you will find Input Transform, Output Transform, and Event Transformation style sheets that you can use as a starting point.

Using Java Interfaces to Customize File Processing

Java interfaces enable you to customize file processing by using Java classes that you write. These interfaces are InputSorter, InputSource, PreProcessor, and PostProcessor.

These enhancements to the driver require Java programming. To implement this functionality, complete the following processes:

- Create a Java class that implements one of the new interfaces.
- Create a Java .jar file that contains your new class.
- Configure the driver to use the new class.

Creating a New Java Class

JavaDoc and a sample class are included with the driver to help you implement this new functionality. Find these files at *platform*\dirxml\drivers\delimitedtext\extensions.

Creating a Java .jar File

After you have implemented your class file, create a Java .jar file (Java archive) using the jar tool. The .jar file must contain the class that you have created. Put the .jar file into the novell/nds/lib directory. The path might differ, depending on the platform you're on, but it should be the same location as the DelimitedTextShim.jar and the DelimitedTextUtil.jar.

Configuring the Driver to Use the New Class

After you have placed the new .jar file in the correct location, configure the driver to use your new class by modifying the driver's properties.

1. In iManager, select DirXML Management > Overview.
2. Locate the driver in its driver set.
3. Click the driver icon to open the Driver Overview Page.
4. Click the driver icon again to open the Modify Object page.
5. Click Driver Configuration > Driver Parameters > Edit XML.
6. Locate the <publisher-options> section of the file.

   This file defines which parameters and values appear in the Driver Parameters section of the Driver Configuration page.

   For each class you created that works on the Publisher channel, you will enter an additional option in the <publisher-options> section. After you've updated this file, you'll see your new options in the interface.
7. For each new class you created on the publisher channel, add an entry corresponding to the interface type. Use the following table as a guide:

Interface	New Entry
InputSorter	<input-sorter display-name="InputSorter Class">com.acme.MyNewClass</input-sorter> <input-sorter-params display-name="InputSorter init string">MY CONFIG PARAMS</input-sorter-params>
InputSource	<input-source display-name="InputSource Class">com.acme.MyNewClass</input-source> <input-source-params display-name="InputSource init string">MY CONFIG PARAMS</input-source-params>
PreProcessor	<pre-processor display-name="PreProcessor Class">com.acme.MyNewClass</pre-processor> <pre-processor-params display-name="PreProcessor init string">MY CONFIG PARAMS</pre-processor-params>

7a Replace *com.acme.MyNewClass* with the name of the class that you have defined along with a full package identifier.

7b Replace *MY CONFIG PARAMS* with any information that you want to pass to the init method of your class.

The init method of your class is then responsible for parsing the information contained in this string. If your class doesn't require a configuration string to be passed to the init method, you can leave off the whole element, in which case null would be passed to the init method.

8 If you created a PostProcessor rule, locate the <subscriber-options> section of the file and add the following lines:

```
<post-processor display-name="PostProcessor Class">com.acme.MyNewClass</
post-processor>
```

```
<post-processor-params display-name="PostProcessor init string">MY CONFIG
PARAMS</post-processor-params>
```

8a Replace *com.acme.MyNewClass* with the name of the class that you have defined along with full package information.

8b Replace *MY CONFIG PARAMS* with any information that you want to pass to the init method of your class.

The init method of your class is then responsible for parsing the information contained in this string. If your class doesn't require a configuration string to be passed to the init method, you can leave off the entire element, in which case null would be passed to the init method.

9 Click OK.

A Documentation Updates

This section contains new or updated information on the DirXML® Driver for Delimited Text.

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

If you need to know whether a copy of the PDF documentation you are using is the most recent, check the date that the PDF file was published. The date is in the Legal Notices section, which immediately follows the title page.

New or updated documentation was published on the following dates:

- "March 31, 2004" on page 41

March 31, 2004

The following updates were made in this section:

Location	Change
Chapter 1, "Introducing the DirXML Driver for Delimited Text," on page 9	Deleted general information that applies to and is found in the *Administration Guide* but doesn't apply to the DirXML Driver for Delimited Text.
Chapter 3, "Installing the DirXML Driver for Delimited Text," on page 17	Provided installation steps by platform (Windows, Linux or Solaris, and NetWare®).
Chapters 2 and 3	Added graphics concerning the interface and configuration settings.