

# Novell Identity Manager Driver for PeopleSoft\*

3.7

[www.novell.com](http://www.novell.com)

IMPLEMENTATION GUIDE

March 19, 2007



Novell®

## Legal Notices

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

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

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

Copyright © 2000-2007 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

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

Novell, Inc.  
404 Wyman Street, Suite 500  
Waltham, MA 02451  
U.S.A.  
[www.novell.com](http://www.novell.com)

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

## **Novell Trademarks**

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

## **Third-Party Materials**

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



# Contents

<b>About This Guide</b>	<b>9</b>
<b>1 Introducing the Identity Manager Driver for PeopleSoft</b>	<b>11</b>
1.1 Changes in Terminology	11
1.2 Benefits	12
1.3 Driver Features	12
1.3.1 Identity Manager 3.5 New Features	13
1.4 Key Driver Features	13
1.4.1 Local Platforms	14
1.4.2 Remote Platforms	14
1.4.3 Role-Based Entitlements	14
1.5 Driver Components	14
1.5.1 Driver Configuration	15
1.5.2 Driver Shim	15
1.5.3 Event Server	15
1.5.4 PeopleSoft Service Agent	15
1.6 Publishing to the Identity Vault	16
1.6.1 Event Descriptions	16
1.7 Subscribing from the Identity Vault	18
<b>2 Installing the Driver</b>	<b>21</b>
2.1 Driver Requirements	21
2.2 Installation Instructions	21
2.3 Importing the PeopleSoft Driver	21
2.3.1 Importing the Driver Configuration File in Designer	21
2.3.2 Importing the Driver Configuration in iManager	22
<b>3 Upgrading the Driver</b>	<b>25</b>
3.1 Upgrading the Driver in Designer	25
3.2 Upgrading the Driver in iManager	27
<b>4 Configuring Your PeopleSoft System</b>	<b>29</b>
4.1 Installing the PeopleSoft Service Agent	29
4.1.1 Installing the PSA on PeopleSoft 7.5	29
4.1.2 Installing the PSA on PeopleSoft 8.1	33
4.2 Running the Message Agent Test Program	37
4.2.1 Testing the DIRXML_TRANS01 PeopleSoft Message Agent	37
4.2.2 Testing the DIRXML_SCHEMA01 PeopleSoft Message Agent	38
4.2.3 Testing the DIRXML_SCHEMA01_UPDATE PeopleSoft Message Agent	39
4.2.4 Testing the DIRXML_SCHEMA01_QUERY PeopleSoft Message Agent	40
4.3 Installing the Event Server	40
4.3.1 Installing the Event Server	40

<b>5</b>	<b>Activating the Driver</b>	<b>45</b>
<b>6</b>	<b>Managing the Driver</b>	<b>47</b>
6.1	Starting, Stopping, or Restarting the Driver	47
6.1.1	Starting the Driver in Designer	47
6.1.2	Starting the Driver in iManager	47
6.1.3	Stopping the Driver in Designer	47
6.1.4	Stopping the Driver in iManager	47
6.1.5	Restarting the Driver in Designer	48
6.1.6	Restarting the Driver in iManager	48
6.2	Migrating and Resynchronizing Data	48
6.3	Using the DirXML Command Line Utility	48
6.4	Viewing Driver Versioning Information	49
6.4.1	Viewing a Hierarchical Display of Versioning Information	49
6.4.2	Viewing the Versioning Information As a Text File	50
6.4.3	Saving Versioning Information	52
6.5	Reassociating a Driver Set Object with a Server Object	53
6.6	Changing the Driver Configuration	53
6.7	Storing Driver Passwords Securely with Named Passwords	54
6.7.1	Using Designer to Configure Named Passwords	54
6.7.2	Using iManager to Configure Named Passwords	55
6.7.3	Using Named Passwords in Driver Policies	56
6.7.4	Using the DirXML Command Line Utility to Configure Named Passwords	57
6.8	Adding a Driver Heartbeat	60
<b>7</b>	<b>Synchronizing Objects</b>	<b>63</b>
7.1	What Is Synchronization?	63
7.2	When Is Synchronization Done?	63
7.3	How Does the Metadirectory Engine Decide Which Object to Synchronize?	64
7.4	How Does Synchronization Work?	65
7.4.1	Scenario One	65
7.4.2	Scenario Two	67
7.4.3	Scenario Three	68
<b>8</b>	<b>Troubleshooting the Driver</b>	<b>69</b>
8.1	Resolving Errors	69
8.1.1	The Event Server Does Not Load	69
8.1.2	The Driver Does Not Start	69
8.1.3	The Driver Is Not Communicating with the Event Server	69
8.1.4	The Event Server Receives Message Agent Errors	70
8.1.5	Attributes Do Not Get Refreshed on the Data Map Object	70
8.1.6	Driver Only Appears to Process Transactions	70
8.1.7	Data Does Not Show up in the Identity Vault on the Publisher Channel	71
8.1.8	Data Does Not Update in PeopleSoft on the Subscriber Channel	71
8.1.9	No Transactions Are Coming Across the Publisher Channel	71
8.1.10	Transactions Do Not Get Placed in the PeopleSoft Queue	71
8.1.11	No Data Is Returned When Running the Message Test Program	71
8.1.12	Transactions Are Left in Selected State and Not Processed	72
8.1.13	Receiving Errors on the Publisher Channel When Processing a Transaction	72
8.1.14	Message Agent Relationships Are Not Functioning	72
8.2	Troubleshooting Driver Processes	74
8.2.1	Viewing Driver Processes	74

<b>9</b>	<b>Backing Up the Driver</b>	<b>81</b>
9.1	Exporting the Driver in Designer . . . . .	81
9.2	Exporting the Driver in iManager . . . . .	81
<b>10</b>	<b>Security: Best Practices</b>	<b>83</b>
<b>A</b>	<b>DirXML Command Line Utility</b>	<b>85</b>
A.1	Interactive Mode . . . . .	85
A.2	Command Line Mode . . . . .	94
<b>B</b>	<b>Properties of the Driver</b>	<b>99</b>
B.1	Driver Configuration . . . . .	99
B.1.1	Driver Module . . . . .	99
B.1.2	Driver Object Password . . . . .	100
B.1.3	Authentication . . . . .	101
B.1.4	Startup Option . . . . .	102
B.1.5	Driver Parameters . . . . .	103
B.2	Global Configuration Values . . . . .	104
B.3	Named Passwords . . . . .	105
B.4	Engine Control Values . . . . .	106
B.5	Log Level . . . . .	107
B.6	Driver Image . . . . .	108
B.7	Security Equals . . . . .	109
B.8	Filter . . . . .	109
B.9	Edit Filter XML . . . . .	109
B.10	Misc . . . . .	110
B.11	Excluded Users . . . . .	110
B.12	Driver Manifest . . . . .	111
B.13	Inspector . . . . .	111
B.14	Server Variables . . . . .	111





# About This Guide

This document is for network administrators, consultants, and PeopleSoft® administrators who are using PeopleSoft 7.5 or 8.1.

The Identity Manager Driver for PeopleSoft, subsequently referred to as the driver, is designed to share data between the Novell® Identity Vault and PeopleSoft. This configurable solution gives organizations the ability to increase productivity and streamline business processes by integrating PeopleSoft and the Identity Vault.

This driver connects to PeopleSoft via the Message Agent, a PeopleTools interface. The driver can be configured to work with any PeopleSoft module.

This guide provides an overview of the driver's technology as well as configuration instructions.

- ♦ Chapter 1, "Introducing the Identity Manager Driver for PeopleSoft," on page 11
- ♦ Chapter 2, "Installing the Driver," on page 21
- ♦ Chapter 4, "Configuring Your PeopleSoft System," on page 29
- ♦ Chapter 8, "Troubleshooting the Driver," on page 69

## Audience

This guide is intended for consultants, administrators, and IS personnel who need to install, configure, and maintain the Identity Manager Driver 3.7 for PeopleSoft.

## Feedback

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

## Documentation Updates

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

## Additional Documentation

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

## Documentation Conventions

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

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

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

# Introducing the Identity Manager Driver for PeopleSoft

# 1

A PeopleSoft application is composed of two primary pieces:

- ♦ PeopleTools, which is an infrastructure
- ♦ One or more modules

Example modules are Human Resources Management System (HRMS), Financials, Student Administration, and Customer Resource Management.

The Identity Manager Driver for PeopleSoft interacts with PeopleSoft at the PeopleTools level. By using object definitions within the PeopleSoft modules, along with a collection of preconfigured objects, you enable PeopleSoft so that you can do the following:

- ♦ Trap events in any PeopleSoft module.
- ♦ Expose trapped events to the Novell® Identity Vault.
- ♦ Collect and process the exposed data.
- ♦ Expose the data through a PeopleTools interface.

The driver creates an automated link between PeopleSoft and the Identity Vault. As new records are added, modified, or deactivated (disabled) in PeopleSoft, network tasks associated with these events can be automatically processed.

For example, when hires, re-hires, terminations, and employee updates occur within the Human Resources module, resultant tasks are often created for the Information Services department. These tasks might include setting up, modifying or disabling an Identity Vault user account, creating an e-mail account, or creating a domain account. By using the driver, you can automate and maintain these and other business processes.

- ♦ [Section 1.1, “Changes in Terminology,” on page 11](#)
- ♦ [Section 1.2, “Benefits,” on page 12](#)
- ♦ [Section 1.3, “Driver Features,” on page 12](#)
- ♦ [Section 1.4, “Key Driver Features,” on page 13](#)
- ♦ [Section 1.5, “Driver Components,” on page 14](#)
- ♦ [Section 1.6, “Publishing to the Identity Vault,” on page 16](#)
- ♦ [Section 1.7, “Subscribing from the Identity Vault,” on page 18](#)

## 1.1 Changes in Terminology

The following terms have changed from earlier releases:

**Table 1-1** *Changes in Terminology*

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

## 1.2 Benefits

As the following examples illustrate, the driver enables you to automate and maintain business processes. You can:

- ♦ Automatically create an Identity Vault account when an individual is hired or a student is admitted.
- ♦ Automatically delete or deactivate Identity Vault accounts when an employee terminates.
- ♦ Synchronize bidirectional data between PeopleSoft and the Identity Vault.
- ♦ Maintain accurate and consistent Identity Vault IDs.
- ♦ Define password policies (for example, a birthdate, social security number, and first and last name combinations).
- ♦ Via Identity Vault, seamlessly allow integration between PeopleSoft and multiple applications (for example, eDirectory, Lotus Notes\*, Netscape\*, Exchange, Active Directory\*).
- ♦ Create other eDirectory objects associated with a PeopleSoft object (for example, account codes or department records).
- ♦ Synchronize attributes between PeopleSoft and the Identity Vault.
- ♦ Synchronize data from the Identity Vault to PeopleSoft.

The driver includes the following:

- ♦ The Event Server program
- ♦ The driver shim
- ♦ The PeopleSoft Service Agent (PSA)
- ♦ A driver configuration file

You can configure the PSA and the driver objects to enhance your organization's business processes. Before installing and configuring the driver, you evaluate and define those processes. During installation, you configure the driver's policies to automate these processes wherever possible.

## 1.3 Driver Features

The driver provides the following features:

- ♦ Support for multiple versions of PeopleTools.

A separate Event Server program is available for each PeopleTools release. This program connects to the driver shim in a Windows\* socket.

Remote processing between the shim and the Event Server is also possible. This means that the Event Server can run on a different machine than the driver shim.

- ◆ Enhanced error processing.

The Event Server can run as a DOS window on a Windows\* NT machine. The screen shows debug messages as an event is processed. The information includes the status of the transaction displayed on the Event Server. This enables you to determine exactly which transaction is being processed.

- ◆ Enhanced DSTRACE capability.

DSTRACE messaging includes a detailed description of the event and event processing.

- ◆ Identity Manager query support.

You can apply a Matching policy for both the Publisher and Subscriber channels. If an association between the User object and the PeopleSoft employee does not exist, the Metadirectory engine executes a Matching policy and requests a query to merge attributes between the two objects.

You can also query PeopleSoft for attribute values needed for data manipulation or processing within a policy without copying the data to the Identity Vault.

- ◆ Command line parameters for the Event Server.

Command line parameters define how the Event Server is to be executed, how to connect to the driver shim, and whether the Event Server should be installed as a service. The parameters also designate the path of the PeopleSoft client environment. For more information, refer to [Section 4.1, “Installing the PeopleSoft Service Agent,” on page 29.](#)

- ◆ Auto-start ability.

You can set the driver to auto-start by installing the Event Server as a Windows service and setting the auto-start flag on the driver. For more information, refer to [Section 4.3, “Installing the Event Server,” on page 40.](#)

- ◆ Support for schema query.

The driver can query the Identity Vault and PeopleSoft schemas for attributes that can be used in the Mapping policy, or query PeopleSoft for objects defined as policy objects on the Driver object.

### 1.3.1 Identity Manager 3.5 New Features

For information about the new features in Identity Manager, see [“What's New in Identity Manager 3.5?” in \*Identity Manager 3.5 Installation Guide\*.](#)

## 1.4 Key Driver Features

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

- ◆ [Section 1.4.1, “Local Platforms,” on page 14](#)
- ◆ [Section 1.4.2, “Remote Platforms,” on page 14](#)
- ◆ [Section 1.4.3, “Role-Based Entitlements,” on page 14](#)

## 1.4.1 Local Platforms

The PeopleSoft driver can be installed locally on the following platforms:

- ♦ Windows\* NT\*, 2000, or 2003 with the latest Service Patch

## 1.4.2 Remote Platforms

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

- ♦ Windows NT, 2000, or 2003 with the latest Service Patch

For more information about installing the Remote Loader services, see “[Installing Remote Loaders](#)” in the *Novell Identity Manager 3.5 Administration Guide*.

## 1.4.3 Role-Based Entitlements

The PeopleSoft driver does not have Role-Based entitlement functionality defined with the default configuration files. The driver does support entitlements, if there are policies created for the driver to consume.

# 1.5 Driver Components

The driver uses the following components:

- ♦ PeopleSoft Service Agent (PSA)
- ♦ Driver Shim
- ♦ XML Import File
- ♦ Event Server
- ♦ PeopleSoft Message Agent (delivered through PeopleSoft)

**Table 1-2** *Components needed for the different PeopleSoft versions*

Component	For PeopleSoft 7.5	For PeopleSoft 8.1
PSA	dirxml_driver 3_6 psa.exe	dirxml_driver 3_6 psa.exe
Driver shim	npsshim.dll	npsshim.dll
XML Import File	peoplesoft36-IDM3_5_0-V1.xml (iManager)	peoplesoft36-IDM3_5_0-V1.xml (iManager)
	peoplesoft36-IDM3_0_1-V1.xml (Designer)	peoplesoft36-IDM3_0_1-V1.xml (Designer)
Event Server	nps75eventserver.exe	nps81eventserver.exe

## 1.5.1 Driver Configuration

The driver is responsible for reporting object change events that it receives from PeopleSoft to the Metadirectory engine. The driver consists of both an Event Server program and the `npsshim.dll` file.

The driver contains both a Publisher channel and a Subscriber channel. The Subscriber channel receives XML-formatted eDirectory events from the Metadirectory engine. The driver then converts these documents to an appropriate data stream, forwards them to the Event Server, and updates PeopleSoft via the Message Agent interface.

The Publisher channel receives XML-formatted PeopleSoft events from the Event Server and submits them to the Metadirectory engine for publication into the Identity Vault. The engine processes the document by sequentially applying all configured policies based on standard driver process flow.

Each policy performs a transformation on the XML document. The engine processes the event document according to the policies and the filter found in the Driver Publisher object.

## 1.5.2 Driver Shim

The driver shim handles communication between the event server and the Metadirectory engine. For this version of the driver, the driver shim is named `npsshim.dll`.

## 1.5.3 Event Server

The Event Server is a standalone executable process. It establishes and maintains connectivity between the driver shim and the PeopleSoft Message Agent.

The Event Server communicates with the driver via a sockets interface and communicates with the Message Agent via proprietary PeopleSoft Message Agent APIs.

The Event Server is a bidirectional component with Publisher and Subscriber channels. The Publisher channel polls the Message Agent for transactions. When transactions are available, the Event Server reads the record data via the Message Agent and transforms the proprietary format of the PeopleSoft records into an XML-formatted representation. The XML records are sent to the driver for submission to the Identity Vault.

Likewise, the Subscriber channel receives XML-formatted eDirectory events from the driver and transforms them into the PeopleSoft proprietary format. The event data is then sent to the Message Agent for publication into the PeopleSoft database.

## 1.5.4 PeopleSoft Service Agent

The PeopleSoft Service Agent (PSA) is software that you receive as a PeopleSoft project. A PSA is a collection of objects. This collection customizes the PeopleSoft module that the objects are applied to.

You can install the PSA components and then configure them for any PeopleSoft application.

The PSA serves three purposes:

- ♦ Traps events as they occur in PeopleSoft

- ◆ Places a transaction for trapped events into the worklist
- ◆ Exposes the transaction (which is joined with relevant, current data) to the driver via the Message Agent

The PSA includes a sample of all objects necessary in PeopleSoft so that any PeopleSoft module can have connectivity to and from the Event Server.

## 1.6 Publishing to the Identity Vault

Publishing events to the Identity Vault begins with updates in PeopleSoft. As these updates occur, transactions are placed in a worklist queue. These events are then made available to the Identity Vault through the Event Server.

As the Event Server receives transactions from PeopleSoft, it transforms the data into an XML document and passes the document to the driver shim. The driver shim then passes the document to the Metadirectory engine, which processes the transaction in the Identity Vault based on the policies as defined in the driver.

The transaction in the worklist is also updated with either a *worked* or *error* status. As the status is being updated in PeopleSoft, additional PeopleCode can be processed to trigger e-mail notifications to users in a defined PeopleSoft workflow role. The e-mail message communicates information regarding both successful transactions and transactions that create errors.

### 1.6.1 Event Descriptions

As updates or actions occur within PeopleSoft, workflow triggers are executed. These triggers place transactions in a PeopleSoft worklist queue. Each transaction is given key fields that uniquely identify the transaction to the driver.

One of these key fields is the Event Name field. Event Names are assigned based on a PeopleSoft action. The driver monitors the queue, checking for transactions that meet the criteria for processing.

The criteria for processing include the following:

- ◆ The transactions have the status of 0.
- ◆ Action, date, and time are less than or equal to current date and time.

After receiving the event, the Event Server converts the event into XML and sends the XML to the driver shim. This shim passes the XML document to the Metadirectory engine. The engine then applies the appropriate policies for that event type (based on the EventName field). The driver changes the XML to Identity Vault commands and sends the events to the Identity Vault.

The delivered solution supports the following workflow events. By default, an event in PeopleSoft equates to an XML functional event or document.

**Table 1-3** *PeopleSoft To XML Events*

PeopleSoft Event	Typical PeopleSoft Action	XML Functional Event
ADD	New hire, new student, new account code, new record.	Add



PeopleSoft Event	Typical PeopleSoft Action	XML Functional Event
UPD	Any change or modification to data. For example, a field on a panel is modified.	Modify
DIS	Termination. A record becomes inactive or disabled.	Delete

To trigger the appropriate event for the type of XML document that is to be generated, you must configure PeopleSoft appropriately. Without proper configuration and review of your business processes, a termination (action) could trigger a UPD event instead of a DIS event. The resulting XML document would be a Modify document instead of a Delete document.

The delivered driver configuration identifies two fields (Email ID and Description) on the Subscriber channel. These fields synchronize data from the Identity Vault to PeopleSoft. Additional fields can be subscribed to PeopleSoft from the Identity Vault, based on the needs of your organization. For Subscriber channel information, refer to [“Subscribing from the Identity Vault” on page 18](#).

## ADD

The ADD event is generally triggered when a new record is added. For the HR module, this is when an employee is hired. When a new Hire record is created within PeopleSoft, the following steps occur:

1. A workflow definition is triggered.
2. An ADD transaction is written to the worklist.
3. The driver reads the worklist to obtain a transaction, which is joined with user-specified data.
4. The driver transforms the data stream into an XML document.
5. The driver passes the document to the Metadirectory engine.
6. The engine applies the policies that have been configured.
7. The engine creates an eDirectory object.

The SSCreate policy is configured as a sample Create policy. You use this policy to create the User object. You can define the eDirectory Common Name, Initial Login Password, and other Create policies within this policy.

Depending on your business requirements, you can apply additional configurations to the PSA to expose additional data elements. You can also configure the policies to meet business objectives.

During the ADD event processing, an association is made between the employee’s PeopleSoft record and the eDirectory object. For the HR module, the EmplID field is used as the key field and saved in the association attribute on the User object.

When an object is created, various attributes or data elements defined in PeopleSoft could be passed through the interface and used to place the object. The Placement policy on the Driver object would use these values.

## UPD

UPD events are generally triggered when data on a specified PeopleSoft field is modified. A workflow is triggered through the use of PeopleCode, and a UPD event is written to the worklist. Through the same process as the ADD transaction, the driver retrieves the transaction and applies

the appropriate policies. In the delivered solution, the eventXform policy is used to apply these events. The policies might vary, based on the individual needs of an organization.

To determine the PeopleSoft fields that should be synchronized and used as part of the policies within the Identity Vault, analyze your business processes.

For example, a user's location, department, or company can determine the location of that individual's User object within eDirectory. If one of these fields is updated from PeopleSoft, the eDirectory User object must be moved to the appropriate container, based on the new information.

PeopleSoft fields can also be stored in the Identity Vault as attributes. Telephone numbers, department, preferred names, business titles, and locations are some common fields that can be shared between the two systems. As you analyze your business processes and needs, you determine the necessary data mappings between the systems.

## DIS

The DIS event is generally triggered when records are disabled. Many different actions within PeopleSoft represent a disabled process. In the HR module, this event occurs when an employee terminates. When a termination record is created within PeopleSoft, a workflow is triggered and a DIS entry is written to the worklist.

---

**IMPORTANT:** If a DIS transaction is triggered, a Delete XML document is the result.

---

An organization can choose to delete, disable, or disable and move the Identity Vault account. As with the modify/update process, the eventXform policy is delivered to show how a Delete XML document is transformed into a Modify document.

---

**NOTE:** By default, when the Publisher channel processes any event, the eDirectory ID and eDirectory DN (distinguished name) are updated in PeopleSoft. You can disable this process by updating the driver's XML properties.

---

## 1.7 Subscribing from the Identity Vault

PeopleSoft is the authoritative process for ADD and DIS (delete or disable) events. Therefore, the default driver only allows updates to PeopleSoft on the Subscriber channel.

Because PeopleSoft is a relational database (instead of a hierarchical database), the Move and Rename processes generally do not apply. If these processes occur in the Identity Vault against an object that the driver is subscribed to, the driver converts them to a Modify event. The driver does this to update the eDirectory ID and eDirectory Distinguished Name in PeopleSoft, if the update process is not disabled.

The driver subscribes to Modify events that occur within the Identity Vault. PeopleSoft is designed to be the authoritative owner of data in the delivered solution. Therefore, if a User object is created in eDirectory, an employee record in PeopleSoft is not created with the default configuration.

The data elements that are being subscribed from the Identity Vault to PeopleSoft must

- Be found on the Subscriber channel filter
- Exist on the DIRXML\_SCHEMA01\_UPDATE Message Definition and on the DIRXML\_SCHEMA01 Message Definition in PeopleSoft (or equivalent definitions)

When the driver is notified of a modification to an attribute selected in its Subscriber filter, the driver sends the event to the Event Server. The Event Server formats the event and then updates PeopleSoft by communicating through the PeopleSoft Message Agent. Data received from the Identity Vault can be placed on a panel or page inside PeopleSoft.

Because this data is being updated and maintained outside of PeopleSoft, it should typically be set to Display Only from within PeopleSoft. It is not advisable to allow the same data element to be modified inside and outside of PeopleSoft. For this to occur, the data element would need to be defined in both the Publisher and the Subscriber channel on the driver.

You can update data in PeopleSoft to an existing PeopleSoft record. To do this, you must write appropriate PeopleCode so that data elements updated on the DIRXML\_STAGE01 record definition are also copied to the desired record within the PeopleSoft environment.



# Installing the Driver

# 2

- ♦ [Section 2.1, “Driver Requirements,” on page 21](#)
- ♦ [Section 2.2, “Installation Instructions,” on page 21](#)
- ♦ [Section 2.3, “Importing the PeopleSoft Driver,” on page 21](#)

## 2.1 Driver Requirements

- ❑ Novell® Identity Manager 3.5
- ❑ Windows NT\* 4.0 with Service Pack 5 or higher, Windows 2000, or Windows 2003 with the latest Service Patch
- ❑ The appropriate version of PeopleTools and Tuxedo\* Application Server

PeopleSoft Platform	PeopleTools Version
PeopleSoft 7.5	PeopleTools 7.57 or higher
PeopleSoft 8.1	PeopleTools 8.17 or higher

## 2.2 Installation Instructions

You install the driver as part of the Novell Identity Manager 3.5 installation program. For installation instructions, refer to the “[Installing Identity Manager](#)” chapter of *Identity Manager 3.5 Installation Guide*.

## 2.3 Importing the PeopleSoft Driver

This section talks briefly about importing the PeopleSoft driver through Designer and iManager utilities.

- ♦ [Section 2.3.1, “Importing the Driver Configuration File in Designer,” on page 21](#)
- ♦ [Section 2.3.2, “Importing the Driver Configuration in iManager,” on page 22](#)

### 2.3.1 Importing the Driver Configuration File in Designer

Designer allows you to import the basic driver configuration file for PeopleSoft. This file creates and configures the objects and policies needed to make the driver work properly. The following instructions explain how to create the driver and import the driver’s configuration.

There are many different ways of importing the driver configuration file. This procedure only documents one way.

- 1 Open a project in Designer and in the Modeler, right-click the Driver Set object, then select *New > Driver*.
- 2 From the drop-down list in the Driver Configuration Wizard, select *PeopleSoft36-IDM3\_0\_1-V1.xml*.

- 3 Select the *Perform required prompt checking* check box.
- 4 Click *Run*.
- 5 Configure the driver by filling in the fields. Specify information for your environment.
- 6 After specifying parameters, click *OK* to import the driver.
- 7 After the driver is imported, customize and test the driver.
- 8 After the driver is fully tested, deploy the driver into the Identity Vault. See “[Deploying and Exporting](#)” in the *Designer 2.0 for Identity Manager 3.5*.

## 2.3.2 Importing the Driver Configuration in iManager

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

The following instructions explain how to create the driver and import the driver’s configuration.

- 1 In Novell iManager, click *Identity Manager Utilities > New Driver*.
- 2 Select a driver set, then click *Next*.  
If you place this driver in a new driver set, you must specify a driver set name, context, and associated server.
- 3 Select *Import a configuration from the server (.XML file)*, then select *PeopleSoft36-IDM3\_5\_0-V1.xml*.  
The driver configuration files are installed on the Web server when you install Identity Manager. During the import, you are prompted for the driver’s parameters and other information.
- 4 Specify the driver’s parameters, then click *OK* to import the driver. See [Table 2-1 on page 22](#) for a list of parameters you can set.  
When the import is finished, you can define security equivalences and exclude administrative roles from replication.  
The driver object must be granted sufficient eDirectory rights to any object it reads or writes. You can do this by granting Security Equivalence to the driver object. The driver must have Read/Write access to users, post offices, resources, and distribution lists, and Create, Read, and Write rights to the post office container. Normally, the driver should be given security equal to Admin.
- 5 Review the driver objects on the Summary page, then click *Finish*.

**Table 2-1** *Driver Configuration Parameters*

Parameter	Description
Driver name	The actual name you want to use for the driver.
Active Users Container	The name of the Organizational Unit object where Active users are placed. You can modify this option through the driver’s global configuration variable (GCV).
Inactive Users Container	The name of the Organizational Unit where Inactive users are placed. You can modify this option through the driver’s GCV.

Parameter	Description
Active Employees Group	The name of the Group object to which Active Employee users are added. You can modify this option through the driver's GCV.
Active Managers Group	The name of the Group object to which Active Manager users are added. You can modify this option through the driver's GCV.
Event Server Host Name and Port	The host name or IP address and port number of the computer where the Event Server that connects to PeopleSoft is running. The default port is 16500.
PeopleSoft Connection String	The host name or IP address and port number for connecting to the appropriate PeopleSoft Application server. This is typically referred to as the PeopleSoft application server connection string. The default port is 9000.
PeopleSoft User ID	The PeopleSoft User ID the driver uses for PeopleSoft authentication.
PeopleSoft User Password	The PeopleSoft User password this driver uses for PeopleSoft authentication.
Password Failure Notification User	Password synchronization policies are configured to send e-mail notifications to the associated user when password updates fail. You have the option of sending a copy of the notification e-mail to another user, such as a security administrator. If you want to send a copy, specify the DN of that user. Otherwise, leave this field blank.
Driver is Remote/Local	Configure the driver for use with the Remote Loader service by selecting the <i>Remote</i> option, or select <i>Local</i> to configure the driver for local use. If <i>Local</i> is selected, you can skip the remaining parameters.
Remote Host Name and Port	Specify the hostname or IP address and port number where the Remote Loader service has been installed and is running for this driver. The default port is 8090.
Driver Password	The driver object password is used by the Remote Loader to authenticate itself to the Metadirectory server. It must be the same password that is specified as the driver object password on the 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 Remote Loader.

The additional driver parameters are set to default values during the import process, but they can be modified in iManager (by clicking the *Driver Configuration* tab on the driver object.)

**Table 2-2** *Additional Driver Parameters*

Parameter	Description
Event Server Host Name	The host name or IP address of the Event Server that connects to where PeopleSoft is running.
Event Server Port Number	The IP port number on which the driver Event Server component listens for connection from the driver shim.

Parameter	Description
Data Record ID Field	The name of the field in the Data Schema CI that uniquely identifies a PeopleSoft object. The value in this field uniquely identifies a PeopleSoft object. The value in this field is used as the DirXML® object association identifier. The default field name is DIRXML_ASSOC_ID.
eDirectory Object Class Name	The eDirectory™ object Class to which the Data Record definition maps. The default value is User.
Schema Data Subscriber Activity	<p>The name of the PeopleSoft Activity object that defines the data that can be synchronized from the Identity Vault to the PeopleSoft application. The default Subscriber Activity is DIRXML_SCHEMA01_UPDATE.</p> <p>The Message Definition within this Activity must match the name of the Activity.</p>
Schema Data Query Activity	<p>The name of the PeopleSoft Activity object that defines the data that the driver reads from the PeopleSoft application. This data is usually the subset of application data elements contained within the Schema Data Publisher Activity. The default Query Activity is DIRXML_SCHEMA01_QUERY.</p> <p>The Message Definition within this Activity must match the name of the Activity.</p>
Queue Poll Interval (seconds)	The number of seconds the driver waits between attempts to process transaction records. This poll interval is only applied when no transactions are available for processing. The default poll interval is 5 seconds.
Queue Retrieval Limit	This parameter specifies how many transactions are retrieved by the driver from the PeopleSoft worklist queue each time the driver accesses the Transaction Access Activity. The default value is 5.
Transaction Access Activity	<p>The name of the PeopleSoft Activity object that defines the set of fields required for the DirXML Transaction interface. The set of fields in the specified transaction activity must contain the same fields and keys identified in the default transaction activity in order for the driver to work. The default Transaction Activity is DIRXML_TRANS01.</p> <p>The Message Definition within this Activity must match the name of the Activity.</p>
Schema Data Publisher Activity	<p>The name of the PeopleSoft Activity object that defines the set of data to be synchronized from the PeopleSoft application to the Identity Vault.</p> <p>The default Publisher Activity is DIRXML_SCHEMA01.</p> <p>You should ensure that the Message Definition within this Activity matches the name of the Activity.</p>
<p><b>NOTE:</b> Additional transaction control key fields (field names are all capitalized) contained within the Message Definition are used for processing purposes only and are not synchronized.</p>	



# Upgrading the Driver

# 3

If you have been using a previous version of the driver, follow these instructions instead of the ones in [Chapter 2, “Installing the Driver,”](#) on page 21.

Identity Manager 3.5 contains a new architecture for how policies reference one another. To take advantage of this new architecture, the driver configuration file provided for PeopleSoft must be upgraded. For more information on the new architecture, see “[Upgrading Identity Manager Policies](#)” in the [Understanding Policies for Identity Manager 3.5](#). You can upgrade the driver in Designer or iManager.

- ♦ [Section 3.1, “Upgrading the Driver in Designer,”](#) on page 25
- ♦ [Section 3.2, “Upgrading the Driver in iManager,”](#) on page 27

## 3.1 Upgrading the Driver in Designer

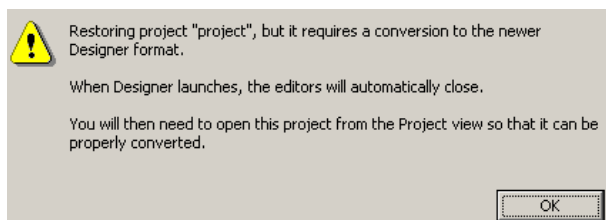
- 1 Make sure you have updated your driver with all the patches for the version you are currently running.

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

- 2 Back up the driver. See [Chapter 9, “Backing Up the Driver,”](#) on page 81 for instruction on how to back up the driver.
- 3 Install Designer version 2.0 or above, then launch Designer.

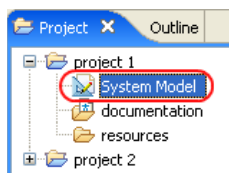
If you had a project open in Designer when you upgraded Designer, proceed to [Step 4](#). If you didn’t have a project open in Designer when you upgraded Designer, skip to [Step 5](#).

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

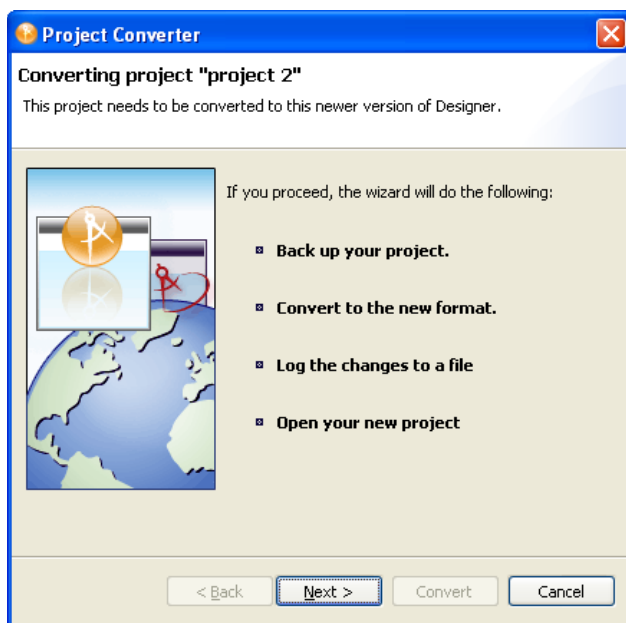


Designer closes the project to preform the upgrade.

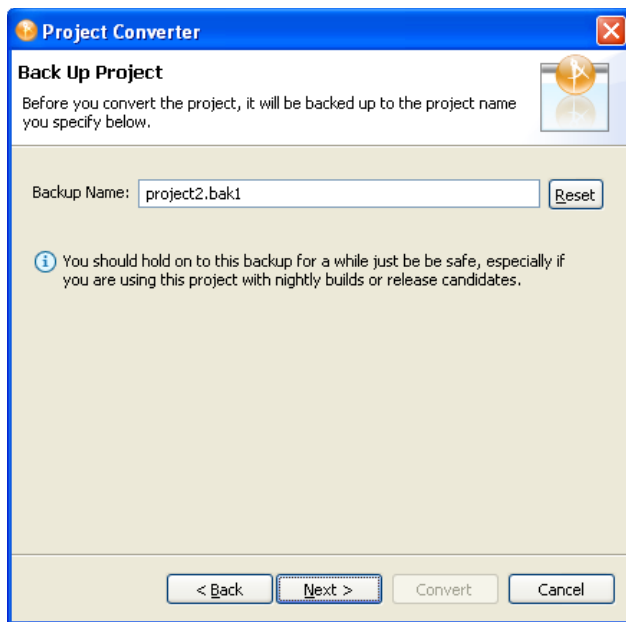
- 5 In the Project view, double-click *System Model* to open and convert the project.



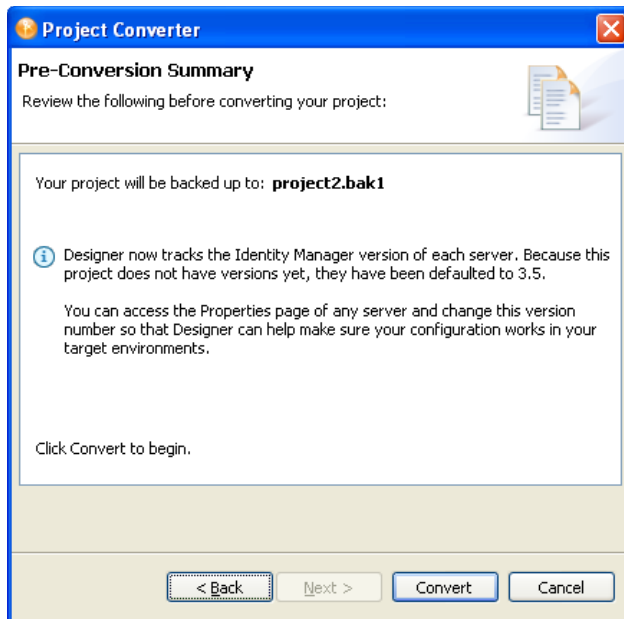
- 6 Read the Project Converter message explaining that the project is backed up, converted to the new format, changes logged to a file, and the new project is opened, then click *Next*.



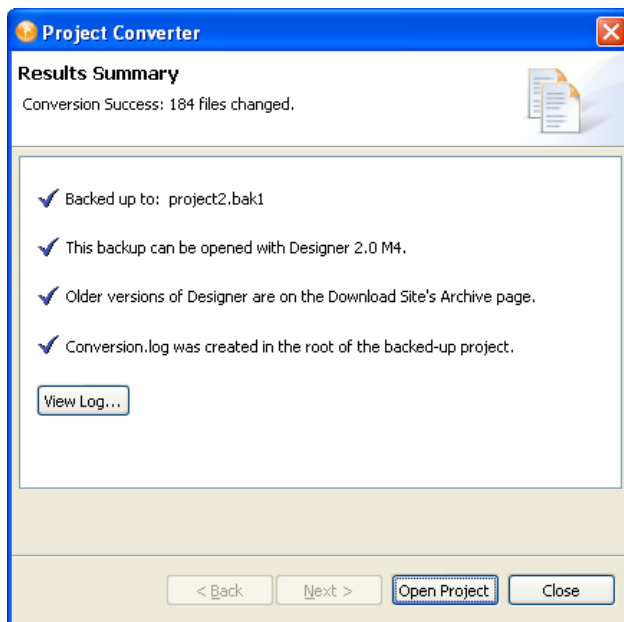
- 7 Specify the name of the backup project name, then click *Next*.



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



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



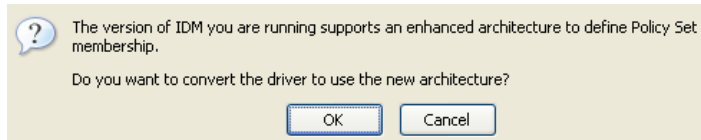
If you want to view the log file that is generated, click *View Log*.

## 3.2 Upgrading the Driver in iManager

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

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

- 2 Back up the driver. See [Chapter 9, “Backing Up the Driver,”](#) on page 81 for instruction on how to back up the driver.
- 3 Verify that Identity Manager 3.5 has been installed and you have the current plug-ins installed, then launch iManager.
- 4 Click *Identity Manager > Identity Manager Overview*.
- 5 Click *Search* to find the Driver Set object, then click the driver you want to upgrade.
- 6 Read the message that is displayed, then click *OK*.



- 7 If there is more than one driver to upgrade, repeat [Step 2](#) through [Step 6](#).

# Configuring Your PeopleSoft System

# 4

Use the information in the following section to configure your PeopleSoft system to share and synchronize data with the Novell® Identity Vault.

- ♦ [Section 4.1, “Installing the PeopleSoft Service Agent,” on page 29](#)
- ♦ [Section 4.2, “Running the Message Agent Test Program,” on page 37](#)
- ♦ [Section 4.3, “Installing the Event Server,” on page 40](#)

## 4.1 Installing the PeopleSoft Service Agent

The process for installing the PSA is different for different versions of PeopleSoft.

- ♦ [Section 4.1.1, “Installing the PSA on PeopleSoft 7.5,” on page 29](#)
- ♦ [Section 4.1.2, “Installing the PSA on PeopleSoft 8.1,” on page 33](#)

---

**NOTE:** Any references to specific paths in these sections represent the defaults indicated during the installation procedures. Apply the necessary changes if applicable.

---

### 4.1.1 Installing the PSA on PeopleSoft 7.5

If you are using PeopleSoft 7.5, you should complete the following tasks to install the PSA:

- ♦ [“Extracting the PSA Files” on page 29](#)
- ♦ [“Installing the PeopleSoft Service Agent” on page 30](#)
- ♦ [“Installing the PSA for PeopleSoft 7.5” on page 30](#)
- ♦ [“Copying the Novell Integration Project into a Target Database” on page 31](#)
- ♦ [“Building Project Record Definitions” on page 31](#)
- ♦ [“Establishing Roles and Initiating Workflow” on page 32](#)
- ♦ [“Establishing Security for PSA Objects in PeopleSoft 7.5” on page 32](#)

#### Extracting the PSA Files

The PSA project comes as a self-extracting project. To extract the project:

- 1 From the location you specified when you installed the components for PeopleSoft, run `dirxml_driver_36.exe`. The default location is `Novell\nds\ps`.
- 2 Click *Next*, select the destination directory, then click *Next* to begin extracting the PSA files.
- 3 Click *Finish* to close the extraction process.

The PSA is now ready to be installed into your PeopleSoft environment.

## Installing the PeopleSoft Service Agent

Before you can install and configure the PeopleSoft objects, you need access to a PeopleSoft user ID and password with Administrator or appropriate developer rights. You can create a unique user ID and password for implementing these objects.

To apply the delivered PSA set of objects, follow PeopleSoft's standard upgrade procedures. For information on PeopleSoft upgrade procedures, refer to PeopleSoft documentation.

## Installing the PSA for PeopleSoft 7.5

Data Mover is an application tool used to import external PeopleSoft objects into an existing PeopleSoft 7.5 database. Using the standard PeopleSoft upgrade process, you use Data Mover to move the Identity Manager for PeopleSoft projects into an organization's PeopleSoft environment.

A Data Mover script (DMS) is provided with the driver. This script imports new and modified objects into the data dictionary tables. You should only run this script on a refreshed Application Update Database (AUD). Refreshing the database resolves any conflicts between the delivered objects and any objects that already exist in the AUD database.

---

**WARNING:** If the DMS is run against a database other than the AUD, it can fail or overwrite existing data. If you run the script directly against the PeopleSoft development database, it corrupts the database.

---

To ensure that the DMS files were copied during the component installation, check the C:\PSA directory. This is the default directory that the installation program uses. If you altered this setting during the installation, check the directory that you specified. The following are default steps to import the external PeopleSoft objects:

- 1 From the Start menu, click *Programs > PeopleSoft > Data Mover*.
- 2 Log in to the AUD database.
- 3 Select *File > Open*.
- 4 Browse to the appropriate file for your installation, then select the DMS script with the appropriate release number. (For example, *DIRXML\_DRIVER\_PSA\_36IMP.DMS*.)  
The release number should be the same as the release number of the PeopleSoft database.
- 5 In the Data Mover syntax dialog box, verify that the input and log files are pointing to the directory containing the data (.dat) files on the local drive.

The following is an example of the delivered syntax:

```
SET OUTPUT c:\PSA\AU_CHECK.DAT;  
SET INPUT c:\PSA\PSA-PS75\DIRXML_DRIVER_PSA_36IMP.DAT;  
SET LOG c:\PSA\DIRXML_DRIVER_PSA_36IMP.LOG;  
SET NO TRACE;  
SET IGNORE_DUPS;  
EXPORT AU_CHECK;  
IMPORT *;
```

- 6 Select *File > Run Script*.

From the report in the bottom status panel, verify if the import was successful. If the import fails, make sure the input and output paths point the proper directories.

- 7 Click *Close* to close the Data Mover.

## Copying the Novell Integration Project into a Target Database

Your target database contains the system catalog, data dictionary structures, and sample application data. When moving from an AUD database to the target database, objects from the data dictionary in the AUD are copied to the data dictionary of the target database.

After the copy is complete, you must build the database. The build creates the physical tables or views in the underlying database so that the objects are accessible. You should be logged in as an administration user when completing these procedures.

We recommend the following:

- ♦ Use a clean copy of a target database for this procedure.

If you apply a copy to a database with production data, do a copy and compare before you apply the copy.

- ♦ Complete an installation and test that installation against a target database.

Do this before you change the configuration and before you apply the database to a development or production environment.

To copy the project into a target database:

- 1 From the *Start* menu, click *Programs > PeopleSoft > PeopleTools*.
- 2 Log in to the AUD database.
- 3 Click *Go > PeopleTools > Application Designer > File > Open*.
- 4 Click *Object Type*, then click *Project*.
- 5 Click *Selection Criteria, Name*, type *DIRXML\_DRIVER\_PSA\_36*, then click *Select*.
- 6 Click the project, then click *Select*.
- 7 Click *Tools > Upgrade > Copy*.
- 8 Type the target database name (the project will be copied to this database), *operator ID*, and *password*, then click *OK*.  
Be sure to select all objects with the project, including the project itself.
- 9 Check the *Export Project* box to export the objects to the target database, then click *Copy*.  
All objects are copied from the AUD to the target DMO database.
- 10 Click *Close* to close the Application Designer window.

## Building Project Record Definitions

After you have imported the project into the PeopleSoft database, you should build project record definitions and then build project views.

- 1 From the *Start* menu, click *Programs > PeopleSoft > PeopleTools*.
- 2 Log in to the target database using an administrator username that has administrative and development rights.
- 3 In the Application Designer, click *File*, then click *Open*.
- 4 Select the *DIRXML\_DRIVER\_PSA\_36* project, then click *Open*.
- 5 Click *Build > Project*.
- 6 From *Build > Options*, click *Create Tables and Execute SQL Now*.

- 7 Click *Build* to create sample project tables. After project tables are created, click *Close* to close the Build Progress window.  
You must create project tables before creating the views. Views are created using information from table fields.
- 8 In the Application Designer, select *Build > Project*.
- 9 In *Build Options*, click *Create Views and Execute SQL Now*.
- 10 Click *Build* to create the sample project views. After the views are created, click *Close* to close the Build Progress window.

You should now establish roles and initiate workflow. Refer to “[Establishing Roles and Initiating Workflow](#)” on page 32.

### Establishing Roles and Initiating Workflow

A role is a class of users who perform the same type of work. Business policies typically determine what role performs what activity. The eDirectory™ administrator has been designated as the role performing the ADD, UPD, and DIS events. This means that workflow must be activated and a role user must be assigned to the eDirectory administrator role. To activate a workflow, complete the following:

- 1 Click *Go > PeopleTools > Workflow Administrator*.
- 2 Click *Use > Workflow System Defaults > Defaults*.
- 3 Select the *Worklist Active* box.
- 4 If you are using e-mail and other electronic forms, select the applicable boxes.
- 5 Ensure that a default role user, typically *WF Admin*, is also assigned, then click *Save*. If workflow is not currently being used by other processes, change this role user to the role of eDirectory Administrator.

You should now assign security rights to the new objects. See “[Establishing Security for PSA Objects in PeopleSoft 7.5](#)” on page 32.

### Establishing Security for PSA Objects in PeopleSoft 7.5

You can assign security rights to all new panels and pages that have been added to a project. Use an administrator-type account to establish security.

#### Applying Security for the New PSA Objects

- 1 Click *Go > PeopleTools > Security Administrator*.
- 2 Click *File > Open*. Select the *ALLPANLS Operator Class*, then click *OK*.
- 3 Click the *Menu Items* icon from the left pane.
- 4 Click *Insert > Menu Name*.
- 5 Double-click *DirXML\_Administrator\_36*, click *Select All*, then click *OK*.
- 6 Click *Select All*, then click *OK*.
- 7 Click *Save*.
- 8 After you save, you can exit the Security Administrator interface.



## Verifying Access to PSA Objects

After you have applied security to the new PSA objects, you should use the sample application to verify that you can access, change, and save data.

You can test to ensure that transactions are created by entering a new person using the sample application. This example uses Departments, so you need to create a sample department, and then add a person (assigning him or her to that department) to validate that the application works.

- 1 Connect to the PeopleSoft database as administrator in two tier mode.
- 2 In the Application Designer, click *Go > DirXML > DirXML Administrator 36 EP*.
- 3 Select *Use > DirXML Sample Department*.
- 4 Click an empty *Department* field row to add a sample department and description.
- 5 Click *Save* to add the Department.
- 6 Select *Use > DirXML Sample People > Add*.
- 7 Specify an ID number for the new person, then click *OK*.
- 8 Provide data in the various fields for this ID, then click *Save*.  
(The required fields include: First Name, Last Name, Birth Date, Status, Title, and Department ID.)
- 9 To validate that an ADD transaction was created, access the *eDirectory Worklist Maintenance* menu. Click *Use > eDirectory Worklist Maintenance > eDirectory Worklist*. You can then search for the ID you just created.
- 10 Click the *Search* button.
- 11 Double-click the event you created. Look at the panels and verify that the data you entered exists.
- 12 Click *Close* to close the PeopleSoft Client.

Now that you have completed the required tasks for installing and configuring the PSA, you should continue with [“Running the Message Agent Test Program” on page 37](#).

### 4.1.2 Installing the PSA on PeopleSoft 8.1

Complete the following tasks to install the sample project for testing and configuration purposes:

- ♦ [“Extracting the PSA Files” on page 34](#)
- ♦ [“Importing the PSA Project into the PeopleSoft Database” on page 34](#)
- ♦ [“Building Project Record Definitions” on page 34](#)
- ♦ [“Testing Sample PeopleSoft Applications” on page 35](#)

---

**NOTE:** Any references to specific paths in these sections represent the defaults indicated during the installation procedures. Apply the necessary changes if applicable.

---

## Extracting the PSA Files

The PSA project comes as a self-extracting project. To extract the project:

- 1 From the location you specified when you installed the components for PeopleSoft, run `DIRXML_DRIVER_36.EXE`. The default location is `Novell\NDS\PS`.
- 2 Click *Next*, select the destination directory, then click *Next* to begin extracting the PSA files.
- 3 Click *Finish* to close the extraction process.

The PSA is now ready to be installed into your PeopleSoft environment.

## Importing the PSA Project into the PeopleSoft Database

With PeopleSoft 8, projects are delivered in a cache directory structure, which is similar to the cache structure found within PeopleSoft. With previous versions of PeopleSoft you needed to use the Data Mover script and compare, but this process is no longer necessary.

To import the PSA into the PeopleSoft database:

- 1 Connect to the PeopleSoft database as administrator in two tier mode.
- 2 From the Application Designer, select *File > Copy Project From File*.
- 3 Click *Browse* and select the PSA project directory: `C:\PSA\PSA-PSA8\`.
- 4 Click *Copy*.
- 5 With all object types selected, click *Copy* to copy all project components into the PeopleSoft database.

## Building Project Record Definitions

After you have imported the project into the PeopleSoft database, you should build project record definitions and then build project views.

- 1 Log into PeopleSoft using an administrator username that has administrative and development rights.
- 2 In the Application Designer, select *Build > Project*.
- 3 In *Build Options*, click *Create Tables and Execute SQL Now*.
- 4 Click *Build* to create sample project tables. After project tables are created, click *Close* to close the Build Progress window.

You must create project tables before creating the views. Views are created using information from table fields.

- 5 In the Application Designer, select *Build > Project*.
- 6 In *Build Options*, click *Create Views and Execute SQL Now*.
- 7 Click *Build* to create the sample project views. After the views are created, click *Close* to close the Build Progress window.

## Testing Sample PeopleSoft Applications

You can test to ensure that transactions are created by entering a new person using the sample application. This example uses Departments, so you need to create a sample department, and then add a person (assigning him or her to that department) to validate that the application works.

- 1 Connect to the PeopleSoft database as administrator in two tier mode.
- 2 From the Application Designer, select *Go > DirXML Administrator36*.
- 3 From the *DirXML Administrator* menu, select *Use > DirXML Sample Department*.
- 4 Click an empty *Department* field row to add a sample department and description.
- 5 Click *Save* to add the Department.
- 6 From the *DirXML Administrator 36* menu, click *Use > DirXML Sample People > Add*.
- 7 Specify an ID number, then click *OK*.
- 8 Provide data in the various fields for this ID, then click *Save*.  
(The required fields include: First Name, Last Name, Birth Date, Status, Title, and Department ID.)
- 9 To validate that an ADD transaction was created, click *Use > eDirectory Worklist Maintenance > eDirectory Worklist*.
- 10 Click the Search button to search for the user you added.

Date/Time	Event Name	Assoc IC
2002-09-24-16:40:27.000000	ADD	P0001
2002-09-25-13:40:42.000000	ADD	P0002
2002-09-25-13:44:01.000000	ADD	P0003
2002-09-25-13:48:30.000000	ADD	P0004
2002-09-25-14:02:56.000000	ADD	P0005
2002-09-25-14:07:45.000000	ADD	P0006
2002-09-25-14:19:29.000000	ADD	P0007
2002-09-25-14:32:26.000000	ADD	P0008
2002-09-25-14:32:50.000000	UPD	P0004
2002-09-27-14:41:50.000000	ADD	P1919
2002-09-27-15:50:51.000000	UPD	P1919
2002-10-04-23:15:15.000000	ADD	001

- 11 Double-click the event to view its details.

The screenshot shows the 'DirXML Administrator 36EP - Use - eDirectory Work List Maint' window. The 'eDirectory Worklist' tab is active, displaying details for a specific event. The event information is as follows:

<b>Bus Proc:</b>	DIRXML_INTEGRATION	<b>WL Tran ID:</b>	14
<b>Activity Name:</b>	DIRXML_EVENT1	<b>Instance ID:</b>	14
<b>Work List Name:</b>	DIRXML ADD	<b>Worked:</b>	
<b>Event Name:</b>	ADD		
<b>Assoc ID:</b>	001	Wood, Aaron S	

Below the event details, the 'Worklist Status' section shows radio buttons for 'Available' (selected), 'Selected', 'Worked', 'Cancelled', and 'Error'. To the right, the 'DirXML ID' is MDALRYMPLE, the 'DirXML DN' is YourOrganization\Users\Active\MDALRYMPLE, and the 'Action Date/Time' is 10/04/2002 11:15:15.000000PM. A 'Long Description' text area is at the bottom.

- 12 After you know that events are configured to process properly, you should ensure that you can make changes to the data and save it. Click *Use > eDirectory Subscriber*.
- 13 Click *Search* and locate the sample person you created.
- 14 Specify a sample value (or modify an existing value) in the phone number field, then click *Save*.
- 15 Click *Use > eDirectory Query > DirXMLQuery01*.
- 16 Click *Search* and double-click your sample person.
- 17 Verify that the phone number you added or modified has changed.

The screenshot shows the same 'DirXML Administrator 36EP - Use - eDirectory Work List Maint' window, but now displaying the details of a specific person. The 'eDirectory Worklist' tab is active, and the person's information is as follows:

<b>Bus Proc:</b>	DIRXML_INTEGRATION	<b>WL Tran ID:</b>	14
<b>Activity Name:</b>	DIRXML_EVENT1	<b>Instance ID:</b>	14
<b>Work List Name:</b>	DIRXML ADD	<b>Worked:</b>	
<b>Event Name:</b>	ADD		
<b>Assoc ID:</b>	001	Wood, Aaron S	

Below the event details, the person's information is displayed:

<b>Last Name:</b>	Wood
<b>First Name:</b>	Aaron
<b>Middle Name:</b>	S
<b>Address Line 1:</b>	123 Sunnyville Lane
<b>City/State/Postal:</b>	Sunshine CA 54321
<b>Date of Birth:</b>	04/12/1984
<b>Status:</b>	A
<b>Home Phone:</b>	111/222-3344

18 You can now close the PeopleSoft client.

Now that you have completed the required steps for configuring the PSA, proceed to “[Running the Message Agent Test Program](#)” on page 37.

## 4.2 Running the Message Agent Test Program

You can test the following PeopleSoft Message Agents:

- ♦ DIRXML\_TRANS01
- ♦ DIRXML\_SCHEMA01
- ♦ DIRXML\_SCHEMA01\_UPDATE
- ♦ DIRXML\_SCHEMA01\_QUERY

For detailed troubleshooting information regarding the Message Agent Test System, see [Chapter 8](#), “[Troubleshooting the Driver](#),” on page 69.

### 4.2.1 Testing the DIRXML\_TRANS01 PeopleSoft Message Agent

`Dirxml_trans01.ist` verifies that the Message Agent is receiving events from the worklist.

---

**NOTE:** Script files that have 8 after the script name are for PeopleSoft 8.1. Other IST scripts are for PeopleSoft 7.5.

---

- 1 Browse to the `C:\PSA\PSA-P75\` or `C:\PSA\PSA-P8\` directory.
- 2 Copy the appropriate test program for a message agent to the `{ps_home}\bin\client\winx86` directory.

PeopleSoft Version	Test Program
PeopleSoft 7.5	<code>psmtst32v75.exe</code>
PeopleSoft 8.1	<code>psmtst32v80.exe</code>

- 3 Create a shortcut on the desktop to the Message Agent Test program.
- 4 View the program’s properties, and validate that the `.ist` file is not set to read-only.
- 5 Double-click the test program.
- 6 Click *File > Open*.
- 7 Click the `dirxml_trans01.ist` file, then click *Open*.
- 8 Ensure that the appropriate connect string, operator ID, and password are listed for connecting to the application server.
- 9 Click *Execute*.

If the Message Agent is working properly, a list of processing key fields is populated.

The following is a sample script:

```
Connect 137.65.147.162:7000,PS,PS;  
StartMessage DIRXML_TRANS01,DIRXML_TRANS01;
```

```
ProcessMessage;
GetOutputall;
Disconnect;
```

In this script the connect string is 137.65.147.162:7000. The User ID is PS, and the password is PS.

You can now close the DIRXML\_TRANS01 window. Do not close the Message Agent Test program, because you use it in the following steps.

## 4.2.2 Testing the DIRXML\_SCHEMA01 PeopleSoft Message Agent

`Dirxml_schema.ist` verifies that all of the fields on the DIRXML\_SCHEMA01 business process definition are being passed to the Message Agent.

- 1 Double-click the message test program.
- 2 Click *File > Open*.
- 3 Click the `dirxml_schema01.ist` file, then click *Open*.
- 4 Verify that the appropriate connect string, operator ID, and password are listed.
- 5 Modify the *INSTANCEID*, *INSTSTATUS*, *ACTIVITY NAME*, and *EVENTNAME* fields.

You can find this information by searching for the person and viewing the DIRXML\_SCHEMA01 panel on the eDirectory Work List Maintenance Search panel shown below:

Date/Time	Event Name	Assoc ID
2002-09-24-16:40:27.000000	ADD	P0001

These fields must correspond to the event that you are trying to receive the output data for. Copy and paste appropriate values from the output from the previous test. This output was generated from testing `dirxml_trans01.ist`.

- 6 Click *Save*.
- 7 Click *Execute*.

If the Message Agent is working properly, all the fields that are populated in PeopleSoft are populated in the output.

The following is a sample script:

```

Connect 137.65.147.182:7000,PS,PS;
StartMessage DIRXML_SCHEMA01,DIRXML_SCHEMA01;
Setfield BUSPROCNAME=DIRXML_INTEGRATION;
Setfield ACTIVITYNAME=DIRXML_EVENT1;
Setfield EVENTNAME=ADD;
Setfield WORKLISTNAME=DIRXML_ADD;
Setfield INSTANCEID=5;
Setfield INSTSTATUS=0;
Setfield COMMENTSHORT=Selected;
ProcessMessage;
GetOutPutall;
Disconnect;

```

- 8 Close the log files.

### 4.2.3 Testing the DIRXML\_SCHEMA01\_UPDATE PeopleSoft Message Agent

`Dirxml_schema01_update.ist` tests updates to the PeopleSoft record `DIRXML_SCHEMA01_UPDATE`. This process tests events that come from eDirectory rather than events that are sent from PeopleSoft.

- 1 Double-click the message test program.
- 2 Click *File > Open*.
- 3 Click the `dirxml_schema01_update.ist` file, then click *Open*.
- 4 Verify that the appropriate connect string, operator ID, and password are listed.
- 5 Modify the *ASSOC ID*, *NDS ID*, *NDS CONTEXT*, and *Description* fields.  
These fields must correspond to the employee that you are trying to update.
- 6 Click *Save*.
- 7 Click *Execute*.

If the Message Agent is working properly, the eDirectory Subscriber page (the NDS Data Update panel for PeopleSoft 7.5) is populated with the appropriate eDirectory ID, Distinguished Name, E-mail, and Description fields for the selected employee.

The following is a sample script:

```

Connect 137.65.139.178:7000,PS,PS;
StartMessage DIRXML_SCHEMA01_UPDATE,DIRXML_SCHEMA01_UPDATE;
SetField Assoc ID=8964;
SetField NDS ID=GHORNBE;
SetField NDS CONTEXT=NCS.PRIV.NOVELL;
SetField Description=Updated;
SetField email=unknown;
ProcessMessage;
GetOutPutall;
Disconnect;

```

- 8 Close the log file.

## 4.2.4 Testing the DIRXML\_SCHEMA01\_QUERY PeopleSoft Message Agent

`DirXML_schema01_query.ist` tests the query function for the driver from PeopleSoft. This process tests the ability to retrieve data from PeopleSoft based on ASSOC ID.

The Output fields defined on the Query object need to be equivalent to those that are defined on the DIRXML\_SCHEMA01 object.

The primary difference between these two objects is the key input fields. This object only has the object ID as the key field (for example, Assoc ID).

- 1 Double-click the message test program.
- 2 Click *File > Open*.
- 3 Click the `dixml_schema01_query.ist` file, then click *Open*.
- 4 Verify that the appropriate connect string, operator ID, and password are listed.
- 5 Modify the *Assoc ID*.

This field must correspond to the employee that you are trying to access.

- 6 Click *Execute*.

If the Message Agent is working properly, all the fields that are populated in PeopleSoft are populated in the output.

The following is a sample script:

```
Connect 137.65.139.178:7000,PS,PS;
StartMessage DIRXML_SCHEMA01_QUERY,DIRXML_SCHEMA01_QUERY;
SetField Assoc ID=8964;
ProcessMessage;
GetOutPutall;
Disconnect
```

- 7 Close the log file.

Now that you have tested your PeopleSoft message agents, you should install the event server. Refer to [“Installing the Event Server” on page 40](#).

## 4.3 Installing the Event Server

Typically, the Event Server is installed when you run the installation program. If either of the following has occurred, manually install or configure the desktop shortcut for the Event Server:

- ♦ You did not install the Event Server when you installed other components.
- ♦ Any directory paths were unavailable during the installation.

### 4.3.1 Installing the Event Server

The following steps assume that eDirectory has been installed on drive C: of the server in the `\novell\remoteloader\` directory.

- 1 Copy the following to the `c:\novell\nds` directory.
  - ♦ `NPS75EventServer.exe` for PeopleSoft 7.5



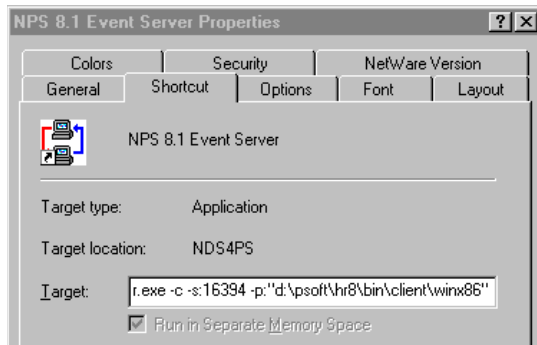
- ♦ NPS81EventServer.exe for PeopleSoft 8.1

## 2 Create a shortcut for the Event Server application on the desktop.

**IMPORTANT:** If you are running multiple PeopleSoft drivers, create one shortcut for each Event Server.

## 3 Edit the command line in the *Target* field.

The following graphic illustrates this field:



Right-click the shortcut, then click *Properties* > *Shortcut*.

Enter parameters for the shortcut. In the following table, {PS\_Home} is the root of the PeopleSoft directory structure.

PeopleSoft Platform	Parameters
PeopleSoft 7.5	C:\Novell\NDS\NPS75EventServer.exe -c -s:16386 -p:PS_Home\bin\client\winx86
PeopleSoft 8.1	C:\Novell\NDS\NPS81EventServer.exe -c -s:16394 -p:PS_Home\bin\client\winx86

The -c parameter configures the Event Server so that it runs as a console application. When you use this parameter, configure the DOS window properties with the following:

- ♦ A scroll bar
- ♦ A different color layout

The Event Server communicates with the driver using a Windows\* sockets interface or listening port. The following table lists the suggested socket number for each Event Server. If these numbers conflict with sockets already being used in your organization, use a different port set.

Event Server	Port
PeopleSoft 7.5 Event Server	16386
PeopleSoft 8.1 Event Server	16394

Because each Event Server requires four consecutive sockets or ports, socket numbers for multiple Event Servers must be at least four digits apart.

The Event Server must establish communications with the PeopleSoft Message Agent. It does this by connecting to the PeopleSoft Message Agent DLL files.

These DLL files are typically located in the `PS_HOME\BIN\CLIENT\WINX86` directory.

For the NPS75EventServer shortcut, an example path to the PeopleSoft Message DLLs is – `p:d:\psoft\hr757\bin\client\winx86`. In this example, `PS_HOME` is equal to `d:\psoft\hr757`.

- 4 Type the following in the Start In field:

`c:\Novell\NDS`

This is the directory where eDirectory is installed. The following figure illustrates this field:



- 5 Select *Apply*, then click *OK*.
- 6 Open the application via the shortcut.

You can customize the scroll bar and color changes by customizing the Property tab on the window.

## Stopping the Event Server

You might need to stop the Event Server to refresh PeopleSoft connectivity, recycle the server, or debug. When you stop the Event Server, the Event Server sends a stop command to the driver shim. This command shuts down the driver shim before the Event Server exits.

To stop the Event Server from a console window:

- 1 Select the Event Server window.
- 2 Enter Q.

---

**IMPORTANT:** You can stop the Event Server by clicking the *Exit* icon in the window. However, do this only if the driver shim is not running. Before clicking the icon, shut down the driver shim. Do this from the DirXML\* Driver Set Properties menu.

---

## Installing the Event Server as a Service

You can install the Event Server as a Windows NT or Windows 2000 service. Normally, the Event Server is installed during the driver installation.

Two additional shortcuts are placed on the desktop during the installation process. These shortcuts enable you to create or remove the Event Server as a service.

To install the Event Server as a service:

- 1 Ensure that the Event Server program is available.
  - ♦ On a local machine, verify that `NPSxEventServer.exe` is saved locally to the machine.
  - ♦ If you are installing the Event Server on a remote server where eDirectory is not installed, copy `dirxmllib.dll` and `expat.dll` from the `novell\nds` directory to the directory that the Event Server will run from.

---

**NOTE:** The PeopleSoft client must always be installed on the server where the Event Server is going to run.

---

The remote server's IP address is used when configuring the Event Server in the driver parameters.

- 2 At the command prompt, go to the directory where the executable file is stored, then enter the following:

```
NPS75EventServer.exe -i -s:16386 -p: PS_HOME\bin\client\winx86
```

The socket number and the path to the PeopleSoft Psmmsg.dll might be different, based on your environment and driver configuration.

- 3 From the Start menu, click *Settings > Control Panel > Services > DirXML Event Server for PS 7.5* (or the appropriate release).
- 4 Click *Startup* > select the Automatic option button to automatically start the Event Server at login, then click *Start*.

To stop the service:

- 1 From the Start menu, click *Settings > Control Panel > Services*.
- 2 Select the service, then click *Stop*.

### Manually Uninstalling the Event Server As a Service

You can uninstall or remove the Event Server as a service. Follow the same steps as for installing the Event Server. However, in Step 2, substitute the -u parameter for the -i parameter.

For example, enter the following:

```
NPS75EventServer.exe -u -s:16386 -p: PS_HOME\bin\client\winx86
```

---

**NOTE:** When the Event Server runs as a service, there is no console window. All debug information is only available in the DSTRACE log.

---



# Activating the Driver

# 5

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

To activate the driver, see “**Activating Novell Identity Manager Products**” in the *Identity Manager 3.5 Installation Guide*.



# Managing the Driver

# 6

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

- ♦ [Section 6.1, “Starting, Stopping, or Restarting the Driver,” on page 47](#)
- ♦ [Section 6.2, “Migrating and Resynchronizing Data,” on page 48](#)
- ♦ [Section 6.3, “Using the DirXML Command Line Utility,” on page 48](#)
- ♦ [Section 6.4, “Viewing Driver Versioning Information,” on page 49](#)
- ♦ [Section 6.5, “Reassociating a Driver Set Object with a Server Object,” on page 53](#)
- ♦ [Section 6.6, “Changing the Driver Configuration,” on page 53](#)
- ♦ [Section 6.7, “Storing Driver Passwords Securely with Named Passwords,” on page 54](#)
- ♦ [Section 6.8, “Adding a Driver Heartbeat,” on page 60](#)

## 6.1 Starting, Stopping, or Restarting the Driver

- ♦ [Section 6.1.1, “Starting the Driver in Designer,” on page 47](#)
- ♦ [Section 6.1.2, “Starting the Driver in iManager,” on page 47](#)
- ♦ [Section 6.1.3, “Stopping the Driver in Designer,” on page 47](#)
- ♦ [Section 6.1.4, “Stopping the Driver in iManager,” on page 47](#)
- ♦ [Section 6.1.5, “Restarting the Driver in Designer,” on page 48](#)
- ♦ [Section 6.1.6, “Restarting the Driver in iManager,” on page 48](#)

### 6.1.1 Starting the Driver in Designer

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

### 6.1.2 Starting the Driver in iManager

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

### 6.1.3 Stopping the Driver in Designer

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

### 6.1.4 Stopping the Driver in iManager

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

- 2 Browse to the driver set where the driver exists, then click *Search*.
- 3 Click the upper right corner of the driver icon, then click *Stop driver*.

### 6.1.5 Restarting the Driver in Designer

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

### 6.1.6 Restarting the Driver in iManager

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

## 6.2 Migrating and Resynchronizing Data

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

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

To use one of the options explained above:

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

For more information, see [Chapter 7, “Synchronizing Objects,” on page 63](#).

## 6.3 Using the DirXML Command Line Utility

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



For example, you could create a shell script on Linux to check the status of the driver. See [Appendix A, “DirXML Command Line Utility,” on page 85](#) for detailed information about the DirXML Command Line utility. For daily tasks, use iManager or Designer.

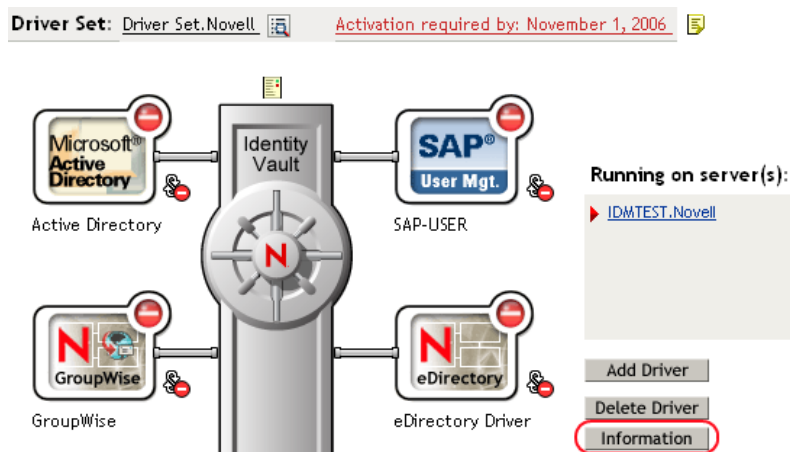
## 6.4 Viewing Driver Versioning Information

The Versioning Discovery tool only exists in iManager.

- [Section 6.4.1, “Viewing a Hierarchical Display of Versioning Information,” on page 49](#)
- [Section 6.4.2, “Viewing the Versioning Information As a Text File,” on page 50](#)
- [Section 6.4.3, “Saving Versioning Information,” on page 52](#)

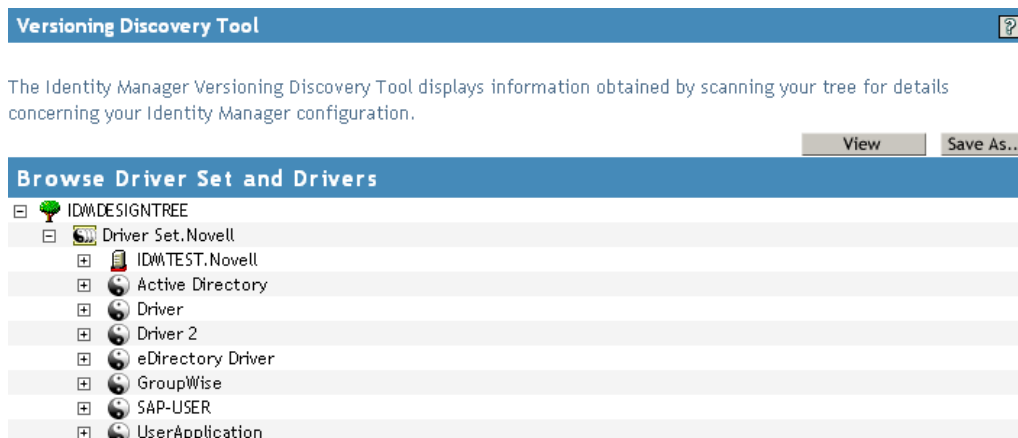
### 6.4.1 Viewing a Hierarchical Display of Versioning Information

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



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

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



The unexpanded hierarchical view displays the following:

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

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

- ♦ Drivers

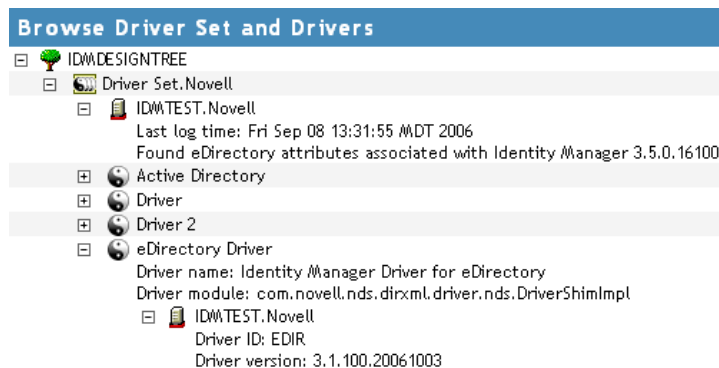
4 View versioning information related to servers by expanding the server icon.



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

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

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



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

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

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

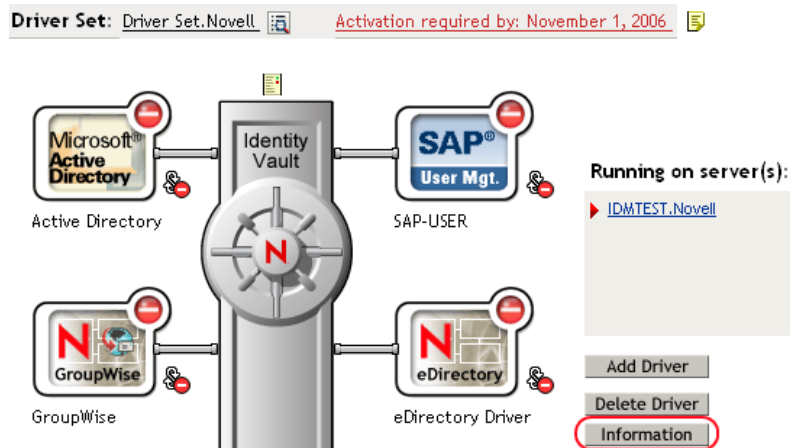
- ♦ The driver ID
- ♦ The version of the instance of the driver running on that server

## 6.4.2 Viewing the Versioning Information As a Text File

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

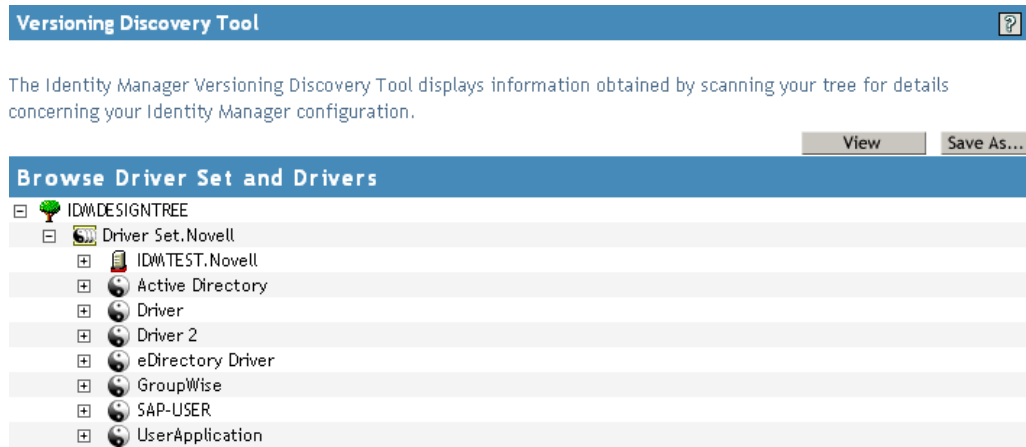
- 1 To find your Driver Set object in iManager, click *Identity Manager > Identity Manager Overview*, then click *Search*.

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



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

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



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

## Versioning Discovery Tool - Report Viewer

```

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

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

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

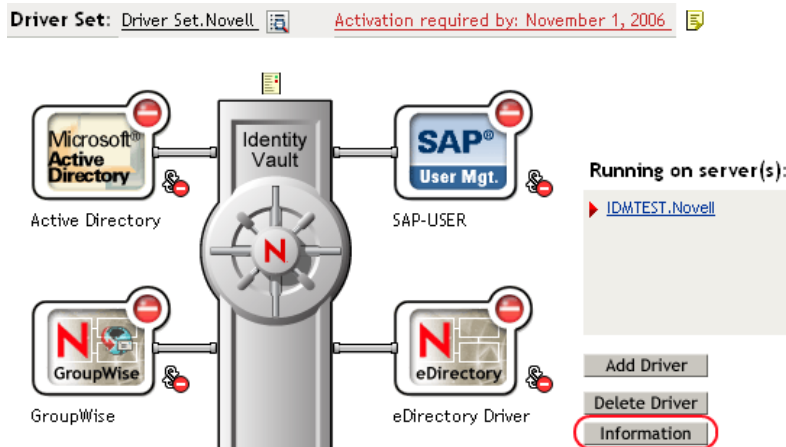
Driver Set:  Driver Set.Novell
  Driver Set running on Identity Vault:  IDMTTEST.Novell
    Last log time:  Fri Sep 08 13:31:55 MDT 2006
    Found eDirectory attributes associated with Identity Manager 3.5.0.1
  Driver:  Active Directory.Driver Set.Novell
    Driver name:  Identity Manager Driver for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

OK

### 6.4.3 Saving Versioning Information

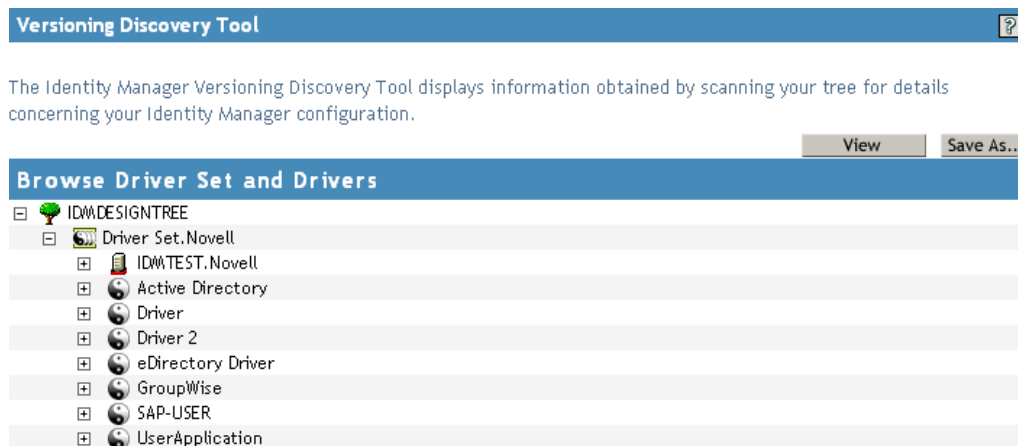
You can save versioning information to a text file on your local or network drive.

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



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

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



- 4 In the File Download dialog box, click *Save*.
- 5 Navigate to the desired directory, type a filename, then click *Save*.  
Identity Manager saves the data to a text file.

## 6.5 Reassociating a Driver Set Object with a Server Object

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

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

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

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

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

## 6.6 Changing the Driver Configuration

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

To change the driver configuration in iManager:

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

To change the driver configuration in Designer:

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

For a listing of all of the configuration fields, see [Appendix B, “Properties of the Driver,” on page 99](#).

## 6.7 Storing Driver Passwords Securely with Named Passwords

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

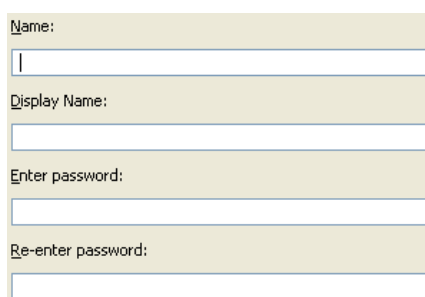
You can also use the Named Passwords feature to store other pieces of information securely, such as a user name.

To use a named password in a driver policy, you refer to it by the name of the password, instead of using the actual password, and the Metadirectory engine sends the password to the driver. The method described in this section for storing and retrieving named passwords can be used with any driver without making changes to the driver shim.

- ♦ [Section 6.7.1, “Using Designer to Configure Named Passwords,” on page 54](#)
- ♦ [Section 6.7.2, “Using iManager to Configure Named Passwords,” on page 55](#)
- ♦ [Section 6.7.3, “Using Named Passwords in Driver Policies,” on page 56](#)
- ♦ [Section 6.7.4, “Using the DirXML Command Line Utility to Configure Named Passwords,” on page 57](#)

### 6.7.1 Using Designer to Configure Named Passwords

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



- 3 Specify the *Name* of the named password.

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

## 6.7.2 Using iManager to Configure Named Passwords

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

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

The screenshot shows the 'Identity Manager' configuration window with the 'Named Passwords' tab selected. The 'Named Passwords' tab is highlighted in the top navigation bar. Below the navigation bar, there is a description: 'Named Passwords lets you securely store multiple passwords for a driver. Instead of including a password in clear text in a driver policy, you can configure the policy to request a Named Password.' Below this description are 'Add' and 'Remove' buttons. A blue header bar reads 'Named Passwords'. Below this, it says 'For server: IDMTTEST.Novell'. There are two entries in the list, each with a checkbox and a text field: 'smtp admin' and 'workflow admin'. At the bottom of the window are 'OK', 'Cancel', and 'Apply' buttons.

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

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

Add Remove

**Named Passwords**


For server: IDMTTEST.Novell

☐ smtp admin

☐ workflow admin

OK Cancel Apply

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

 **Named Password**

---

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

Name:

Display name:

Enter password:

Reenter password:

---

- 5 Specify a name, display name and a password, then click *OK* twice.  
You can use this feature to store other kinds of information securely, such as a username.
- 6 Click *OK* to restart the driver and have the changes take effect.
- 7 To remove a Named Password, select the password name, then click *Remove*.  
The password is removed without prompting you to confirm the action.

### 6.7.3 Using Named Passwords in Driver Policies

- ♦ “Using the Policy Builder” on page 56
- ♦ “Using XSLT” on page 57

#### Using the Policy Builder

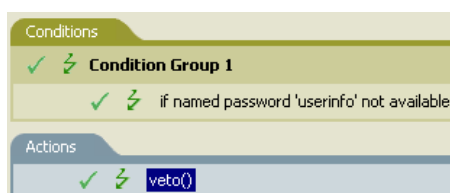
Policy Builder allows you to make a call to a named password. Create a new rule and select Named Password as the condition, then set an action depending upon if the Named Password is available or not available.

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

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



**Figure 6-1** A Policy Using Named Passwords



## Using XSLT

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

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

## 6.7.4 Using the DirXML Command Line Utility to Configure Named Passwords

- “Creating a Named Password in the DirXML Command Line Utility” on page 57
- “Using the DirXML Command Line Utility to Remove a Named Password” on page 58

### Creating a Named Password in the DirXML Command Line Utility

- 1 Run the DirXML Command Line utility.

For information, see [Appendix A, “DirXML Command Line Utility,”](#) on page 85.

- 2 Enter your username and password.

The following list of options appears.

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

- 3 Enter 3 for driver operations.

A numbered list of drivers appears.

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

The following list of options appears.

```
Select a driver operation for:  
driver_name  
1: Start driver  
2: Stop driver
```

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

- 5** Enter 13 for password operations.

The following list of options appears.

Select a password operation

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

- 6** Enter 5 to set a new named password.

The following prompt appears:

Enter password name:

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

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

Enter password:

The characters you type for the password are not displayed.

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

Confirm password:

- 10** After you enter and confirm the password, you are returned to the password operations menu.
- 11** After completing this procedure, you can use the 99 option twice to exit the menu and quit the DirXML Command Line Utility.

### Using the DirXML Command Line Utility to Remove a Named Password

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

- 1** Run the DirXML Command Line utility.

For information, see [Appendix A, “DirXML Command Line Utility,” on page 85](#).

- 2** Enter your username and password.

The following list of options appears.

DirXML commands

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

Enter choice:

**3** Enter 3 for driver operations.

A numbered list of drivers appears.

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

The following list of options appears.

Select a driver operation for:

*driver\_name*

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

Enter choice:

**5** Enter 13 for password operations.

The following list of options appears.

Select a password operation

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

Enter choice:

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

The list of existing named passwords is displayed.

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

**7** Enter 6 to remove one or more named passwords.

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

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

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

Enter password name:

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

Select a password operation

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

Enter choice:

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

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

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

## 6.8 Adding a Driver Heartbeat

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

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

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

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

To configure the heartbeat:

- 1** In iManager, click *Identity Manager > Identity Manager Overview*.
- 2** Browse to and select your driver set object, then click *Search*.

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

**4** On the Identity Manager tab, click *Driver Configuration*, scroll to *Driver Parameters*, then look for Heart Beat or a similar display name.

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

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

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

**5** If a driver parameter does not exist for heartbeat, click *Edit XML*.

**6** Add a driver parameter entry like the following example, as a child of <publisher-options>. (For an AD driver, make it a child of <driver-options>.)

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

---

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

---

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

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



# Synchronizing Objects

# 7

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

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

- ♦ [Section 7.1, “What Is Synchronization?” on page 63](#)
- ♦ [Section 7.2, “When Is Synchronization Done?” on page 63](#)
- ♦ [Section 7.3, “How Does the Metadirectory Engine Decide Which Object to Synchronize?” on page 64](#)
- ♦ [Section 7.4, “How Does Synchronization Work?” on page 65](#)

## 7.1 What Is Synchronization?

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

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

## 7.2 When Is Synchronization Done?

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

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

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

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

- ♦ The user issues a manual driver synchronization request. This corresponds to the *Resync* button in the Driver Set property page in ConsoleOne®, or to the *Synchronize* button on the iManager Identity Manager Driver Overview page.
- ♦ The Metadirectory engine encounters an error with the driver's cache and cannot recover from the cache error. The driver's cache is deleted and the engine generates object synchronization commands as detailed in [Section 7.3, "How Does the Metadirectory Engine Decide Which Object to Synchronize?," on page 64](#).

## 7.3 How Does the Metadirectory Engine Decide Which Object to Synchronize?

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

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

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

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

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

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



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

## 7.4 How Does Synchronization Work?

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

1. Each system (the Identity Vault and the connected system) is queried for all attribute values in the appropriate filters.
  - ♦ eDirectory is queried for all values in the Subscriber filter, and for values that are marked for synchronization in Identity Manager 2.x and Identity Manager 3.x.
  - ♦ The connected system is queried for all values in the Publisher filter, and for values that are marked for synchronization in Identity Manager 2.x and Identity Manager 3.x.
2. The returned attribute values are compared and modification lists are prepared for the Identity Vault and the connected system according to [Table 7-1 on page 66](#), [Table 7-2 on page 67](#), and [Table 7-3 on page 68](#).

In the tables the following pseudo-equations are used:

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

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

- ♦ [Section 7.4.1, “Scenario One,” on page 65](#)
- ♦ [Section 7.4.2, “Scenario Two,” on page 67](#)
- ♦ [Section 7.4.3, “Scenario Three,” on page 68](#)

### 7.4.1 Scenario One

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

**Figure 7-1** *Scenario One*

Class Name: User

Attribute Name: Facsimile Telephone Num

Publish

☒ Synchronize  
☐ Ignore  
☐ Notify  
☐ Reset

Subscribe

☒ Synchronize  
☐ Ignore  
☐ Notify  
☐ Reset

Merge Authority

☒ Default  
☐ Identity Vault  
☐ Application  
☐ None

Optimize modifications to Identity Vault

☒ Yes  
☐ No

The following table contains the values that the Metadirectory engine synchronizes when the attribute is sent through a filter that is set to the configuration for Scenario One. The table shows different outputs depending upon whether the attribute comes from the Identity Vault or the Application, if the attribute is single-valued or multi-valued, and if the attribute is empty or non-empty.

**Table 7-1** *Output of Scenario One*

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

## 7.4.2 Scenario Two

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

**Figure 7-2** *Scenario Two*

Class Name: User

Attribute Name: Description

Publish

☐ Synchronize  
☒ Ignore  
☐ Notify  
☐ Reset

Subscribe

☒ Synchronize  
☐ Ignore  
☐ Notify  
☐ Reset

Merge Authority

☐ Default  
☒ Identity Vault  
☐ Application  
☐ None

Optimize modifications to Identity Vault

☒ Yes  
☐ No

The following table contains the values that the Metadirectory engine synchronizes when the attribute is sent through a filter that is set to the configuration for Scenario Two. The table shows different outputs depending upon whether the attribute comes from the Identity Vault or the Application, if the attribute is single-valued or multi-valued, and if the attribute is empty or non-empty.

**Table 7-2** *Output of Scenario Two*

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

### 7.4.3 Scenario Three

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

**Figure 7-3** Scenario Three

Class Name: User

Attribute Name: DirXML-ADAliasName

Publish

☒ Synchronize  
☐ Ignore  
☐ Notify  
☐ Reset

Subscribe

☐ Synchronize  
☒ Ignore  
☐ Notify  
☐ Reset

Merge Authority

☐ Default  
☐ Identity Vault  
☒ Application  
☐ None

Optimize modifications to Identity Vault

☒ Yes  
☐ No

The following table contains the values that the Metadirectory engine synchronizes when the attribute is sent through a filter that is set to the configuration for Scenario Three. The table shows different outputs depending upon whether the attribute comes from the Identity Vault or the Application, if the attribute is single-valued or multi-valued, and if the attribute is empty or non-empty.

**Table 7-3** Output of Scenario Three

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

# Troubleshooting the Driver

# 8

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

## 8.1 Resolving Errors

The Resolving Errors section is broken into they types of error codes you may encounter and how to resolve those errors.

- ◆ [Section 8.1.1, “The Event Server Does Not Load,” on page 69](#)
- ◆ [Section 8.1.2, “The Driver Does Not Start,” on page 69](#)
- ◆ [Section 8.1.3, “The Driver Is Not Communicating with the Event Server,” on page 69](#)
- ◆ [Section 8.1.4, “The Event Server Receives Message Agent Errors,” on page 70](#)
- ◆ [Section 8.1.5, “Attributes Do Not Get Refreshed on the Data Map Object,” on page 70](#)
- ◆ [Section 8.1.6, “Driver Only Appears to Process Transactions,” on page 70](#)
- ◆ [Section 8.1.7, “Data Does Not Show up in the Identity Vault on the Publisher Channel,” on page 71](#)
- ◆ [Section 8.1.8, “Data Does Not Update in PeopleSoft on the Subscriber Channel,” on page 71](#)
- ◆ [Section 8.1.9, “No Transactions Are Coming Across the Publisher Channel,” on page 71](#)
- ◆ [Section 8.1.10, “Transactions Do Not Get Placed in the PeopleSoft Queue,” on page 71](#)
- ◆ [Section 8.1.11, “No Data Is Returned When Running the Message Test Program,” on page 71](#)
- ◆ [Section 8.1.12, “Transactions Are Left in Selected State and Not Processed,” on page 72](#)
- ◆ [Section 8.1.13, “Receiving Errors on the Publisher Channel When Processing a Transaction,” on page 72](#)
- ◆ [Section 8.1.14, “Message Agent Relationships Are Not Functioning,” on page 72](#)

### 8.1.1 The Event Server Does Not Load

- ◆ Check the path environment variable to assure that the PeopleSoft client and the Identity Manager directory are in its path.

### 8.1.2 The Driver Does Not Start

- ◆ Check that the Event Server is running.
- ◆ Check that the `npsshim.dll` file is accessible.
- ◆ Check that the connection parameters are set correctly.

### 8.1.3 The Driver Is Not Communicating with the Event Server

- ◆ Verify that the Event Server was started first.
- ◆ Verify that the Event Server and driver do not have multiple copies running.

- ♦ Verify that the connection was not previously broken.
- ♦ You might need to restart Novell® eDirectory™ and then restart the driver components.

### 8.1.4 The Event Server Receives Message Agent Errors

- ♦ Verify that the APIs are configured appropriately. (Use the Message Agent Test Program.)
- ♦ Verify that the Application Server is running.
- ♦ Verify connectivity to the Message Agent.
- ♦ Makes sure you have the correct PeopleSoft Application Server Connection String:  
You can use either: HOSTNAME:PORT or IP:PORT for single instance of Application Server  
or  
Use Fail Over Mode: //HOSTNAME:PORT, //HOSTNAME:PORT where the first entry is the first Application Server and the second entry is the second Application Server to which the driver tries to connect.
- ♦ Make sure the names of the Message Agent APIs are spelled correctly. These names are case sensitive.
- ♦ Check for errors when updating the Worklist transaction.
- ♦ Check for errors on the Application Server.
- ♦ Check for errors on the SMTP gateway.
- ♦ Verify password usage of the driver. This might be case sensitive depending on the platform you use.

### 8.1.5 Attributes Do Not Get Refreshed on the Data Map Object

- ♦ Verify that the Message Agent APIs are working correctly.
- ♦ Verify that both the worklist and data update APIs are working correctly (DIRXML\_SCHEMA01 and DIRXML\_SCHEMA01\_UPDATE).

### 8.1.6 Driver Only Appears to Process Transactions

Transactions appear to be read from the worklist, but nothing gets processed by the driver. Messages repeat themselves. The following text output might appear on the Event Server console:

```
Record Count: 5
Processing record 1 of 5.
Processing record 2 of 5.
Processing record 3 of 5.
Processing record 4 of 5.
Processing record 5 of 5.
```

However, no data is being processed. This is a result of reading transactions out of the worklist, but being unable to retrieve the data from the transaction. Make sure that you can read the data from the default DIRXML\_SCHEMA01 Message definition. You should also:

- ♦ Verify that you can read a particular transaction with the Message Test Program.
- ♦ Verify that you only get one record when accessing the same transaction online.

- ♦ Verify that the *ACTIONDTTM* field has been removed as an input key field on the API.

### **8.1.7 Data Does Not Show up in the Identity Vault on the Publisher Channel**

- ♦ Verify that the Mapping policy and filters are configured correctly.
- ♦ Verify that the APIs are working correctly and data is being produced by them.

### **8.1.8 Data Does Not Update in PeopleSoft on the Subscriber Channel**

- ♦ Verify that the Mapping policy and filters are configured correctly.
- ♦ Verify that the APIs are working correctly.

### **8.1.9 No Transactions Are Coming Across the Publisher Channel**

- ♦ Verify that the eDirectory Default Synchronization check box is selected.
- ♦ Verify that there are active transactions in the queue ready for processing.
- ♦ Ensure that driver parameters are pointing to the correct PeopleSoft database. For example, transactions do not process if they are in the PROD database, but the driver is still pointing to the test database (which is configured to run with the driver, but holds no transactions).

### **8.1.10 Transactions Do Not Get Placed in the PeopleSoft Queue**

- ♦ Verify that PeopleCode is working properly.
- ♦ Verify that you are not in correction mode when performing updates.

### **8.1.11 No Data Is Returned When Running the Message Test Program**

No data is returned, particularly when you are running the Message Test Program. You might see errors such as “No Text Available.”

There are two typical reasons for this error:

- ♦ The Key Input elements are not pointing to the Search Record on the Field Mapping definition of the Activity.
- ♦ The field elements point to an invalid reference.

Make sure the Key Input elements are associated with the Search Record entry on the field mapping definition and not directly with the table. Also, make sure that the field elements are mapped to an appropriate field record definition that exists in the application buffer. You should also ensure that the data elements exist on a page within the application and that the links are configured correctly.

## 8.1.12 Transactions Are Left in Selected State and Not Processed

- ♦ Verify that all of the Message Activities can be processed and that the states can be updated to a 2 (worked) or 4 (error).

If e-mail is configured in PeopleSoft and the SMTP gateway is down, an error can occur causing the update of the transaction list to fail. You should verify that all online processing of the application works correctly. PeopleCode attached to the update might sometimes fail, thus causing the transaction to fail. If system connectivity is lost, the database or application server goes down during processing and causes the driver to abandon the transaction. The transaction is left in the state of “selected” with a status of 1.

## 8.1.13 Receiving Errors on the Publisher Channel When Processing a Transaction

**Table 8-1** Sample errors and possible solutions

Sample Errors	Possible Solutions
Operation vetoed by Create policy	Possible required data missing in the Create policy or other criteria in the Create policy has an error.
generateKeyPair: -216 DSERR_PASSWORD_TOO_SHORT	The attribute used for the initial password does not comply to policy, but the User object will still be created.
Unable to read current state of 8101	No association exists for this identity.
nameToID: -601 ERR_NO_SUCH_ENTRY	Possible Placement policy error with an invalid container object designated.
No DN generated by Placement policy	Possible missing or invalid data causing no valid DN to be created.

## 8.1.14 Message Agent Relationships Are Not Functioning

If data does not show up in the attributes, or isn’t getting posted into PeopleSoft, or data is missing, you should begin looking at the message agent relationships. You should:

- ♦ Verify that the API is getting the data from the PeopleSoft buffer. (Use the Message Agent Test program and follow all four steps in the driver process to test all of the APIs.)

For the delivered Message Agent Activities, the following procedure represents the process flow of testing the Message Agent Scripts with the Message Agent Test Program. The purpose of this test is to completely validate and walk through the process that the driver uses to interface with PeopleSoft and retrieve transactions and update the PeopleSoft environment. Each of the applications that these Message Activities are connected with should be tested online through the use of PeopleTools prior to testing the Message Agent scripts.

- 1 Ensure that you only have one active/available transaction in the worklist ready for processing.
- 2 Execute the `dirxml_trans01.ist` script to retrieve the transaction in the worklist.



- 3 Verify that key elements of the transaction are returned and that they are unique.
- 4 Execute the second script, DIRXML\_SCHEMA01, replacing the key element values with the values retrieved from Step 2 above from the execution of dirxml\_trans01.ist script. Also, change the INSTSTATUS to 1 in order to represent updating the PeopleSoft transaction to a selected state.
- 5 Validate that all of the data elements appear as desired.

If elements are missing, there is probably an issue with the data being retrieved on one of the pages in the application. You should check the ...DATA1X, ...DATA2X, and ...DATA3X pages in the associated application. Sometimes, even though the data shows up on the page, it doesn't appear in the results when running the script because there is a problem in the related display definition.

Another reason why the data might not appear is that the process direction is not set to output on the Field Mapping within the Activity definition.

- 6 Repeat steps 4 and 5 by executing the script again for the same transaction while changing the status to 2 (worked) and 4 (error).

The driver, when processing a transaction, always sets the transaction to a status of 1 (selected). When the transaction is processed it either sets the transaction to 2 (worked) or 4 (error). If an error occurs, the comment field is also updated. Be sure to test all possible updates and verify online that the update occurred as desired.

- 7 Execute the DIRXML\_SCHEMA01\_UPDATE script for a particular object. The default would be for a particular employee by specifying the EMPLID of the employee. The NDS ID and NDS CONTEXT fields are updated in PeopleSoft on the DIRXML\_TRANS01 record when the driver complete processing of a transaction retrieved on the publisher channel. If this fails, the driver cannot complete the update of the status that occurs in the prior steps. The update of these fields is for documentation purposes only or to make them available elsewhere in PeopleSoft, or for using in an e-mail generated by a PeopleSoft workflow, etc. This functionality can be turned off by using an override parameter defined on the properties parameter page in the driver configuration.

---

**NOTE:** You should execute the script multiple times by replacing the contents of the Description and the Email ID fields and any additional fields that might have been added to the DIRXML\_TRANS01 table and this definition that are set to be updated by the driver on the Subscriber channel, such as from the Identity Vault to PeopleSoft.

---

- 8 Execute the DIRXML\_SCHEMA01\_QUERY script to validate that the data elements for a particular object are accessible.

These elements should all be set to output except for the key input value. The elements listed here on the field mapping should be consistent with those listed on the DIRXML\_SCHEMA01 definition. The only difference should be the key input fields. On this Activity, there is normally only one key input field for the object being accessed. This would equate to EMPLID for the delivered configuration against the HR database.

When all of the Message Agent Test Scripts have been tested completely with validation of all processes that the driver is configured to do, there should be no issues regarding the driver accessing PeopleSoft through the Message Agent. Typical other problems include:

- ♦ Connectivity IP address and port for the application server
- ♦ ID and password
- ♦ Correct naming of all activities in the parameters for the driver.

Remember that there are three basic test phases:

1. Test all of the processes manually online using the PeopleSoft applications as configured.
2. Test all of the processes using the four test scripts with the Message Agent Test program.
3. Test the driver connecting to the Activities through the Message Agent.

## 8.2 Troubleshooting Driver Processes

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

### 8.2.1 Viewing Driver Processes

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

- ♦ “Adding Trace Levels in Designer” on page 74
- ♦ “Adding Trace Levels in iManager” on page 76
- ♦ “Capturing Driver Processes to a File” on page 76

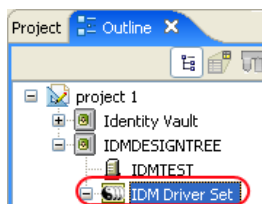
#### Adding Trace Levels in Designer

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

- ♦ “Driver Set” on page 74
- ♦ “Driver” on page 75

#### Driver Set

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



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

Parameter	Description
Driver trace level	As the driver object trace level increases, the amount of information displayed in DSTRACE increases.  Trace level 1 shows errors, but not the cause of the errors. If you want to see password synchronization information, set the trace level to 5.

Parameter	Description
XSL trace level	DSTRACE displays XSL events. Set this trace level only when troubleshooting XSL style sheets. If you do not want to see XSL information, set the level to zero.
Java debug port	Allows developers to attach a Java* debugger.
Java trace file	When a value is set in this field, all Java information for the driver set object is written to a file. The value for this field is the path for that file.  As long as the file is specified, Java information is written to this file. If you do not need to debug Java, leave this field blank.
Trace file size limit	Allows you to set a limit for the Java trace file. If you set the file size to <i>Unlimited</i> , the file grows in size until there is no disk space left.

If you set the trace level on the driver set object, all drivers appear in the DSTRACE logs.

## Driver

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

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

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

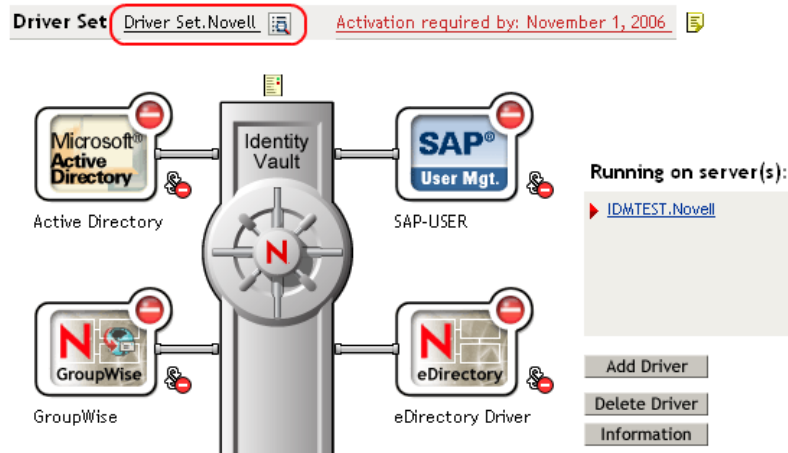
## Adding Trace Levels in iManager

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

- ♦ “Driver Set” on page 76
- ♦ “Driver” on page 76

### Driver Set

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



- 4 Select the *Misc* tab for the driver set object.
  - 5 Set the parameters for tracing, then click *OK*.
- See “Misc” on page 110 for the parameters.

### Driver

- 1 In iManager, select *Identity Manager > Identity Manager Overview*.
  - 2 Browse to the driver set object where the driver object resides, then click *Search*.
  - 3 Click the upper right corner of the driver object, then click *Edit properties*.
  - 4 Select the *Misc* tab for the driver object.
  - 5 Set the parameters for tracing, then click *OK*.
- See “Misc” on page 110 for the parameters.

---

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

---

## Capturing Driver Processes to a File

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

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

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

- ♦ “NetWare” on page 77
- ♦ “Windows” on page 77
- ♦ “UNIX” on page 78
- ♦ “iMonitor” on page 78
- ♦ “Remote Loader” on page 79

## NetWare

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

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

## Windows

- 1 Open the *Control Panel* > *NDS Services* > `dstrace.dlm`, then click *Start* to display the NDS Server Trace utility window.
- 2 Click *Edit* > *Options*, then click *Clear All* to clear all of the default flags.
- 3 Select *DirXML* and *DirXML Drivers*.
- 4 Click OK.
- 5 Click *File* > *New*.
- 6 Specify the filename and location where you want the DSTRACE information saved, then click *Open*.

- 7 Wait for the event to occur.
- 8 Click *File > Close*.  
This stops the information from being written to the log file.
- 9 Open the file in a text editor and search for the event or the object you modified.

## UNIX

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

## iMonitor

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

- ♦ `ndsimon.nlm` runs on NetWare®.
- ♦ `ndsimon.dlm` runs on Windows.
- ♦ `ndsimonitor` runs on UNIX\*.

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

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

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

## Remote Loader

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

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

You can also enable tracing from the command line by using the following switches. For more information, see “[Configuring the Remote Loader](#)” in the *Novell Identity Manager 3.5 Administration Guide*.

**Table 8-2** Command Line Tracing Switches

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

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



# Backing Up the Driver

# 9

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

---

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

---

Not all server-specific information stored on the driver is contained in the XML file. Make sure this information is documented through the Doc Gen process in Designer. See “[Documenting Projects](#)” in the *Designer 2.0 for Identity Manager 3.5*.

- ♦ [Section 9.1, “Exporting the Driver in Designer,” on page 81](#)
- ♦ [Section 9.2, “Exporting the Driver in iManager,” on page 81](#)

## 9.1 Exporting the Driver in Designer

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

## 9.2 Exporting the Driver in iManager

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



# Security: Best Practices

# 10

In order to secure the driver and the information it is synchronizing, see “[Security: Best Practices](#)” in the *Novell Identity Manager 3.5 Administration Guide*.



# DirXML Command Line Utility

# A

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

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

eDirectory 8.7.x

- ♦ Windows: \Novell\Nds\dxcmd.bat
- ♦ NetWare<sup>®</sup>: sys:\system\dxcmd.ncf
- ♦ UNIX: /usr/bin/dxcmd

eDirectory 8.8.x

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

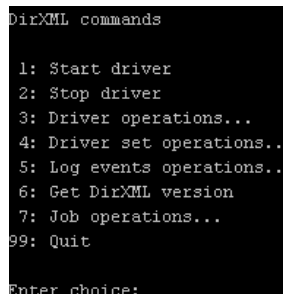
There are two different methods for using the DirXML Command Line utility:

- ♦ [Section A.1, “Interactive Mode,” on page 85](#)
- ♦ [Section A.2, “Command Line Mode,” on page 94](#)

## A.1 Interactive Mode

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

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



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

- 4 Enter the number of the command you want to perform.  
[Table A-1 on page 86](#) contains the list of options and what functionality is available.
- 5 Enter 99 to quit the utility.

---

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

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

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

---

**Table A-1** *Interactive Mode Options*

Option	Description
1: <i>Start Driver</i>	Starts the driver. If there is more than one driver, each driver is listed with a number. Enter the number of the driver to start the driver.
2: <i>Stop Driver</i>	Stops the driver. If there is more than one driver, each driver is listed with a number. Enter the number of the driver to stop the driver.
3: <i>Driver operations</i>	Lists the operations available for the driver. If there is more than one driver, each driver is listed with a number. Enter the number of the driver to see the operations available. See <a href="#">Table A-2 on page 87</a> for a list of operations.
4: <i>Driver set operations</i>	Lists the operations available for the driver set. <ul style="list-style-type: none"><li>♦ 1: Associate driver set with server</li><li>♦ 2: Disassociate driver set from server</li><li>♦ 99: Exit</li></ul>
5: <i>Log events operations</i>	Lists the operations available for logging events through Novell® Audit. See <a href="#">Table A-5 on page 92</a> for a description of these options.
6: <i>Get DirXML version</i>	Lists the version of the Identity Manager installed.
7: <i>Job operations</i>	Manages jobs created for Identity Manager.1
99: <i>Quit</i>	Exits the DirXML Command Line utility

---

**Figure A-1** *Driver Options*

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

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

Enter choice:
```

**Table A-2** *Driver Options*

Options	Description
1: <i>Start driver</i>	Starts the driver.
2: <i>Stop driver</i>	Stops the driver.
3: <i>Get driver state</i>	Lists the state of the driver. <ul style="list-style-type: none"> <li>♦ 0 - Driver is stopped</li> <li>♦ 1 - Driver is starting</li> <li>♦ 2 - Driver is running</li> <li>♦ 3 - Driver is stopping</li> </ul>
4: <i>Get driver start option</i>	Lists the current driver start option. <ul style="list-style-type: none"> <li>♦ 1 - Disabled</li> <li>♦ 2 - Manual</li> <li>♦ 3 - Auto</li> </ul>
5: <i>Set driver start option</i>	Changes the start option of the driver. <ul style="list-style-type: none"> <li>♦ 1 - Disabled</li> <li>♦ 2 - Manual</li> <li>♦ 3 - Auto</li> <li>♦ 99 - Exit</li> </ul>

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



Options	Description
10: <i>Queue event for driver</i>	<p>Adds and event to the driver queue</p> <p><i>Enter filename of XDS event document:</i></p> <p>Examples:</p> <p>NetWare: <code>sys:\files\add.xml</code></p> <p>Windows: <code>c:\files\add.xml</code></p> <p>Linux: <code>/files/add.xml</code></p>
11: <i>Check object password</i>	<p>Validates that an object's password in the connected system is associated with a driver. It matches the object's eDirectory password (Distribution Password, used with Universal Password).</p> <p><i>Enter user name:</i></p>
12: <i>Initialize new driver object</i>	<p>Performs an internal initialization of data on a new Driver object. This is only for testing purposes.</p>
13: <i>Password operations</i>	<p>There are nine Password options. See <a href="#">Table A-3 on page 89</a> for a description of these options.</p>
14: <i>Cache operations</i>	<p>There are five Cache operations. See <a href="#">Table A-4 on page 91</a> for a descriptions of these options.</p>
99: <i>Exit</i>	<p>Exits the driver options.</p>

**Figure A-2** Password Operations

```

Select a password operation

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

Enter choice:

```

**Table A-3** Password Operations

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

Operation	Description
3: <i>Set Remote Loader password</i>	<p>The Remote Loader password is used to control access to the Remote Loader instance.</p> <p>Enter the Remote Loader password, then confirm the password by typing it again.</p>
4: <i>Clear Remote Loader password</i>	<p>Clears the Remote Loader password so no Remote Loader password is set on the Driver object.</p>
5: <i>Set named password</i>	<p>Allows you to store a password or other pieces of security information on the driver. See <a href="#">Section 6.7, "Storing Driver Passwords Securely with Named Passwords,"</a> on page 54 for more information.</p> <p>There are four prompts to fill in:</p> <ul style="list-style-type: none"> <li>♦ <i>Enter password name:</i></li> <li>♦ <i>Enter password description:</i></li> <li>♦ <i>Enter password:</i></li> <li>♦ <i>Confirm password:</i></li> </ul>
6: <i>Clear named passwords</i>	<p>Clears a specified named password or all named passwords that are stored on the driver object: <i>Do you want to clear all named passwords? (yes/no).</i></p> <p>If you enter Yes, all Named Passwords are cleared. If you enter No, you are prompted to specify the password name that you want to clear.</p>
7: <i>List named passwords</i>	<p>Lists all named passwords that are stored on the driver object. It lists the password name and the password description.</p>
8: <i>Get password state</i>	<p>Lists if a password is set for:</p> <ul style="list-style-type: none"> <li>♦ Driver Object password</li> <li>♦ Application password</li> <li>♦ Remote loader password</li> </ul> <p>The dxcmnd utility allows you to set the Application password and the Remote Loader password. You cannot set the Driver Object password with this utility. It shows if the password has been set or not.</p>
99: <i>Exit</i>	<p>Exits the current menu and takes you back to the Driver options.</p>

**Figure A-3** *Cache Operations*

```
Enter choice: 14

Select a cache operation

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

Enter choice:
```

**Table A-4** *Cache Operations*

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

**Figure A-4** *Log Event Operations*

```
Select a log events operation

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

Enter choice:
```

**Table A-5** *Log Events Operations*

Operation	Description
1: <i>Set driver set log events</i>	Allows you to log driver set events through Novell Audit. There are 49 items you can select to log. See <a href="#">Table A-6 on page 92</a> for a list of these options.  Type the number of the item you want to log. After the items are selected, enter 99 to accept the selections.
2: <i>Reset driver set log events</i>	Resets all of the log event options.
3: <i>Set driver log events</i>	Allows you to log driver events through Novell Audit. There are 49 items to select to log. See <a href="#">Table A-6 on page 92</a> for a list of these options.  Type the number of the item you want to log. After the items are selected, enter 99 to accept the selections.
4: <i>Reset driver log events</i>	Resets all of the log event options.
99: <i>Exit</i>	Exits the log events operations menu.

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

Options
1: Status success
2: Status retry
3: Status warning
4: Status error
5: Status fatal
6: Status other
7: Query elements
8: Add elements
9: Remove elements
10: Modify elements
11: Rename elements
12: Move elements
13: Add-association elements
14: Remove-association elements
15: Query-schema elements
16: Check-password elements

---

**Options**

---

- 17: Check-object-password elements
  - 18: Modify-password elements
  - 19: Sync elements
  - 20: Pre-transformed XDS document from shim
  - 21: Post input transformation XDS document
  - 22: Post output transformation XDS document
  - 23: Post event transformation XDS document
  - 24: Post placement transformation XDS document
  - 25: Post create transformation XDS document
  - 26: Post mapping transformation <inbound> XDS document
  - 27: Post mapping transformation <outbound> XDS document
  - 28: Post matching transformation XDS document
  - 29: Post command transformation XDS document
  - 30: Post-filtered XDS document <Publisher>
  - 31: User agent XDS command document
  - 32: Driver resync request
  - 33: Driver migrate from application
  - 34: Driver start
  - 35: Driver stop
  - 36: Password sync
  - 37: Password request
  - 38: Engine error
  - 39: Engine warning
  - 40: Add attribute
  - 41: Clear attribute
  - 42: Add value
  - 43: Remove value
  - 44: Merge entire
  - 45: Get named password
  - 46: Reset Attributes
  - 47: Add Value - Add Entry
  - 48: Set SSO Credential
-

---

**Options**

---

49: Clear SSO Credential

50: Set SSO Passphrase

51: User defined IDs

99: Accept checked items

---

**Table A-7** *Enter Table Title Here*

Options	Description
1: <i>Get available job definitions</i>	Allows you to select an existing job.  <i>Enter the job number:</i>  <i>Do you want to filter the job definitions by containment? Enter Yes or No</i>  <i>Enter name of the file for response:</i>  Examples:  NetWare: sys:\files\user.log  Windows: c:\files\user.log  Linux: /files/user.log
2: <i>Operations on specific job object</i>	Allows you to perform operations for a specific job.

---

## A.2 Command Line Mode

The command line mode allows you to use script or batch files. [Table A-8 on page 94](#) lists the different options that are available.

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

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

This example command starts the driver.

**Table A-8** *Command Line Options*

Option	Description
<b>Configuration</b>	
-user <user name>	Specify the name of a user with administrative rights to the drivers you want to test.
-host <name or IP address>	Specify the IP address of the server where the driver is installed.
-password <user password>	Specify the password of the user specified above.

---

Option	Description
-port <port number>	Specify a port number, if the default port is not used.
-q <quiet mode>	Displays very little information when a command is executed.
-v <verbose mode>	Displays detailed information when a command is executed.
-s <stdout>	Writes the results of the <code>dxcmd</code> command to <code>stdout</code> .
-? <show this message>	Displays the help menu.
-help <show this message>	Displays the help menu.
<b>Actions</b>	
-start <driver dn>	Starts the driver.
-stop <driver dn>	Stops the driver.
-getstate <driver dn>	Shows the state of the driver as running or stopped.
-getstartoption <driver dn>	Shows the startup option of the driver.
-setstartoption <driver dn> <disabled manual auto> <resync noresync>	Sets how the driver starts if the server is rebooted. Sets whether the objects are to be resynchronized when the driver restarts.
-getcachelimit <driver dn>	Lists the cache limit set for the driver.
-setcachelimit <driver dn> <0 or positive integer>	Sets the cache limit for the driver.
-migrateapp <driver dn> <filename>	Processes an XML document that contains a query command.  Create the XML document that contains a query command by using the Novell <code>nds.dtd</code> ( <a href="http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_dtd/data/dtdndsoverview.html#dtdndsoverview">http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_dtd/data/dtdndsoverview.html#dtdndsoverview</a> ).
-setshimpassword <driver dn> <password>	Sets the application password. This is the password of the user account you are using to authenticate into the connected system with.
-clearshimpassword <driver dn> <password>	Clears the application password.
-setremoteloaderpassword <driver dn> <password>	Sets the Remote Loader password.  The Remote Loader password is used to control access to the Remote Loader instance.
<clearremoteloaderpassword <driver dn>	Clears the Remote Loader password.

Option	Description
-sendcommand <driver dn> <input filename> <output filename>	<p>Processes an XDS command document.</p> <p>Specify the XDS command document as the input file.</p> <p>Examples:</p> <p>NetWare: sys:\files\user.xml</p> <p>Windows: c:\files\user.xml</p> <p>Linux: /files/user.log</p> <p>Specify the output filename to see the results.</p> <p>Examples:</p> <p>NetWare: sys:\files\user.log</p> <p>Windows: c:\files\user.log</p> <p>Linux: /files/user.log</p>
-sendevent <driver dn> <input filename>	Submits a document to the driver's Subscriber channel, bypassing the driver cache. The document is processed ahead of anything that might be in the cache at the time of the submission. It also means that the submission fails if the driver is not running.
-queueevent <driver dn> <input filename>	Submits a document to the driver's Subscriber channel by queuing the document in the driver cache. The document gets processed after anything that might be in the cache at the time of the submission. The submission won't fail if the driver isn't running.
-setlogevents <dn> <integer ...>	Sets Novell Audit log events on the driver. The integer is the option of the item to log. See <a href="#">Table A-6 on page 92</a> for the list of the integers to enter.
-clearlogevents <dn>	Clears all Novell Audit log events that are set on the driver.
-setdriverset <driver set dn>	Associates a driver set with the server.
-cleardriverset	Clears the driver set association from the server.
-getversion	Shows the version of Identity Manager that is installed.
-initdriver object <dn>	Performs an internal initialization of data on a new Driver object. This is only for testing purposes.
-setnamedpassword <driver dn> <name> <password> [description]	Sets named passwords on the driver object. You specify the name, the password, and the description of the named password.
-clearnamedpassword <driver dn> <name>	Clears a specified named password.
-startjob <job dn>	Starts the specified job.



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

If a command line is executed successfully, it returns a zero. If the command line returns anything other than zero, it is an error. For example 0 means success, and -641 means invalid operation. -641 is an eDirectory error code. [Table A-9 on page 97](#) contains other values for specific command line options.

**Table A-9** *Command Line Option Values*


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

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

# Properties of the Driver

# B

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

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

- ◆ [Section B.1, “Driver Configuration,” on page 99](#)
- ◆ [Section B.2, “Global Configuration Values,” on page 104](#)
- ◆ [Section B.3, “Named Passwords,” on page 105](#)
- ◆ [Section B.4, “Engine Control Values,” on page 106](#)
- ◆ [Section B.5, “Log Level,” on page 107](#)
- ◆ [Section B.6, “Driver Image,” on page 108](#)
- ◆ [Section B.7, “Security Equals,” on page 109](#)
- ◆ [Section B.8, “Filter,” on page 109](#)
- ◆ [Section B.9, “Edit Filter XML,” on page 109](#)
- ◆ [Section B.10, “Misc,” on page 110](#)
- ◆ [Section B.11, “Excluded Users,” on page 110](#)
- ◆ [Section B.12, “Driver Manifest,” on page 111](#)
- ◆ [Section B.13, “Inspector,” on page 111](#)
- ◆ [Section B.14, “Server Variables,” on page 111](#)

## B.1 Driver Configuration

In iManager:

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

In Designer:

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

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

### B.1.1 Driver Module



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

In iManager:

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

In Designer:

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

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

## B.1.2 Driver Object Password

In iManager:

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

In Designer:

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

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

### B.1.3 Authentication







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





In iManager:

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

In Designer:

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

Option	Description
<i>Authentication ID</i> or  <i>User ID</i>	Specify a user application ID. This ID is used to pass Identity Vault subscription information to the application.  Example: Administrator
<i>Authentication Context</i> or  <i>Connection Information</i>	Specify the IP address or name of the server the application shim should communicate with.
<i>Remote Loader Connection Parameters</i> or  <i>Host name</i>  <i>Port</i>  <i>KMO</i>  <i>Other parameters</i>	Used only if the driver is connecting to the application through the remote loader. The parameter to enter is hostname=xxx.xxx.xxx.xxx port=xxxx kmo=certificatename, when the host name is the IP address of the application server running the Remote Loader server and the port is the port the remote loader is listening on. The default port for the Remote Loader is 8090.  The kmo entry is optional. It is only used when there is an SSL connection between the Remote Loader and the Metadirectory engine.  Example: hostname=10.0.0.1 port=8090 kmo=IDMCertificate

Option	Description
<i>Driver Cache Limit (kilobytes)</i>	Specify the maximum event cache file size (in KB). If it is set to zero, the file size is unlimited.
or	 Click <i>Unlimited</i> to set the file size to unlimited in Designer.
 <i>Cache limit (KB)</i>	
<i>Application Password</i>	Specify the password for the user object listed in the <i>Authentication ID</i> field.
or	
 <i>Set Password</i>	
<i>Remote Loader Password</i>	Used only if the driver is connecting to the application through the Remote Loader. The password is used to control access to the Remote Loader instance. It must be the same password specified during the configuration of the Remote Loader on the connected system.
or	
 <i>Set Password</i>	

## B.1.4 Startup Option


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

In iManager:

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

In Designer:

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

Option	Description
<i>Auto start</i>	The driver starts every time the Identity Manager server is started.
<i>Manual</i>	The driver does not start when the Identity Manager server is started. The driver must be started through Designer or iManager.
<i>Disabled</i>	The driver has a cache file that stores all of the events. When the driver is set to 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.
 <i>Do not automatically synchronize the driver</i>	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.

## B.1.5 Driver Parameters

Below is a listing of the driver parameters that are available to the Identity Manager 3.7 driver for PeopleSoft.

In iManager:

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

In Designer:

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

Parameter	Description
<b>Driver Settings</b>	
Driver parameters for server: Server-name	Specifies the name of the server whose driver parameters you want to modify.
<i>Enter Server Host Name</i>	Specifies the host name or IP address of the Event Server that connects to PeopleSoft.
<i>Enter Server Port</i>	Specifies the port address of the Event Server that connects to PeopleSoft. The default is <i>16500</i> .
<i>Data Record ID Field</i>	Specifies the name of the data record definition that uniquely identifies a PeopleSoft object. The default is <i>Assoc ID</i> .
<i>eDirectory Object Classname</i>	The eDirectory object class to which the Data Record definition maps. The default is <i>User</i> .
<b>Subscriber Setting</b>	
<i>Schema Data Subscriber Activity</i>	Specifies the activity definition that is used to subscribe data modifications into PeopleSoft. The default is <i>DIRXML_SCHEMA01_UPDATE</i> .
<i>Schema Data Query Activity</i>	Specifies the query activity definition that is used to subscribe data modifications into PeopleSoft. The default is <i>DIRXML_SCHEMA01_QUERY</i> .
<b>Publisher Settings</b>	
<i>Schema Data Publisher Activity</i>	Specifies the activity definition that is used to obtain Data records that are published from PeopleSoft. The default is <i>DIRXML_SCHEMA01</i> .
<i>Transaction Access Activity</i>	Specifies the transaction activity definition that is used to obtain Data records that are published from PeopleSoft. The default is <i>DIRXML_TRANS01</i> .
<i>Queue Poll Interval (seconds)</i>	Specifies the number of seconds between checks for available transactions to process. Default is <i>5</i> .

Parameter	Description
<i>Queue Retrieval Limit</i>	Specifies the maximum number of Transaction records the driver retrieves at one time. The default is 5.

## B.2 Global Configuration Values

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

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

In iManager:

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

In Designer:

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

For *Password Configuration*, you should only edit the first two settings listed here. The others are GCVs regarding Password Synchronization that are common to all drivers. They should be edited using iManager in *Passwords > Password Synchronization*, not here. Some of them have dependencies on each other that are represented only in the iManager interface. They are explained in “[Password Synchronization across Connected Systems](#)” in the *Novell Identity Manager 3.5 Administration Guide*.

**Table B-1** *Global Configuration Values > Password Configuration*

Option	Description
<i>Identity Manager accepts passwords from application</i>	If <i>True</i> , allows passwords to flow from the connected system to Identity Manager.
<i>Publish passwords to NDS password</i>	Use the password from the connected system to set the non-reversible NDS <sup>®</sup> password in eDirectory.
<i>Publish passwords to Distribution Password</i>	Use the password from the connected system to set the NMAS <sup>™</sup> Distribution Password used for Identity Manager password synchronization.



Option	Description
<i>Require password policy validation before publishing passwords</i>	If <i>True</i> , applies NMAS password policies during publish password operations. The password is not written to the data store if it does not comply.
<i>Notify the user of password synchronization failure via e-mail</i>	If <i>True</i> , notify the user by e-mail of any Password Synchronization failures.
<i>Connected System or Driver Name</i>	The name of the connected system, application, or Identity Manager driver. This value is used by the e-mail notification templates.
<i>Password Failure Notification User</i>	Password synchronization policies are configured to send e-mail notifications to the associated user when password updates fail. To send a notification copy, specify the DN of that user. Otherwise, leave this field blank.
<i>Inactive Users Container</i>	The name of the Organizational Unit object where the inactive users will be placed. Specify the DN of the OU object. Otherwise, leave this field blank.
<i>Active Users Container</i>	The name of the Organizational Unit object where Active users will be placed. Specify the DN of the OU object. Otherwise, leave this field blank.
<i>Active Employees Group</i>	The name of the Group object where Active Employee users will be added. Specify the DN of the object. Otherwise, leave this field blank.
<i>Active Managers Group</i>	The name of the Group object where Active Manager users will be added. Specify the DN of the object. Otherwise, leave this field blank.

## B.3 Named Passwords

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

You can also use the Named Passwords feature to store other pieces of information securely, such as a user name. To configured Named Passwords, see [Section 6.7, “Storing Driver Passwords Securely with Named Passwords,” on page 54](#).

In iManager:

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

In Designer:

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

## B.4 Engine Control Values

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

In iManager:

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

This option does not exist in Designer at this time.

**Table B-2** Engine Control Values

Option	Description
<i>Subscriber channel retry interval in seconds</i>	The Subscriber channel retry interval controls how frequently the Metadirectory engine retries the processing of a cached transaction after the application shim's Subscriber object returns a retry status.
<i>Qualified form for DN-syntax attribute values</i>	The qualified specification for DN-syntax attribute values controls whether values for DN-syntax attribute values are presented in unqualified slash form or qualified slash form. A True setting means the values are presented in qualified form.
<i>Qualified form from rename events</i>	The qualified form for rename events controls whether the new-name portion of rename events coming from the Identity Vault are presented to the Subscriber channel with type qualifiers. For example, CN=. A True setting means the names are presented in qualified form.
<i>Maximum eDirectory replication wait time in seconds</i>	The maximum eDirectory™ replication wait time controls the maximum time that the Metadirectory engine waits for a particular change to replicate between the local replica and a remote replica. This only affects operations where the Metadirectory engine is required to contact a remote eDirectory server in the same tree to perform an operation and might need to wait until some change has replicated to or from the remote server before the operation can be completed (for example, object moves when the Identity Manager server does not hold the master replica of the moved object; file system rights operations for Users created from a template.)
<i>Use non-compliant backwards-compatible mode for XSLT</i>	<p>This control sets the XSLT processor used by the Metadirectory engine to a backwards-compatible mode. The backwards-compatible mode causes the XSLT processor to use one or more behaviors that are not XPath 1.0 and XSLT 1.0 standards-compliant. This is done in the interest of backwards-compatibility with existing DirXML® style sheets that depend on the non-standard behaviors.</p> <p>For example, the behavior of the XPath “!=” operator when one operand is a node-set and the other operand is other than a node-set is incorrect in DirXML releases up to and including Identity Manager 2.0. This behavior has been corrected; however, the corrected behavior is disabled by default through this control in favor of backwards-compatibility with existing DirXML style sheets.</p>

Option	Description
<i>Maximum application objects to migrate at once</i>	<p>This control is used to limit the number of application objects that the Metadirectory engine requests from an application during a single query that is performed as part of a Migrate Objects from Application operation.</p> <p>If java.lang.OutOfMemoryError errors are encountered during a Migrate from Application operation, this number should be set lower than the default. The default is 50.</p> <hr/> <p><b>NOTE:</b> This control does not limit the number of application objects that can be migrated; it merely limits the batch size.</p> <hr/>
<i>Set creatorsName on objects created in Identity Vault</i>	<p>This control is used by the Identity Manager engine to determine if the creatorsName attribute should be set to the DN of this driver on all objects created in the Identity Vault by this driver.</p> <p>Setting the creatorsName attribute allows for easily identifying objects created by this driver, but also carries a performance penalty. If not set, the creatorsName attribute defaults to the DN of the NCP™ Server object that is hosting the driver.</p>
<i>Write pending associations</i>	<p>This control determines whether the Identity Manager engine writes a pending association on an object during Subscriber channel processing.</p> <p>Writing a pending association confers little or no benefit but does incur a performance penalty. Nevertheless, the option exists to turn it on for backward compatibility.</p>
<i>Use password event values</i>	<p>This control determines the source of the value reported for the nspmDistributionPassword attribute for Subscriber channel Add and Modify events.</p> <p>Setting the control to False means that the current value of the nspmDistributionPassword is obtained and reported as the value of the attribute event. This means that only the current password value is available. This is the default behavior.</p> <p>Setting the control to True means that the value recorded with the eDirectory event is decrypted and is reported as the value of the attribute event. This means that both the old password value (if it exists) and the replacement password value at the time of the event are available. This is useful for synchronizing passwords to certain applications that require the old password to enable setting a new password.</p>
<i>Enable password synchronization status reporting</i>	<p>This control determines whether the Identity Manager engine reports the status of Subscriber channel password change events.</p> <p>Reporting the status of Subscriber channel password change events allows applications such as the Identity Manager User Application to monitor the synchronization progress of a password change that should be synchronized to the connected application.</p> <hr/>

## B.5 Log Level

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


Novell® recommends that you use Novell Audit instead of setting the log levels. See “[Integrating Identity Manager with Novell Audit](#)” in the *Identity Manager 3.5 Logging and Reporting*.

In iManager:

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

In Designer:

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

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

## B.6 Driver Image

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

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

---

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

---

In iManager:

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

In Designer:

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

## B.7 Security Equals

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

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

In iManager:

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

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

## B.8 Filter

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

In iManager:

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

The filter editor is accessed through the outline view in Designer.

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

## B.9 Edit Filter XML

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

In iManager:

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

You can edit the Filter in XML through the Filter Editor.

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

## B.10 Misc


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

In iManager:

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

In Designer:

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

Option	Description
<i>Trace level</i>	Increases the amount of information displayed in DSTRACE. Trace level 1 shows errors, but not the cause of the errors. If you want to see password synchronization information, set the trace level to 5.
<i>Trace file</i>	<p>When a value is set in this field, all Java information for the driver is written to the file. The value for this field is the path for that file.</p> <p>As long as the file is specified, Java information is written to this file. If you do not need to debug Java, leave this field blank.</p>
<i>Trace file size limit</i>	Allows you to set a limit for the Java trace file. If you set the file size to Unlimited, the file grows in size until there is no disk space left.
<i>Trace name</i>	Driver trace messages are prepended with the value entered in this field.
 <i>Use setting from Driver Set</i>	This option is only available in Designer. It allows the driver to use the same setting that is set on the Driver Set object.

## B.11 Excluded Users

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

In iManager:

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

Designer does not list the excluded users.

## B.12 Driver Manifest

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

In iManager:

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

In Designer:

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

## B.13 Inspector

The Inspector displays information about the connected system without directly accessing the system. Designer does not have this option.

## B.14 Server Variables

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

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

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

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

To change these settings in iManager:

- 1 In iManager, select *Passwords > Password Policies*.

2 Select a password policy, then click *Edit*.

3 Select *Universal Password*.

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

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

---

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

---

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



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

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