

# Novell Identity Manager Driver for WorkOrder

3.5

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

IMPLEMENTATION GUIDE

July 30, 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. See the [Novell International Trade Services Web page \(http://www.novell.com/info/exports/\)](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 © 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 on the [Novell Legal Patents Web page \(http://www.novell.com/company/legal/patents/\)](http://www.novell.com/company/legal/patents/) and one or more additional patents or pending patent applications in the U.S. and in other countries.

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

*Online Documentation:* To access the latest online documentation for this and other Novell products, see [the Novell Documentation Web page \(http://www.novell.com/documentation\)](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 Overview</b>	<b>11</b>
1.1 . . . . . The Role of the WorkOrder Driver	11
1.2 Terminology . . . . .	12
1.3 Key Functions . . . . .	12
1.3.1 Local Platforms . . . . .	12
1.3.2 Remote Platforms . . . . .	12
1.3.3 Role-Based Entitlements . . . . .	12
1.3.4 Password Synchronization Support . . . . .	12
1.3.5 Synchronized Objects . . . . .	12
1.4 WorkOrder Driver Features . . . . .	13
1.5 The Subscriber and Publisher Channels . . . . .	13
<b>2 Planning</b>	<b>15</b>
2.1 Planning Issues for All Configurations . . . . .	15
2.2 Designing the Driver . . . . .	16
2.3 Prerequisites . . . . .	16
<b>3 Understanding Driver Architecture</b>	<b>17</b>
3.1 Base Configuration . . . . .	17
3.2 Subscriber Channel Functions . . . . .	17
3.3 Publisher Channel Functions . . . . .	19
3.3.1 The Publisher Channel Wakes Up . . . . .	19
3.3.2 How the Publisher Channel Processes Work Orders . . . . .	20
3.3.3 How the Publisher Channel Deletes Work Orders . . . . .	22
<b>4 Objects and Attributes Used</b>	<b>23</b>
4.1 New Objects Used by the Driver . . . . .	23
4.1.1 DirXML-WorkOrder Object . . . . .	23
4.1.2 DirXML-WorkToDo Object . . . . .	23
4.2 DoltNow and SendToPublisher Flags . . . . .	23
4.2.1 DoltNow Flag . . . . .	24
4.2.2 SendToPublisherFlag . . . . .	24
<b>5 Customizing the Driver</b>	<b>25</b>
5.1 Default Driver Parameters . . . . .	25
5.2 Customizing the Driver . . . . .	25
5.2.1 Customizing Policies . . . . .	25
5.2.2 Customizing Driver Parameters . . . . .	25
5.3 Additional Solutions . . . . .	25
5.3.1 Human Resource Sample Using an HR Driver . . . . .	26
5.3.2 Human Resource Sample without an HR Driver . . . . .	28

<b>6</b>	<b>Installing and Configuring</b>	<b>31</b>
6.1	Prerequisites . . . . .	31
6.2	Files Included with the WorkOrder Driver . . . . .	31
6.3	Creating Containers . . . . .	31
6.4	Installation . . . . .	32
6.5	Configuration Information . . . . .	32
6.6	Importing the Driver Configuration . . . . .	32
6.7	Configuring the Driver in Designer . . . . .	33
6.8	Configuring the Driver in iManager . . . . .	33
6.9	Configuring the Subscriber and Publisher Channels . . . . .	35
6.9.1	Configuring the Subscriber Channel . . . . .	35
6.9.2	Configuring the Publisher Channel . . . . .	36
<b>7</b>	<b>Creating and Managing Work Orders</b>	<b>37</b>
7.1	Work Orders Created from Events . . . . .	37
7.2	Using the iManager Plug-in . . . . .	37
7.2.1	Creating and Deleting a New Work Order . . . . .	37
7.2.2	Specifying and Editing Work Order Properties . . . . .	38
<b>8</b>	<b>Managing the Driver</b>	<b>39</b>
8.1	Starting, Stopping, or Restarting the Driver . . . . .	39
8.1.1	Starting the Driver in Designer . . . . .	39
8.1.2	Starting the Driver in iManager . . . . .	39
8.1.3	Stopping the Driver in Designer . . . . .	39
8.1.4	Stopping the Driver in iManager . . . . .	39
8.1.5	Restarting the Driver in Designer . . . . .	40
8.1.6	Restarting the Driver in iManager . . . . .	40
8.2	Using the DirXML Command Line Utility . . . . .	40
8.3	Viewing Driver Versioning Information . . . . .	40
8.3.1	Viewing a Hierarchical Display of Versioning Information . . . . .	40
8.3.2	Viewing the Versioning Information As a Text File . . . . .	42
8.3.3	Saving Versioning Information . . . . .	44
8.4	Reassociating a Driver Set Object with a Server Object . . . . .	45
8.5	Changing the Driver Configuration . . . . .	46
8.6	Storing Driver Passwords Securely with Named Passwords . . . . .	46
8.6.1	Using Designer to Configure Named Passwords . . . . .	47
8.6.2	Using iManager to Configure Named Passwords . . . . .	47
8.6.3	Using Named Passwords in Driver Policies . . . . .	49
8.6.4	Using the DirXML Command Line Utility to Configure Named Passwords . . . . .	49
8.7	Adding a Driver Heartbeat . . . . .	53
<b>9</b>	<b>Troubleshooting Driver Processes</b>	<b>55</b>
9.1	Viewing Driver Processes . . . . .	55
9.1.1	Adding Trace Levels in Designer . . . . .	55
9.1.2	Adding Trace Levels in iManager . . . . .	57
9.1.3	Capturing Driver Processes to a File . . . . .	57
<b>10</b>	<b>Backing Up the Driver</b>	<b>63</b>
10.1	Exporting the Driver in Designer . . . . .	63
10.2	Exporting the Driver in iManager . . . . .	63

<b>11 Security: Best Practices</b>	<b>65</b>
<b>A DirXML Command Line Utility</b>	<b>67</b>
A.1 Interactive Mode . . . . .	67
A.2 Command Line Mode . . . . .	76
<b>B Schema For Work Order Management</b>	<b>81</b>
B.1 DirXML-WorkOrder Object . . . . .	81
B.2 DirXML-WorkToDo Object . . . . .	83
B.3 Publisher Placement Rule . . . . .	84
B.4 Subscriber Placement Rule . . . . .	84
B.5 Subscriber Create Rule . . . . .	84
<b>C Properties of the Driver</b>	<b>85</b>
C.1 Driver Configuration . . . . .	85
C.1.1 Driver Module . . . . .	85
C.1.2 Driver Object Password . . . . .	86
C.1.3 Authentication . . . . .	87
C.1.4 Startup Option . . . . .	88
C.1.5 Driver Parameters . . . . .	88
C.2 Global Configuration Values . . . . .	89
C.3 Named Passwords . . . . .	90
C.4 Engine Control Values . . . . .	90
C.5 Log Level . . . . .	92
C.6 Driver Image . . . . .	93
C.7 Security Equals . . . . .	93
C.8 Filter . . . . .	94
C.9 Edit Filter XML . . . . .	94
C.10 Misc . . . . .	94
C.11 Excluded Users . . . . .	95
C.12 Driver Manifest . . . . .	95
C.13 Inspector . . . . .	96
C.14 Server Variables . . . . .	96
<b>D Documentation Update</b>	<b>99</b>
D.1 July 30, 2007 . . . . .	99
D.1.1 Schema for Work Order Management . . . . .	99
D.2 June 29, 2007 . . . . .	99
D.2.1 Understanding Driver Architecture . . . . .	99
D.3 May 17, 2007 . . . . .	99
D.3.1 Schema for Work Order Management . . . . .	99
D.4 April 11, 2007 . . . . .	100
D.4.1 Additional Solutions . . . . .	100





# About This Guide

This guide explains how to install and configure the Novell® Identity Manager WorkOrder driver 1.0.

This guide contains the following sections:

- ♦ Chapter 1, “Overview,” on page 11
- ♦ Chapter 2, “Planning,” on page 15
- ♦ Chapter 3, “Understanding Driver Architecture,” on page 17
- ♦ Chapter 4, “Objects and Attributes Used,” on page 23
- ♦ Chapter 5, “Customizing the Driver,” on page 25
- ♦ Chapter 6, “Installing and Configuring,” on page 31
- ♦ Chapter 7, “Creating and Managing Work Orders,” on page 37
- ♦ Chapter 8, “Managing the Driver,” on page 39
- ♦ Chapter 9, “Troubleshooting Driver Processes,” on page 55
- ♦ Chapter 10, “Backing Up the Driver,” on page 63
- ♦ Chapter 11, “Security: Best Practices,” on page 65
- ♦ Appendix A, “DirXML Command Line Utility,” on page 67
- ♦ Appendix B, “Schema For Work Order Management,” on page 81
- ♦ Appendix C, “Properties of the Driver,” on page 85

## Audience

This guide is intended for developers and administrators using Identity Manager and the WorkOrder driver.

## 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 [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 the *Identity Manager Work Order Guide*, visit the [Identity Manager Documentation Web site \(http://www.novell.com/documentation/idm35drivers\)](http://www.novell.com/documentation/idm35drivers).

## Additional Documentation

For documentation on other Identity Manager drivers, see the [Identity Manager Documentation Web site \(http://www.novell.com/documentation/idm35drivers\)](http://www.novell.com/documentation/idm35drivers).

## **Documentation Conventions**

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

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

This section provides an overview of the Novell® Identity Manager 3.5 WorkOrder driver.

- ♦ [Section 1.1, “The Role of the WorkOrder Driver,” on page 11](#)
- ♦ [Section 1.2, “Terminology,” on page 12](#)
- ♦ [Section 1.3, “Key Functions,” on page 12](#)
- ♦ [Section 1.4, “WorkOrder Driver Features,” on page 13](#)
- ♦ [Section 1.5, “The Subscriber and Publisher Channels,” on page 13](#)

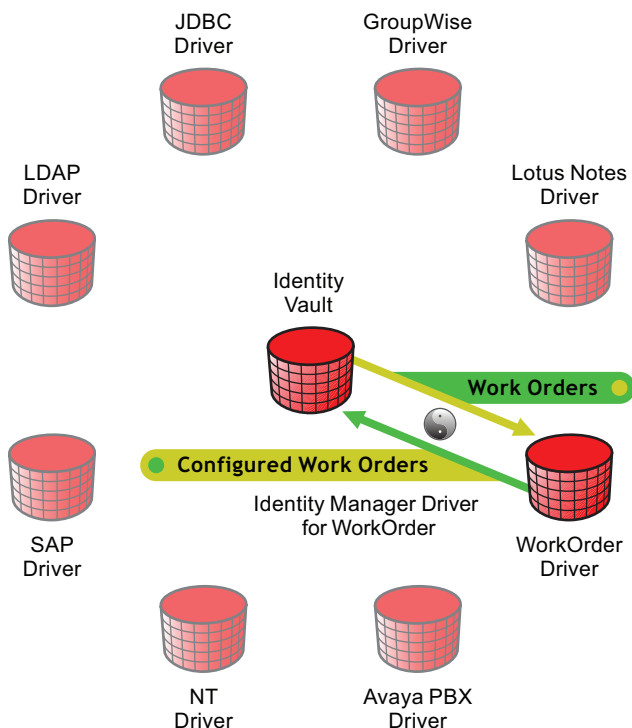
## 1.1 The Role of the WorkOrder Driver

The purpose of the Identity Manager WorkOrder driver is to allow work to be scheduled. The driver uses information from the work order container to see which WorkOrder objects are pending and due. Work orders that are pending and due are processed by the driver. The definition of the work to do is contained in the work order. For example, the work to do can be modifying an object attribute in the Identity Vault or sending an e-mail to the administrator. Policies in the WorkOrder driver perform the work that is defined in the work order.

The WorkOrder driver publishes the WorkToDo object to publish the work order.

The following diagram illustrates how the WorkOrder driver connects to the Identity Vault to manage work orders for other systems in the network.

**Figure 1-1** *The WorkOrder driver Connected to the Identity Vault*



The WorkOrder driver does not replace the workflow functionality of the User Application.

## 1.2 Terminology

The following terms are used by the WorkOrder driver:

- ♦ **Due Date:** The date and time the work order is to be executed.
- ♦ **Content:** The definition of the work that is to be processed.
- ♦ **Interval:** The amount of time until the work order is to be repeated.
- ♦ **Dependency:** The distinguished name of a work order that should be completed first.
- ♦ **Status:** The value returned by the driver after the work order was processed (Configured, Error, etc.).
- ♦ **Process Log:** The description of the events that occurred when the work order was processed.
- ♦ **DeleteDueDate:** The date the work order will be deleted from the Identity Vault.
- ♦ **Pending:** A work order that is not yet due.

## 1.3 Key Functions

Review the following key functions to see if they are supported by the WorkOrder driver.

### 1.3.1 Local Platforms

The WorkOrder driver can run on the Identity Manager machine.

### 1.3.2 Remote Platforms

The WorkOrder driver works on all the local and remote platforms that the engine runs on.

### 1.3.3 Role-Based Entitlements

The WorkOrder driver does not support Role-Based Entitlements.

### 1.3.4 Password Synchronization Support

The WorkOrder driver does not support Password Synchronization.

### 1.3.5 Synchronized Objects

The WorkOrder driver synchronizes WorkOrder objects.

## 1.4 WorkOrder Driver Features

Driver features:

- ♦ **Schedules work orders:** The WorkOrder driver allows work to be scheduled for a specific date and time.
- ♦ **Supports dependent work orders:** If a work order is dependent on another work order, it is not processed until the dependent work order has been processed successfully.
- ♦ **Repeats work orders:** The driver allows for work orders to be repeated at a set interval.
- ♦ **Facilitates tracking and accountability of work orders:** Each work order carries with it the creator and main contact of the work order, a description of the action taken and the errors it encountered.

## 1.5 The Subscriber and Publisher Channels

In the driver configuration, the Subscriber channel processes only events that pertain to work orders. For many drivers, the Subscriber channel performs changes in the third-party application in response to events in the Identity Vault. However, for the WorkOrder driver, the Publisher channel is the agent that performs the work orders.

Through the Publisher channel, the Identity Manager 3.5 WorkOrder driver queries the Identity Vault for work orders. The Publisher channel then configures the work orders in the driver.

For more information on how the Subscriber channel and Publisher channel are configured, see [Section 6.9, “Configuring the Subscriber and Publisher Channels,” on page 35](#).



# Planning

Planning issues vary significantly, depending on your goals and the environment. This section provides a starting point to plan your customized WorkOrder driver.

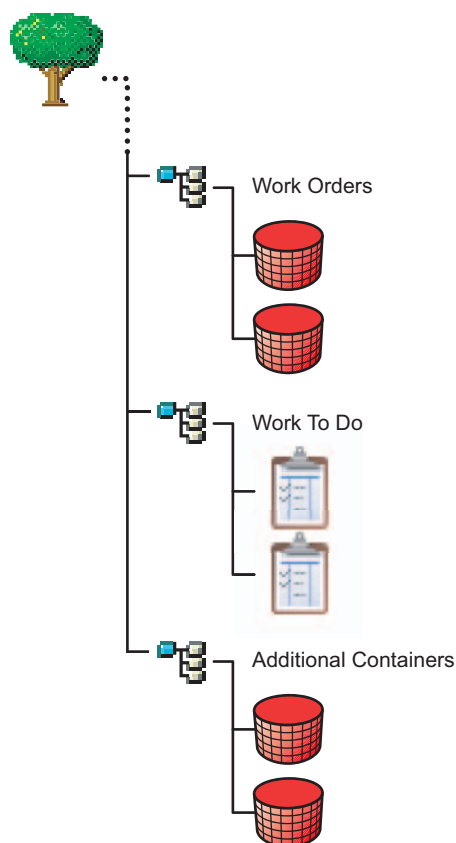
- ♦ [Section 2.1, “Planning Issues for All Configurations,” on page 15](#)
- ♦ [Section 2.2, “Designing the Driver,” on page 16](#)
- ♦ [Section 2.3, “Prerequisites,” on page 16](#)

## 2.1 Planning Issues for All Configurations

Use the following list to plan how you want to implement the WorkOrder driver.

- ♦ Specify or create containers to hold the new objects used by the WorkOrder driver. Driver configuration requires that container objects be provided for the creation and processing of DirXML-WorkOrder objects, as well as the creation of DirXML-WorkToDo objects that are generated to process DirXML-WorkOrder objects.
  - ♦ Which container do you want to use for DirXML-WorkOrder objects?
  - ♦ Which container do you want to use for DirXML-WorkToDo objects?

**Figure 2-1** *Example of Containers for New Objects*



---

**NOTE:** You should restrict rights to these containers so that only authorized administrators can change the containers or the objects they hold.

---

- ♦ Identify when you want the driver to poll for work orders that are due. You can control the timing by using the polling interval, time or day, or both.

To learn how to configure the polling time, see [Section 6.5, “Configuration Information,” on page 32](#).

## 2.2 Designing the Driver

Novell® Identity Manager 3.5 contains a visual configuration tool that provides a simple yet powerful way to design and configure Identity Manager projects. Designer for Identity Manager allows you to:

- ♦ Graphically model the implementation
- ♦ Re-use configurations to help reduce deployment time
- ♦ Create and test scenarios to ensure proper policy definition before deploying in production
- ♦ Automatically generate project documentation for all implementation details
- ♦ Use the offline mode to safely configure implementations outside of the production environment
- ♦ Design and manage policies

Drivers can be deployed through Designer or iManager. Novell recommends that you use Designer to configure and test the drivers, and that you use iManager for administration after drivers are deployed into your environment. For more information about Designer, see the Designer Administration Guide (<http://www.novell.com/documentation/designer/>).

## 2.3 Prerequisites

Before installing the driver, ensure that you meet the following prerequisites required for the Identity Manager 3.5 WorkOrder driver:

- ♦ Novell Identity Manager 3.5 with the latest patches and product updates.
- ♦ Java Developer Kit or the Java Runtime Environment version 1.3.1 or later.
- ♦ eDirectory™ administrator username and password, so you can log in during the installation to allow schema extension. The schema extensions are described in [Appendix B, “Schema For Work Order Management,” on page 81](#).



The following sections describe how the Publisher channel and Subscriber channel work with the Novell® Identity Manager 3.5 WorkOrder driver. This driver functions differently than the traditional Identity Manager drivers, so it is important to review this information.

- ♦ [Section 3.1, “Base Configuration,” on page 17](#)
- ♦ [Section 3.2, “Subscriber Channel Functions,” on page 17](#)
- ♦ [Section 3.3, “Publisher Channel Functions,” on page 19](#)

## 3.1 Base Configuration

The base configuration demonstrates the functionality of the WorkOrder driver. It shows how the driver manages WorkOrder objects in the Identity Vault. In the base configuration, the driver processes WorkOrder objects from the configured work order container. The rules and policies in the base configuration are set up so you can create WorkOrder objects using a new object class, DirXML-WorkOrder, in a work order container. The following information explains how the base configuration works when a DirXML-WorkOrder object has been created in the directory.

## 3.2 Subscriber Channel Functions

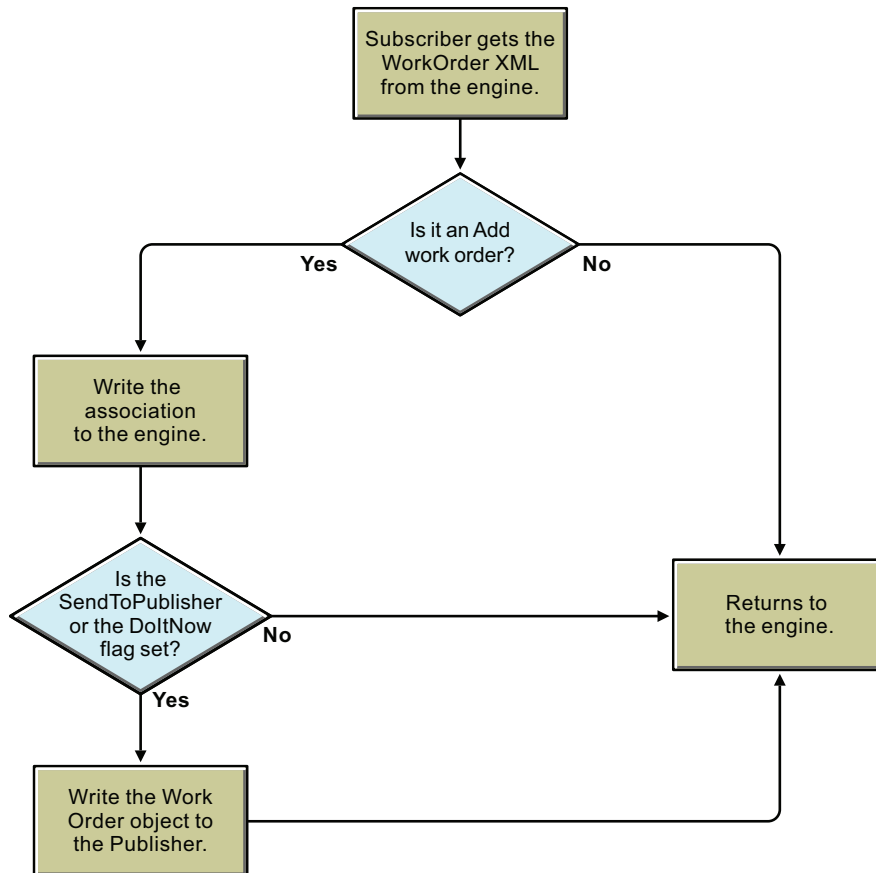
This section provides a basic understanding of the functions the Subscriber channel performs in the WorkOrder driver.

First, Placement and Create rules are configured so all new work orders that contain the required attributes are sent to the Subscriber channel. The following attributes must be present for a work order to pass the Create rule and go to the Subscriber channel:

- ♦ DirXML-nwoContent
- ♦ DirXML-nwoStatus
- ♦ DirXML-DoItNow Flag
- ♦ DirXML-SendToPublisher Flag

Figure 3-1 shows what happens when the Subscriber channel receives a work order.

**Figure 3-1** *Subscriber Channel Configuration*



The Subscriber channel performs the following actions:

1. Creates an association for each WorkOrder object it receives.
2. Checks if the DoItNow and SendToPublisher flags are set to True. If these attributes are set to True, the Subscriber channel builds a work order and sends it immediately to the Publisher channel.
3. If the DoItNow and SendToPublisher flags are not set to True, the Subscriber channel waits until the next event.

The Placement rule needs to be configured so that the dest-dn attribute has the name of the driver in it. For example, `\drivername\workorder01`. The Subscriber channel looks at the dest-dn to make sure that the work order object is for it. The Subscriber does that by making sure that the dest-dn attribute has `\drivername\` where the driver name is the name of the work order of driver.

The Placement rule can verify this is build correctly. Any work order object that comes to the Subscriber channel needs to have the dest-dn attribute correct even if the object is built through policies. This behavior was built into the driver so that multiple work order drivers could run under the same driver set and not conflict with each other.

## 3.3 Publisher Channel Functions

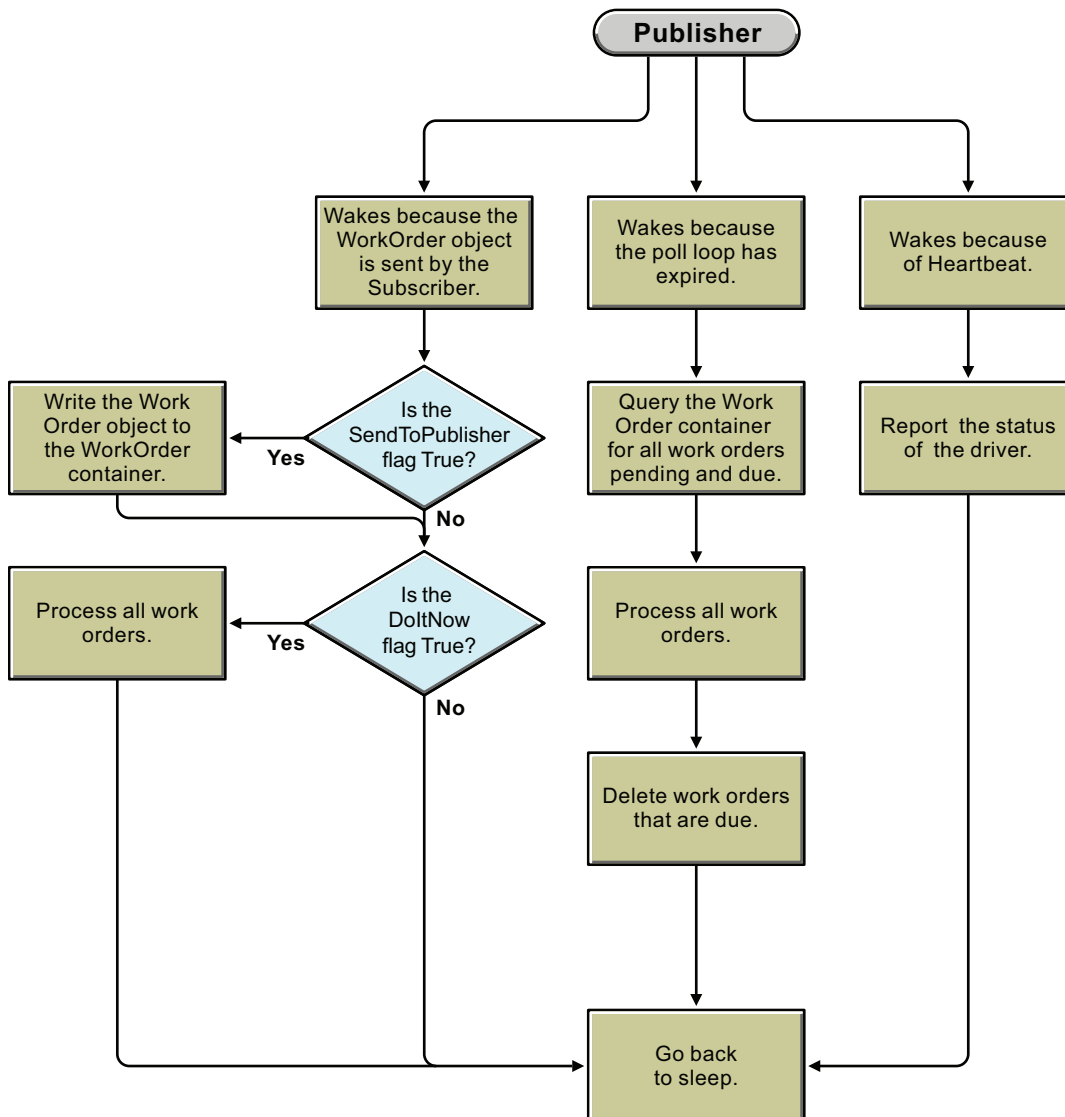
This section reviews the functions of the Publisher channel.

- ♦ [Section 3.3.1, “The Publisher Channel Wakes Up,” on page 19](#)
- ♦ [Section 3.3.2, “How the Publisher Channel Processes Work Orders,” on page 20](#)
- ♦ [Section 3.3.3, “How the Publisher Channel Deletes Work Orders,” on page 22](#)

### 3.3.1 The Publisher Channel Wakes Up

The following flowchart illustrates the Publisher channel’s action when it wakes up.

**Figure 3-2** *Publisher Channel Configuration*



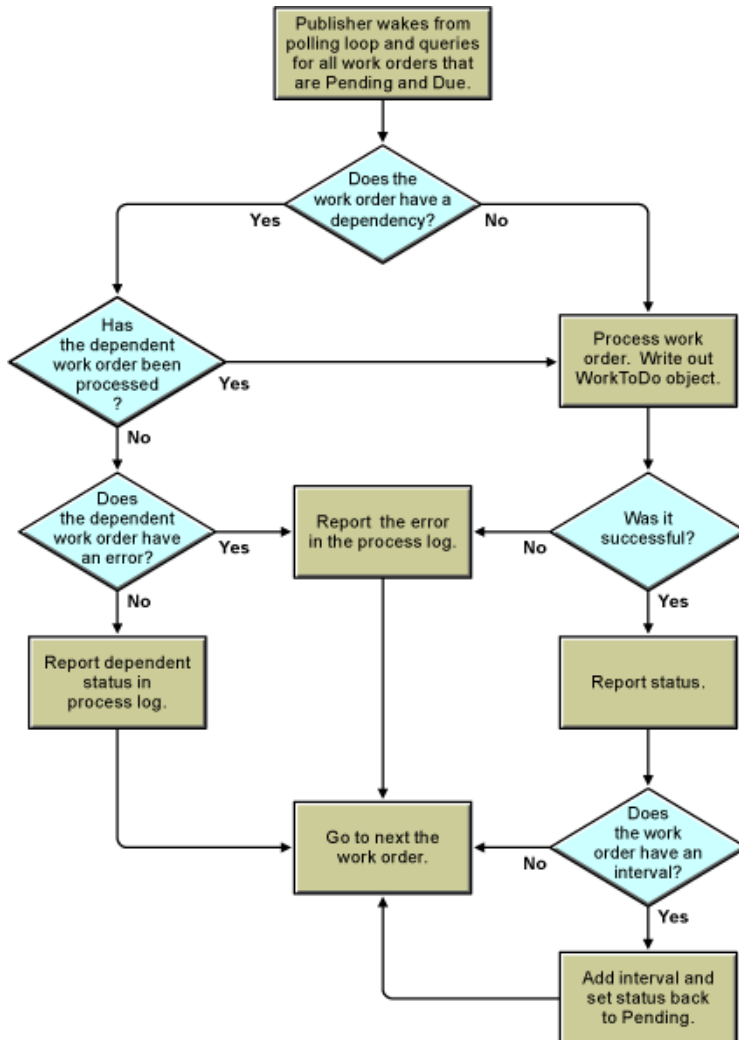
1. The Publisher channel wakes because the Subscriber channel sends a WorkOrder object. If the SendToPublisher flag is set to True, the work order is written out to the work order container. If the DoItNow flag is set to True, the work order is processed immediately.

2. The Publisher channel wakes when the polling time has expired and queries the work order container for work orders that are pending and due. The driver processes these work orders. Work orders with delete due dates are deleted.
  - a. The Publisher channel queries the work order container for work orders that are pending and due. See [Section 3.3.2, “How the Publisher Channel Processes Work Orders,” on page 20.](#)
  - b. The Publisher channel queries all work orders for expired DeleteDueDates. See [Section 3.3.3, “How the Publisher Channel Deletes Work Orders,” on page 22.](#)
3. If the driver heartbeat is configured, the driver wakes to report the driver status.

### **3.3.2 How the Publisher Channel Processes Work Orders**

After the Publisher channel queries the Identity Vault for work orders, it configures the work orders in the driver. The following flowchart illustrates how the Publisher channel processes work orders.

**Figure 3-3** *How the Publisher Processes Work Orders*



1. Before a work order is processed, the driver checks the `DependentWorkOrder` attribute to see if the work order is dependent on another work order. If there is a dependent work order, the Publisher channel queries Identity Manager to see the status of the dependent work order. If the dependent work order status is configured, the Publisher channel processes the work order. If not, the work order waits until the next polling loop to see if the dependent work order has been configured.
2. The Publisher channel performs the work orders that are due, completing the appropriate action based on the attributes of the `DirXML-WorkOrder` objects.
3. To process the work order, the driver writes a `DirXML-WorkToDo` object to the `WorkToDo` container. The `DirXML-nwoContent` attribute of the `WorkToDo` object contains the value of the `DirXML-nwoContent` attribute of the `WorkOrder` object. The default configuration does not do anything else with the `WorkToDo` object. A policy could use the `WorkToDo` object to process the work order. For example, the content attribute might contain the DN of a user object whose `LogOnDisabled` flag should be changed from `True` to `False` at the due date.
4. The Publisher channel updates the `DirXML-WorkOrder` with the results. If the `WorkToDo` object was processed without an error, the status of the work order will be changed to

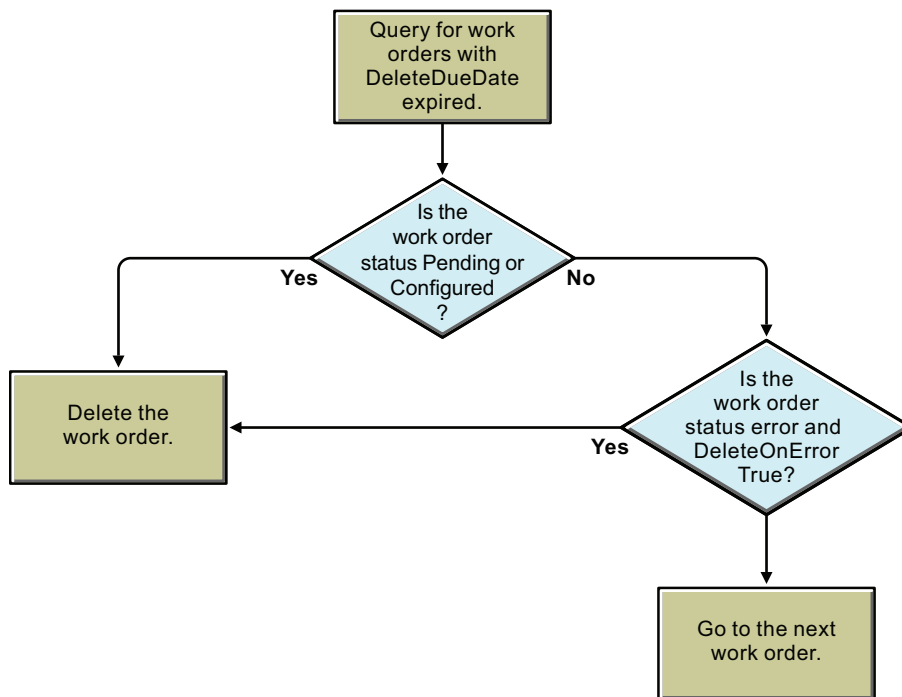
configured. If an error occurred, then the status is changed to Error. The work order process log is updated to contain the results.

5. If the WorkOrder object has a repeat interval value, the value is added to the Due Date and the work order status remains pending. This allows for the work order to be repeated. The process log contains the results.

### 3.3.3 How the Publisher Channel Deletes Work Orders

The Publisher channel now queries the work order container for work orders with an expired DeleteDueDate attribute. If the status of the work order is pending or configured, and the DeleteDueDate has expired, the work order is deleted. The work order is also deleted if it has an error status and the DeleteOnError attribute is set to True. The following flowchart illustrates this process.

**Figure 3-4** *The DeleteDueDate Process*



# Objects and Attributes Used

This section reviews the new objects and attributes used by the driver.

- ♦ [Section 4.1, “New Objects Used by the Driver,” on page 23](#)
- ♦ [Section 4.2, “DoItNow and SendToPublisher Flags,” on page 23](#)

## 4.1 New Objects Used by the Driver

Using two new object classes in the Identity Vault, the Identity Manager WorkOrder driver configures work orders and records the results. For a description of a schema for these objects, see [Appendix B, “Schema For Work Order Management,” on page 81](#).

- ♦ [Section 4.1.1, “DirXML-WorkOrder Object,” on page 23](#)
- ♦ [Section 4.1.2, “DirXML-WorkToDo Object,” on page 23](#)

### 4.1.1 DirXML-WorkOrder Object

The DirXML-WorkOrder object delays the work order to be processed until the scheduled date and time or until a dependent work order is configured. The driver also repeats work orders if the work order has a repeating interval.

If the work order is marked DoItNow, the driver performs it immediately and doesn't wait for a polling time or time of day. To learn how to use the DoItNow and SendToPublisher flags, see [Section 4.2, “DoItNow and SendToPublisher Flags,” on page 23](#).

An iManager plug-in is provided to help you create and maintain work orders. To learn how to use the plug-in, see [Section 7.2, “Using the iManager Plug-in,” on page 37](#).

### 4.1.2 DirXML-WorkToDo Object

The driver creates this object and writes it to the Identity Vault to process the work order. The Value of the WorkOrder Content attribute becomes the value of the DirXML-WorkToDo Content attribute. The driver sends this object to the Identity Vault and returns the status of the work order (Configured, Error, etc.) and writes it in the ProcessLog attribute. Any results or information available to the driver are recorded in the ProcessLog.

If the work order has a repeat attribute, the work order gets a new due date with the interval added and the status remains pending, allowing it to be processed again on the new due date.

## 4.2 DoItNow and SendToPublisher Flags

The Novell Identity Manager 3.5 WorkOrder driver has two flags to initiate a work order event.

- ♦ [Section 4.2.1, “DoItNow Flag,” on page 24](#)
- ♦ [Section 4.2.2, “SendToPublisherFlag,” on page 24](#)

### **4.2.1 DoltNow Flag**

When this flag is set to True, the Subscriber channel wakes up the Publisher channel by sending the work order to the Publisher channel. This allows the Publisher channel to perform the work order immediately instead of waiting for the next polling time or polling interval.

Use this flag when you want the work order completed immediately. You can set this flag to True when you manually create a work order, or in an automated solution you can use policies to determine whether the flag should be set.

### **4.2.2 SendToPublisherFlag**

When this flag is set to True for a work order, the Subscriber channel sends the work order to the Publisher channel and the Publisher channel writes the WorkOrder object to the WorkOrder container specified in the configuration parameters.

This flag is usually set to False. However, if a work order is created by a policy in response to an event in the Identity Vault, setting the flag to True enables the work order to be written in the work order container.



# Customizing the Driver

Identity Manager 3.5 is designed so you can customize the driver for your specific business needs. To customize the WorkOrder driver, you need to know what the default template does. The following sections explain the default functionality of the driver and how to customize the driver to meet your business needs.

- ♦ [Section 5.1, “Default Driver Parameters,” on page 25](#)
- ♦ [Section 5.2, “Customizing the Driver,” on page 25](#)
- ♦ [Section 5.3, “Additional Solutions,” on page 25](#)

## 5.1 Default Driver Parameters

The driver import file defines some basic functionality in the driver. How the driver handles this information for each channel is explained in the tables in [Section 6.5, “Configuration Information,” on page 32](#) and [Section 6.9, “Configuring the Subscriber and Publisher Channels,” on page 35](#).

## 5.2 Customizing the Driver

- ♦ [Section 5.2.1, “Customizing Policies,” on page 25](#)
- ♦ [Section 5.2.2, “Customizing Driver Parameters,” on page 25](#)

### 5.2.1 Customizing Policies

To change the default functionality of the driver, use the Policy Builder to change the policies. For more information, see the *Policy Builder and Driver Customization Guide* (<http://www.novell.com/documentation/idm/>)

### 5.2.2 Customizing Driver Parameters

You can also change the default functionality of the driver by changing the driver’s parameters.

- 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 and select *Edit properties*.
- 4 Select *Driver Configuration* and scroll down to *Driver Parameters*.
- 5 Make the changes you want, then click *OK*.

## 5.3 Additional Solutions

The WorkOrder driver can be used in conjunction with other drivers to create and schedule work orders. The following samples illustrate two possible solutions for customizing your driver.

- ♦ [Section 5.3.1, “Human Resource Sample Using an HR Driver,” on page 26](#)
- ♦ [Section 5.3.2, “Human Resource Sample without an HR Driver,” on page 28](#)

### 5.3.1 Human Resource Sample Using an HR Driver

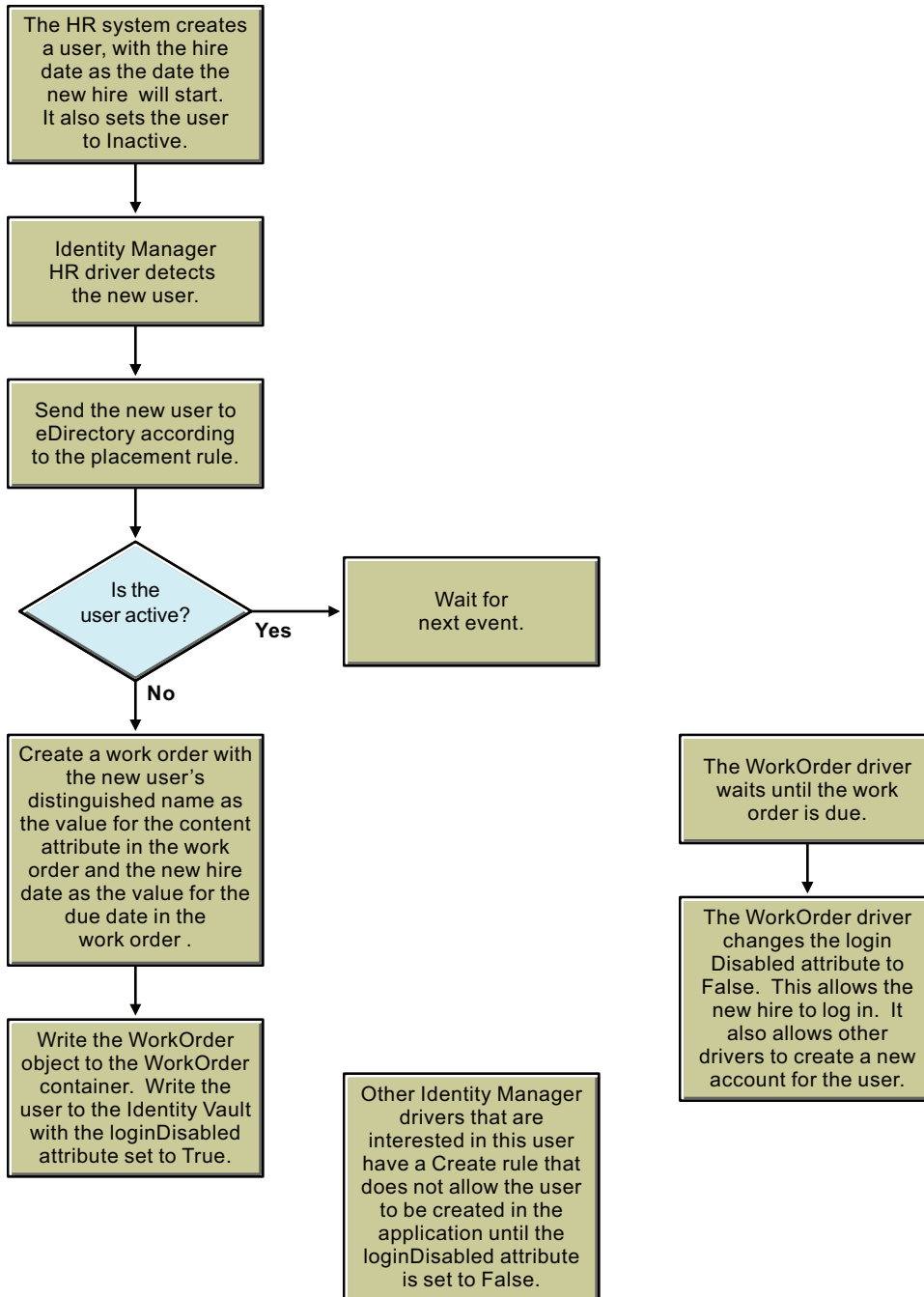
The following samples illustrate how the WorkOrder driver can be used with an HR driver to create a new user and postpone activating the new employee's access to the system until the hire date. Figure 5-1 illustrates how these drivers work together in the sample configuration.

In this scenario, the new employee is hired, but does not begin work until a future date and time. The new employee is put into the HR system with his hire date set. The employee is marked as not active and does not have access to the system.

The HR Identity Manager driver writes the new user object to the Identity Vault. A policy in that driver checks to see if he is active. If he is active, the driver performs the work. If he is not active, the policy creates a work order to activate the new employee on the hire date. The work order is marked pending. A policy in the WorkOrder driver processes the work order on the hire date. The policy in the WorkOrder driver sets the user object's loginDisabled attribute to False, allowing the user to log in.

The sample could be extended to allow other Identity Manager drivers to have a Create rule to disallow the creation of the user object in other connected systems until the user object's loginDisabled attribute is set to False. The result is that the user's system access is provisioned on his hire date and not before.

**Figure 5-1** Data Flow with an HR Driver



## Human Resource Driver Policies

The following policies show how to implement this sample. In the sample, the Delimited Text Driver is acting as the HR system interface. The Delimited Text Driver is configured to provide the needed attributes: LastName, FirstName, HireDate, Disabled.

## Mapping Rule

The mapping Rule maps the attributes used in the Delimited Text Driver to attributes in the Identity Vault. You can view the sample at [hr-drv-schema-map.xml](#) ([../samples/hr-drv-schema-map.xml](#)).

## Filter

The filter attribute allows only the attributes that are needed by this example to be passed through. The DirXML-DueDate is notify only. This attribute should not be applied to the user object. However, it should be available for the Command Transformation. You can view the sample at [hr-drv-schema-map.xml](#) ([../samples/hr-drv-filter.xml](#)).

## Command Transformation Policy

The Command Transformation policy checks to see if a user object is being added to the Identity Vault. It also ensures that the loginDisabled attribute is set to True. If the conditions are satisfied, the policy then creates a work order and places it in the WorkOrder container. The WorkOrder driver looks in this container for work orders to process.

The policy puts the DN of the user that was created in the DirXML-nwoContent attribute. It also puts the DirXML-DueDate from the user into the WorkOrder object DirXML-DueDate and then sets the work order status to “pending”. You can view the sample at [hr-drv-cmd-transform.xml](#) ([../samples/hr-drv-cmd-transform.xml](#)).

## WorkOrder Driver Policy

The WorkOrder driver Policy only uses the Publisher Command Transformation policy, described below:

### Publisher Command Transformation Policy

The Work Order Command Transformation policy checks to see that a DirXML-WorkToDo object is being added. If it is, the policy gets the DN of the user from the DirXML-nwoContent attribute. It then sets the users Login Disable attribute to False. This allows the user to log in.

---

**NOTE:** `<do-add-dest-attr-value class-name="User" direct="true" name="Login Disabled">` should not be used.

---

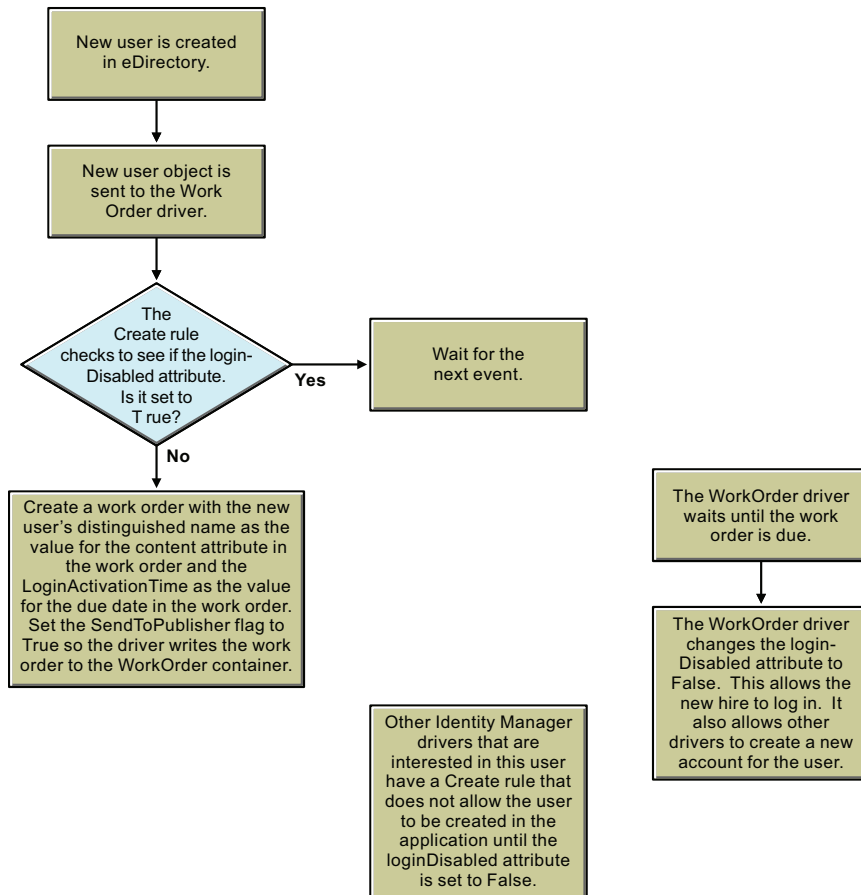
When direct is equal to True, the action is performed as desired, but the results are not returned to the driver. Therefore, the driver cannot report the results of the write correctly. You can view the sample at [hr-wo-drv-pub-cmd-transform.xml](#) ([../samples/hr-wo-drv-pub-cmd-transform.xml](#)).

## 5.3.2 Human Resource Sample without an HR Driver

This sample creates a new user and postpones activating the new employee’s access to the system until the hire date by putting policies in the WorkOrder driver to create the work order. Figure 5-2 illustrates this sample configuration.

When a new user object is created in the Identity Vault, a policy in the WorkOrder driver checks to see if the loginDisabled attribute is set to True. If it is not set to True, the Create rule blocks the event. If it is set to True, the policy creates a work order to set the loginDisabled attribute on the user to False on the loginActivationTime.

**Figure 5-2** *Data flow without an HR Driver*



The following policies show how to implement the sample configuration:

## Filter Additions

Modify the filter to allow user objects with loginActivationTime and loginDisabled attributes to synchronize on the Subscriber channel. You can view the sample at [wo-filter.xml](#) ([../samples/wo-filter.xml](#)).

## Subscriber Create Rule

The Create rule vetoes this event if the loginActivationTime or the loginDisabled attributes are not present. It also vetoes this event if the loginDisabled attribute is set to False. You can view the sample at [wo-create.xml](#) ([../samples/wo-create.xml](#)).

## Subscriber Command Transform

This policy checks to see if the event is an Add of a user object. If that is true, the policy creates a WorkOrder object. The DN of the user object is added to the DirXML-nwoContent attribute. The DirXML-DueDate is set to the loginActivationTime. The DirXML-nwoStatus is set to pending. The DirXML-nwoSendToPublisher attribute is set to True.

This work order has not yet been created in the Identity Vault. This sample configuration creates the work order in the Identity Vault by setting the SendToPublisher attribute to True. This tells the

publisher in the WorkOrder driver to write the policy to the work order container that it looks in for work orders to be processed. You can view the sample at [wo-sub-cmd-transform.xml](#) ([../samples/wo-sub-cmd-transform.xml](#)).

### **Work Order E-Mail Notification of Work Order Completion**

This policy can be used with the WorkOrder driver to send e-mail notification of a completed work order. This policy is in the Publisher Command Transform. The policy checks to see if a DirXML-WorkOrder modify event is happening. If it is, it builds an e-mail from the status, description, and process log of the work order and then sends it to an administrator. This notifies the administrator that a work order has been processed and gives them the results. You can view the sample at [wo-pub-cmd-transform.xml](#) ([../samples/wo-pub-cmd-transform.xml](#)).

# Installing and Configuring

The WorkOrder driver is installed as part of the Novell® Identity Manager 3.5. The following sections review the tasks required to install and configure the driver:

- ♦ [Section 6.1, “Prerequisites,” on page 31](#)
- ♦ [Section 6.2, “Files Included with the WorkOrder Driver,” on page 31](#)
- ♦ [Section 6.3, “Creating Containers,” on page 31](#)
- ♦ [Section 6.4, “Installation,” on page 32](#)
- ♦ [Section 6.5, “Configuration Information,” on page 32](#)
- ♦ [Section 6.6, “Importing the Driver Configuration,” on page 32](#)
- ♦ [Section 6.7, “Configuring the Driver in Designer,” on page 33](#)
- ♦ [Section 6.8, “Configuring the Driver in iManager,” on page 33](#)
- ♦ [Section 6.9, “Configuring the Subscriber and Publisher Channels,” on page 35](#)

## 6.1 Prerequisites

Before installing the WorkOrder driver as part of Novell Identity Manager 3.5, see [Section 2.3, “Prerequisites,” on page 16](#).

## 6.2 Files Included with the WorkOrder Driver

The following files are included with the WorkOrder driver:

- ♦ WorkOrder.jar is the driver shim. This file is copied to the server where Identity Manager is installed or to the server where Remote Loader is installed.
- ♦ WorkOrderDriver-IDM3\_5\_0-V1.XML is the import file with configuration and policies for the driver. This file is copied to the server where iManager is installed.
- ♦ The latest iManager plug-ins are included for creating and managing work orders in the directory.

## 6.3 Creating Containers

The driver configuration requires that container objects be provided for creating and processing DirXML-WorkOrder objects and DirXML-WorkToDo objects. These objects are generated after the Publisher configures the DirXML-WorkOrder objects.

To plan the implementation of the WorkOrder driver:

- ♦ Identify the container for the DirXML-WorkOrder objects
- ♦ Identify the container for the DirXML-WorkToDo objects
- ♦ Identify additional containers to store work orders based on the return status of configured work orders. For example, separate containers can store work orders with error or warning messages.

## 6.4 Installation

The WorkOrder driver is installed automatically with the Novell Identity Manager 3.5. To install, see the *Identity Manager 3.5 Installation Guide*.

## 6.5 Configuration Information

The WorkOrder driver configuration file is a sample configuration you customize to your environment. After the WorkOrder driver is installed, you must import and customize the driver configuration file in either Designer or iManager.

As you import the driver configuration file, you are prompted for certain information depending on the configuration selection you made. The following table explains the parameters you must provide during initial driver configuration.

**Table 6-1** *Configuring the Driver*

Parameter Name	Parameter Descriptions
Driver Name	The actual name you want to use for the driver.
WorkOrders Container	The name of the container where work orders are to be stored.
WorkToDo Container	The name of the container to store configured work orders.
Polling Method	Specifies the polling method by interval or time. <i>Interval</i> indicates that the driver will poll at a specified time interval. <i>Polling by time</i> indicates a specific time of day.
Driver Heartbeat	Specifies if the Publisher should emit heartbeat documents. The driver emits heartbeat documents to indicate to the Identity Manager that the driver is still functioning.
Install Driver as Remote or Local	Select <i>Remote</i> to configure the driver for use with the Remote Loader service.  Select <i>Local</i> to configure the driver for local use.
Poll Interval	The polling interval (in minutes) at which the Publisher channel polls the Identity Manager Vault for work orders to be configured.
Poll Time	Time of day the Publisher channel wakes up to check the Identity Manager Vault for work orders to be configured.

## 6.6 Importing the Driver Configuration

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



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.  
If you place this driver in a new driver set, you must specify a driver set name, context, and associated server.
- 3 Select *Import a Driver Configuration from the Server*, then select *WorkOrderDriver-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. Refer to [Section 6.5, "Configuration Information," on page 32](#) for more information.
- 4 Specify the driver's parameters, then click OK to import the driver. Refer to [Section 6.5, "Configuration Information," on page 32](#) for details.
- 5 Define Security Equivalences and Exclude Administrative Roles.
  - ♦ **Security Equivalences:** Select the eDirectory™ object from which the driver will get its rights. The driver object must be granted sufficient eDirectory rights to any object it reads or writes. Normally, the driver should be given security equal to Admin.
  - ♦ **Exclude Administrative Roles:** Select the objects you want to prevent from being synchronized to other systems. Identity Manager does not send event data for objects in this list to the WorkOrder driver.
- 6 Review the driver objects in the Summary page, then click *Finish*.

## 6.7 Configuring the Driver in Designer

Designer allows you to import the basic driver configuration file for the WorkOrder driver. This file creates and configures the objects and policies needed to make the driver work properly.

There are different ways to import the driver configuration file. The following example documents one method to import the driver's configuration in Designer.

- 1 Open a project in Designer. Go to the Modeler and right-click the Driver Set object, then select *New>Driver* to open a project in Designer.
- 2 Define the driver parameters.  
For information on the settings, see [Chapter 5, "Customizing the Driver," on page 25](#).
- 3 After defining the driver parameters, click *OK* to import the driver.
- 4 Customize and test the driver before deploying the driver into the production environment. To learn how to customize the driver, see [Chapter 5, "Customizing the Driver," on page 25](#).
- 5 After the driver is fully tested, deploy the driver into the Identity Vault as described in "Deploying a Driver to an Identity Vault" in the *Designer 2.0 for Identity Manager 3.5: Administration Guide* (<http://www.novell.com/documentation/designer>).

## 6.8 Configuring the Driver in iManager

To import the WorkOrder driver configuration in iManager:

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

- 2 Select an existing driver set or select a new driver set.
- 3 If you selected an existing driver set, continue with **Step 4**.  
or  
If you placed the driver in a new driver set, skip to **Step 5**.
- 4 If you selected an existing driver set:
  - 4a Browse to and select the driver set, then click *Next*.
  - 4b Skip to **Step 6**.
- 5 If you selected to place the driver in a new driver set, click *Next*, then define the properties of the new driver set:
  - 5a Specify the name of the driver set.
  - 5b Browse to and select the context where the driver set is created.
  - 5c Browse to and select the server you want to associate with the driver set.
  - 5d Select the *Create a new partition on this driver set* option.
  - 5e Click *Next*.

Novell recommends that you create a partition for the driver object. For Identity Manager to function, the server that is associated with the driver set must hold a real replica of the Identity Manager object. If the server holds a Master or Read/Write replica of the context where the WorkOrder objects are created, the partition is not required.
- 6 Select the WorkOrder driver, then click *Next*.
- 7 Define the driver parameters, then click *Next*.

For information on the settings, see **Section 6.5, “Configuration Information,”** on page 32.
- 8 Assign security rights to the WorkOrder driver object:
  - 8a Select *Define Security Equivalences*.
  - 8b Click *Add*, then browse to and select a user object that has the rights the driver needs to have on the server.

Many administrators give the WorkOrder object security equivalence to the Administrator User object in the Identity Vault. However, you might want to create another object, such as a DriversUser, and assign security equivalence to that user.

Whatever object you select must have Read/Write access to all objects the driver will read or write.
  - 8c Click *OK* twice.
- 9 Exclude the administrative roles from replication.
  - 9a Select *Exclude Administrative Roles*.
  - 9b Click *Add*.
  - 9c Browse to the Identity Vault and select the security-equivalence object that you specified in **Step 8** (for example, DriversUser) and exclude the object from replication.

---

**IMPORTANT:** If you delete the security-equivalence object, you remove the rights from the driver. Consequently, the driver can't make changes to Identity Manager.

---

If there are objects that are currently excluded, they do not appear in the Excluded users list unless you select *Retrieve Current Exclusions*.

- 9d Click *OK* twice.
- 10 Click *Next*.
- 11 View the summary, then click *Finish*.
- 12 To view information about configuring additional driver properties, see [Chapter 5](#), “Customizing the Driver,” on page 25.

## 6.9 Configuring the Subscriber and Publisher Channels

This section provides the rules needed to configure the Subscriber channel and the Publisher channel. For an overview on how the Subscriber and Publisher channels work, see [Section 3.2](#), “Subscriber Channel Functions,” on page 17 and [Section 3.3](#), “Publisher Channel Functions,” on page 19.

### 6.9.1 Configuring the Subscriber Channel

The Subscriber channel processes only events that pertain to the work orders. The following table lists the rules and policies used in configuring the Subscriber channel.

**Table 6-2** *Configuring the Subscriber Channel*

Rule or Policy	What it does
Subscriber Filter	Allows only events for nwoWorkOrder Objects to be processed.
Event Transformation	Not used in the sample configuration.
Matching Rule	Not used in the sample configuration.
Create Rule	<p>Contains rules only for WorkOrder objects.</p> <p>Requires values for the following attributes on a WorkOrder object:</p> <p>nwoStatus</p> <p>nwoSendToPublisher</p> <p>nwoDoltNow</p> <p>nwoContent</p> <p>If the values are not present, the work order is not sent to the Publisher channel and it is not configured by the driver.</p> <p>For a description of these attributes, see <a href="#">Appendix B</a>, “Schema For Work Order Management,” on page 81.</p>

Rule or Policy	What it does
Placement Rule	Maps work orders from the work order container you specified to the driver. This mapping is necessary so that the Subscriber channel can check the work orders to see if the DoltNow flag is set to True.
Command Transformation	Not used in the sample configuration.
Schema Mapping	Maps the eDirectory namespace to the Work Order namespace.
Output Transformation	Not used in the sample configuration.

## 6.9.2 Configuring the Publisher Channel

The following table lists the rules and policies used to configure the Publisher channel.

**Table 6-3** *Configuring the Publisher channel*

Rule or Policy	What it does
Schema Mapping	Maps the WorkOrder driver namespace to the eDirectory namespace.
Event Transformation	Not used in the sample configuration.
Publisher Filter	Allows only events for nwoWorkOrder objects to be processed.
Matching Rule	Not used in the sample configuration.
Placement Rule	Places WorkOrder objects in the correct container as defined in the driver's configuration parameters.  Places nwoWorkToDo objects in the correct container.
Command Transformation	Not used in the sample configuration.

# Creating and Managing Work Orders

There are two ways to create work orders in the Novell® Identity Manager 3.5 WorkOrder driver. The following sections review how this is accomplished:

- ♦ [Section 7.1, “Work Orders Created from Events,” on page 37](#)
- ♦ [Section 7.2, “Using the iManager Plug-in,” on page 37](#)

## 7.1 Work Orders Created from Events

Work orders can be created from events on the Subscriber channel. For an example of how this is done, see [Section 5.3, “Additional Solutions,” on page 25](#).

## 7.2 Using the iManager Plug-in

An iManager plug-in is provided to help you create and maintain work orders.

- ♦ [Section 7.2.1, “Creating and Deleting a New Work Order,” on page 37](#)
- ♦ [Section 7.2.2, “Specifying and Editing Work Order Properties,” on page 38](#)

### 7.2.1 Creating and Deleting a New Work Order

To create a new work order:

- 1 In iManager, select *Work Orders > Work Order Management*.
- 2 Select *New* under Work Order Management.
- 3 Specify the name for the work order. This value creates the CN of the resource object in the Identity Vault.
- 4 Select the new work order to specify the properties.

To delete a work order, select the box and click *Delete*.

To sort the list according to name, status or date, click the appropriate selection button on the table heading. After selecting the column, an arrow icon will appear next to the column heading. This allows you to sort in ascending or descending order.

To filter the work order list:

- 1 Click *Show* under Work Order Management.
- 2 From the drop-down menu, select the filter type:
  - ♦ **Show all:** All work orders associated with the driver are listed.
  - ♦ **Configured:** Only configured work orders associated with the driver are listed.
  - ♦ **Error:** Only work orders with an error status are listed.
  - ♦ **On Hold:** Work orders that have been manually placed on hold are listed.

- ♦ **Pending:** Work order that are not yet due are listed.

## 7.2.2 Specifying and Editing Work Order Properties

The following section explains the Work Order Property Page that allows you to create or edit a DirXML-nwoWorkOrder object. This interface allows you to specify the properties for new work orders, or edit the properties of existing work orders.

- 1 Select the status of the work order. Normally, the status is marked *Pending*. You can stop a work order by marking it *On Hold*.
- 2 Select whether to have the driver do the work order immediately, or use the calendar to schedule the work order due date.
- 3 Specify if it is a repeating work order. Specify the time interval by choosing the number of weeks, days, hours, or minutes before the work order is to be repeated. The work order stops repeating on the Delete DueDate, or until it is manually deleted by editing the work order, or if the driver sends an error message.
- 4 Use the calendar to specify the date the work order will be deleted.
- 5 List any dependent work orders by clicking the Search icon and selecting dependent work order. Click the Subtract icon to delete dependent work orders.
- 6 Write the information about the work order. This attribute is passed through to the WorkToDo object when the work order is processed.
- 7 (Optional) Assign a unique work order number. This value can be assigned by a corporate work order system.
- 8 Indicate the information of the person responsible for the work order.
- 9 The Work Order Processing Log is for the driver to log the results of the work order. Use this field to check on the status of a work order or to identify any problems the driver encountered while attempting to configure the work order. The following results can be recorded:
  - ♦ **Pending:** The driver is waiting for the due date to complete the work order.
  - ♦ **Configured:** The work order has been successfully processed.
  - ♦ **Error:** The driver was unable to perform the work order.
  - ♦ **Warning:** There is a warning regarding the work order. For example, if the work order has a dependent work order with a later due date, the driver processes a warning.
- 10 Provide a description of the work order.
- 11 In the Work Order content field, indicate the data used by the driver's rules to process the work order. For example, it might be the XML the Command Transformation uses to process the work order.
- 12 Select one of the following options when you are finished specifying or editing the work order properties:
  - ♦ Click *Apply* to save the current information and continue working.
  - ♦ Click *OK* to save and close the work order.
  - ♦ Click *Cancel* to close the work order without saving the information.

# Managing the Driver

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

- ♦ [Section 8.1, “Starting, Stopping, or Restarting the Driver,” on page 39](#)
- ♦ [Section 8.2, “Using the DirXML Command Line Utility,” on page 40](#)
- ♦ [Section 8.3, “Viewing Driver Versioning Information,” on page 40](#)
- ♦ [Section 8.4, “Reassociating a Driver Set Object with a Server Object,” on page 45](#)
- ♦ [Section 8.5, “Changing the Driver Configuration,” on page 46](#)
- ♦ [Section 8.6, “Storing Driver Passwords Securely with Named Passwords,” on page 46](#)
- ♦ [Section 8.7, “Adding a Driver Heartbeat,” on page 53](#)

## 8.1 Starting, Stopping, or Restarting the Driver

- ♦ [Section 8.1.1, “Starting the Driver in Designer,” on page 39](#)
- ♦ [Section 8.1.2, “Starting the Driver in iManager,” on page 39](#)
- ♦ [Section 8.1.3, “Stopping the Driver in Designer,” on page 39](#)
- ♦ [Section 8.1.4, “Stopping the Driver in iManager,” on page 39](#)
- ♦ [Section 8.1.5, “Restarting the Driver in Designer,” on page 40](#)
- ♦ [Section 8.1.6, “Restarting the Driver in iManager,” on page 40](#)

### 8.1.1 Starting the Driver in Designer

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

### 8.1.2 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*.

### 8.1.3 Stopping the Driver in Designer

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

### 8.1.4 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*.

### 8.1.5 Restarting the Driver in Designer

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

### 8.1.6 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*.

## 8.2 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 67](#) for detailed information about the DirXML Command Line utility. For daily tasks, use iManager or Designer.

## 8.3 Viewing Driver Versioning Information

The Versioning Discovery tool only exists in iManager.

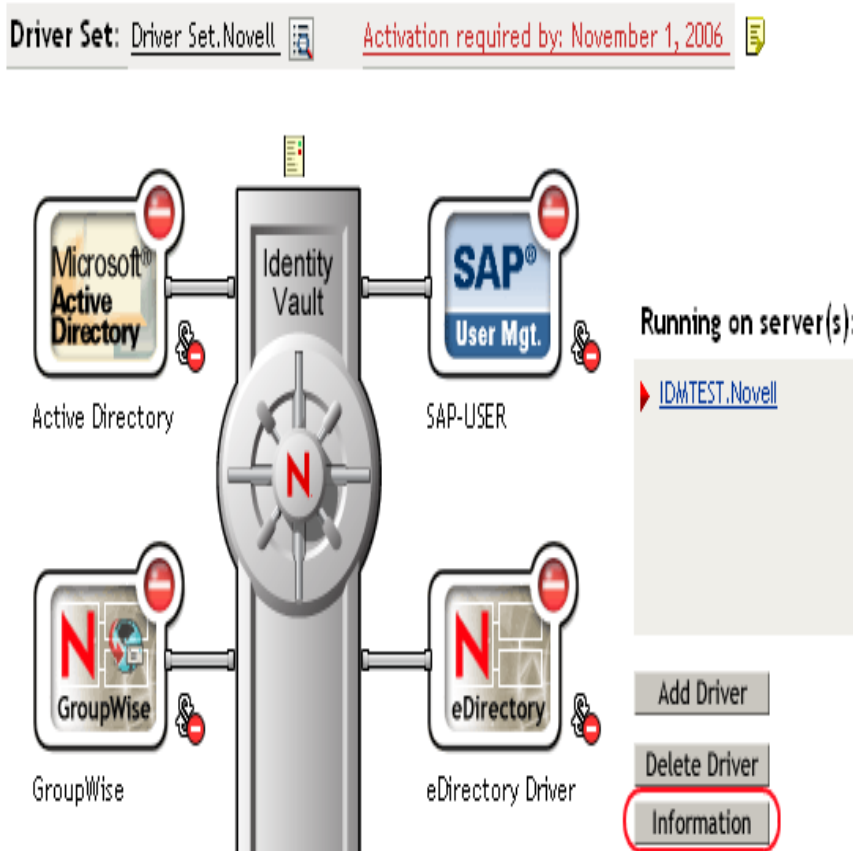
- ♦ [Section 8.3.1, “Viewing a Hierarchical Display of Versioning Information,” on page 40](#)
- ♦ [Section 8.3.2, “Viewing the Versioning Information As a Text File,” on page 42](#)
- ♦ [Section 8.3.3, “Saving Versioning Information,” on page 44](#)

### 8.3.1 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*.

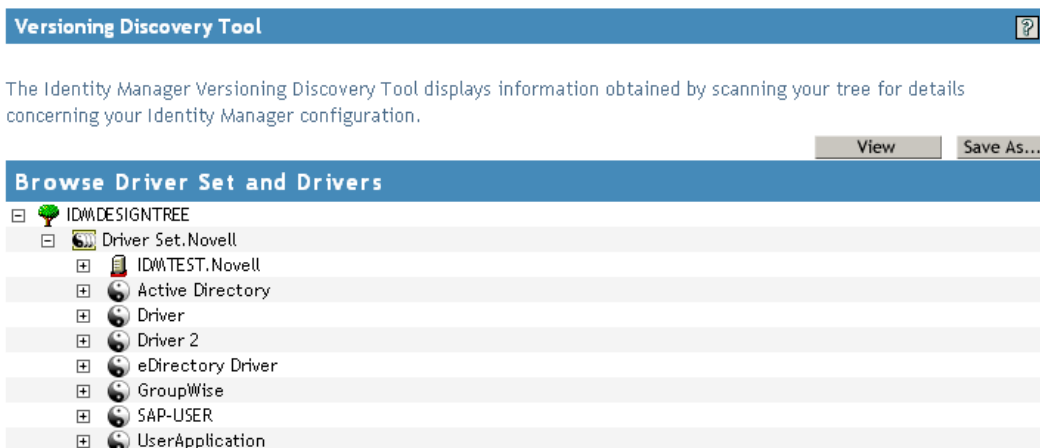


2 In the Identity Manager Overview, click *Information*.



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:

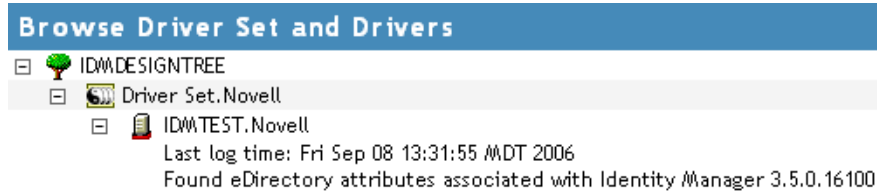
- The eDirectory™ tree that you are authenticated to
- The Driver Set object that you selected

- ♦ 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

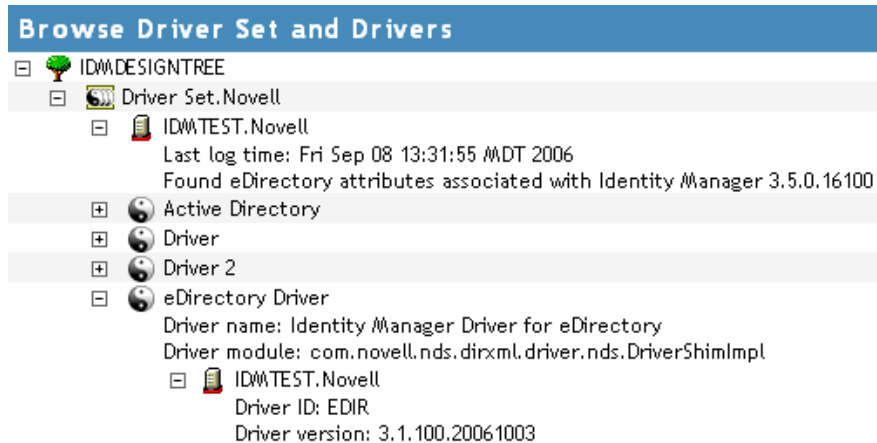
#### 4 View versioning information related to servers by expanding the server icon.



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:

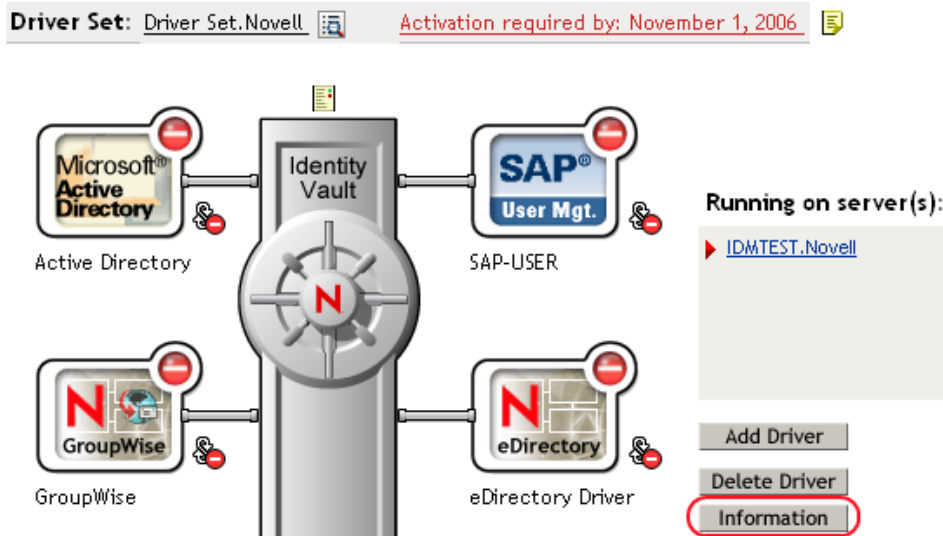
- ♦ The driver ID
- ♦ The version of the instance of the driver running on that server

## 8.3.2 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.

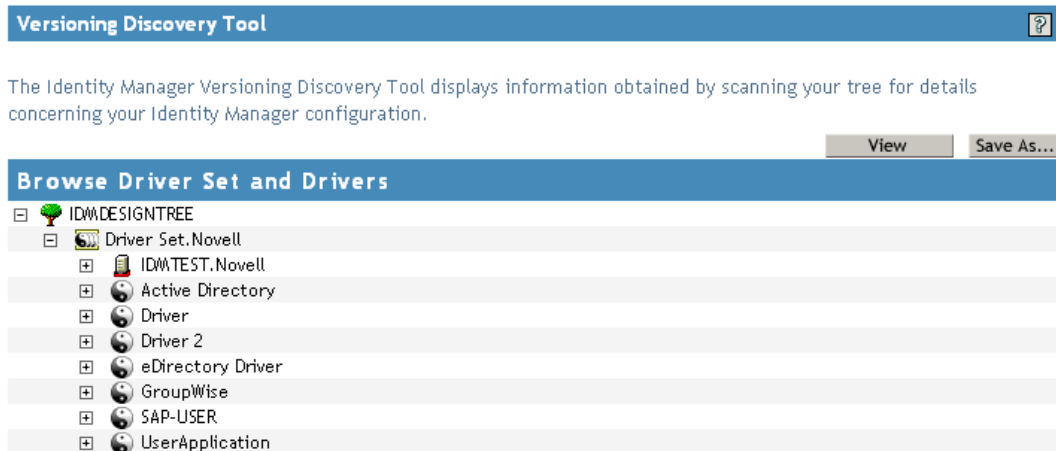
- 1 To find your Driver Set object in iManager, click *Identity Manager* > *Identity Manager Overview*, then click *Search*.

- 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.

```

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

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

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

Driver Set:  Driver Set.Novell
    Driver Set running on Identity Vault:  IDMTEST.Novell
        Last log time:  Fri Sep 08 13:31:55 MDT 2006
        Found eDirectory attributes associated with Identity Manager 3.5.0.1
    Driver:  Active Directory.Driver Set.Novell
        Driver name:  Identity Manager Driver for Active Directory and Exchange
        Driver module:  addriver.dll
        Driver Set running on Identity Vault:  IDMTEST.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:  IDMTEST.Novell

```

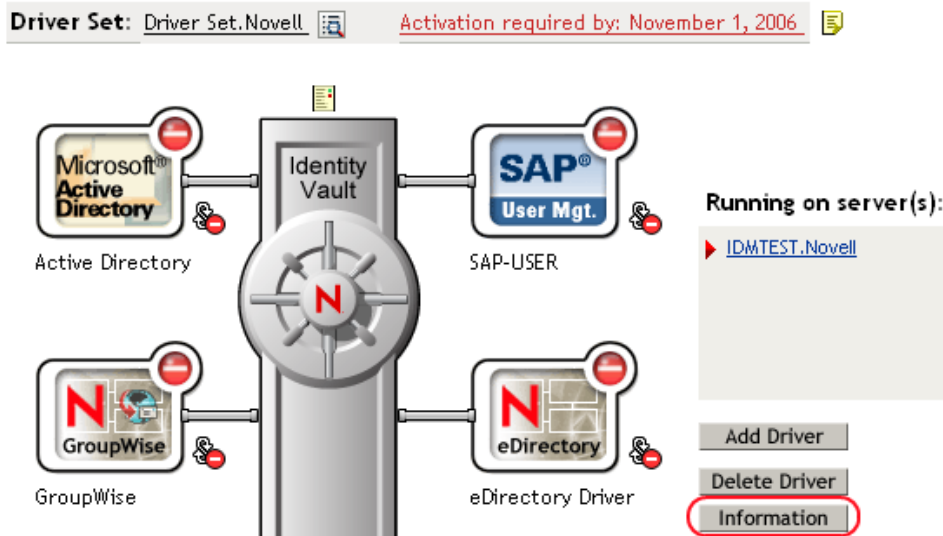
OK

### 8.3.3 Saving Versioning Information

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

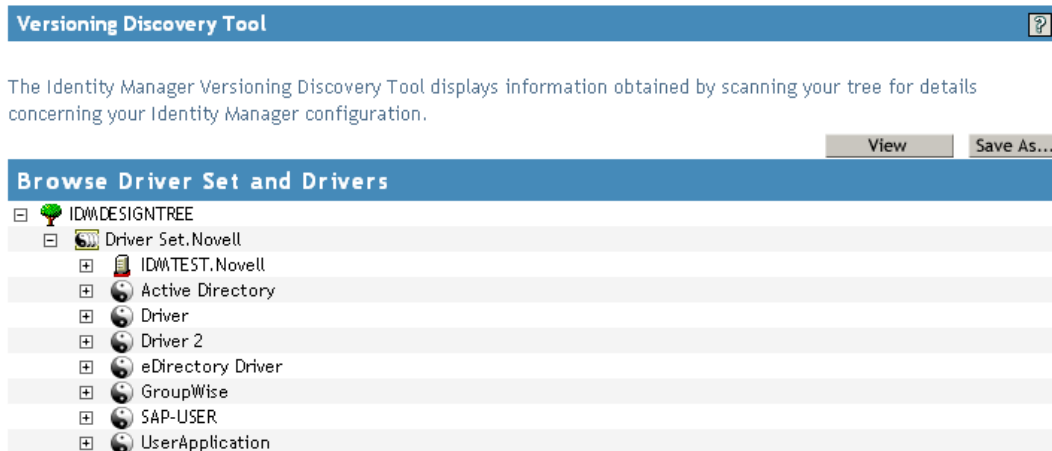
- 1 To find the Driver Set object in iManager, click *Identity Manager > Identity Manager Overview*, then click *Search*.

- 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.

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

To resolve this issue, disassociate the Driver Set object and the Server object, then reassociate them.

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

## 8.5 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 and select 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 C, “Properties of the Driver,” on page 85](#).

## 8.6 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.

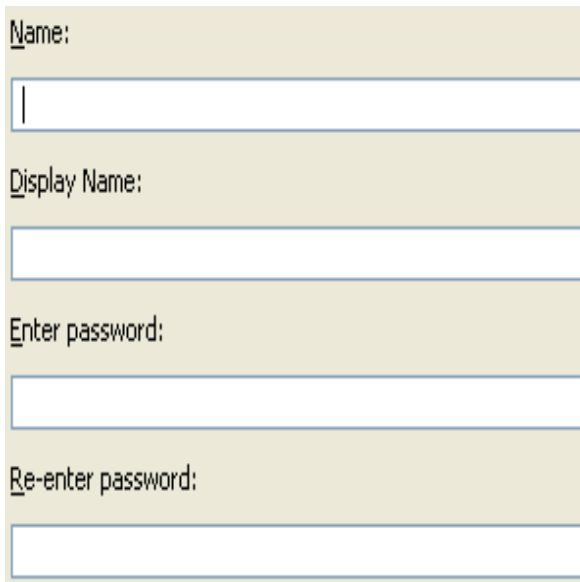
To use a named password in a driver policy, you refer to it by the name of the password, instead of using the actual password, and the Metadirectory engine sends the password to the driver. The

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 8.6.1, “Using Designer to Configure Named Passwords,” on page 47](#)
- ♦ [Section 8.6.2, “Using iManager to Configure Named Passwords,” on page 47](#)
- ♦ [Section 8.6.3, “Using Named Passwords in Driver Policies,” on page 49](#)
- ♦ [Section 8.6.4, “Using the DirXML Command Line Utility to Configure Named Passwords,” on page 49](#)

## 8.6.1 Using Designer to Configure Named Passwords

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



The screenshot shows a configuration dialog box with a light beige background. It contains four labeled text input fields stacked vertically. The labels are: 'Name:', 'Display Name:', 'Enter password:', and 'Re-enter password:'. Each label is followed by a white rectangular text box with a thin blue border. The first text box for 'Name:' contains a single vertical bar character '|'. The other three text boxes are empty.

- 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.

## 8.6.2 Using iManager to Configure Named Passwords

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

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

Identity Manager | Server Variables | General | **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: IDMTEST.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:

OK Cancel

- 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.



## 8.6.3 Using Named Passwords in Driver Policies

- ♦ “Using the Policy Builder” on page 49
- ♦ “Using XSLT” on page 49

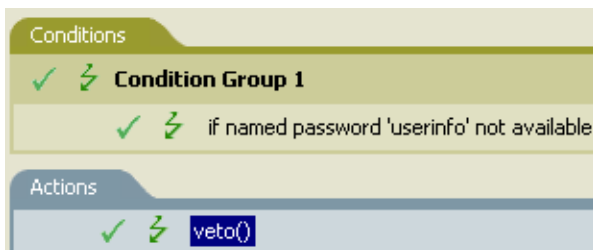
### Using the Policy Builder

The 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 the 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 8-1** 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/>
```

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

- ♦ “Creating a Named Password in the DirXML Command Line Utility” on page 50
- ♦ “Using the DirXML Command Line Utility to Remove a Named Password” on page 51

## 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 67.

- 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 67](#).

- 2 Enter your username and password.

The following list of options appears.

```
DirXML commands
```

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

```
Enter choice:
```

- 3 Enter 3 for driver operations.

A numbered list of drivers appears.

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

The following list of options appears.

```
Select a driver operation for:
```

```
driver_name
```

```
1: Start driver
2: Stop driver
3: Get driver state
4: Get driver start option
5: Set driver start option
6: Resync driver
```

```
7: Migrate from application into DirXML
8: Submit XDS command document to driver
9: Submit XDS event document to driver
10: Queue event for driver
11: Check object password
12: Initialize new driver object
13: Passwords operations
14: Cache operations
99: Exit
Enter choice:
```

**5** Enter 13 for password operations.

The following list of options appears.

Select a password operation

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

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

The list of existing named passwords is displayed.

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

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

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

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

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

```
Enter password name:
```

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

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

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

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

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

## 8.7 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. When a heartbeat is set to Yes, 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 you want the driver to send heart beat documents, set the value to Yes.
- 5 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.



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

## 9.1 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.

- ♦ [Section 9.1.1, “Adding Trace Levels in Designer,” on page 55](#)
- ♦ [Section 9.1.2, “Adding Trace Levels in iManager,” on page 57](#)
- ♦ [Section 9.1.3, “Capturing Driver Processes to a File,” on page 57](#)

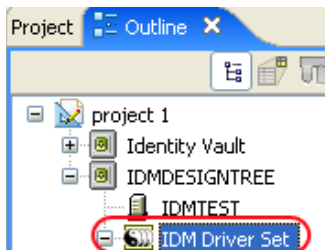
### 9.1.1 Adding Trace Levels in Designer

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

- ♦ [“Driver Set” on page 55](#)
- ♦ [“Driver” on page 56](#)

#### 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 patch 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.



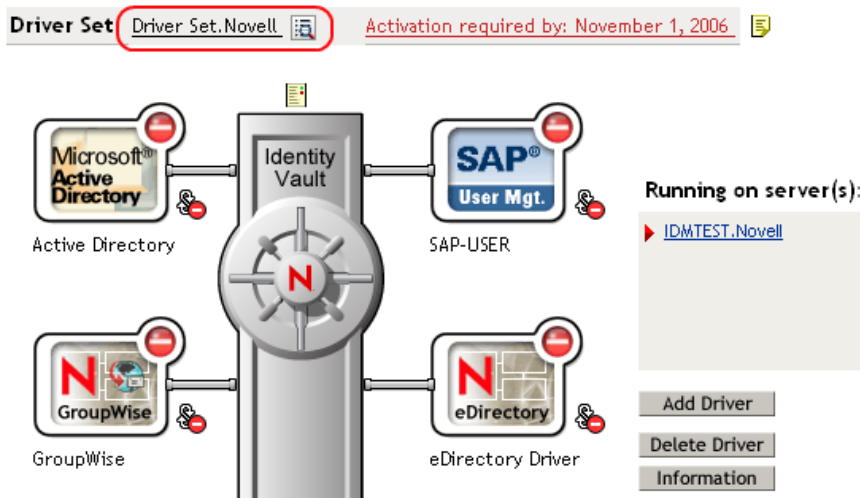
## 9.1.2 Adding Trace Levels in iManager

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

- ♦ “Driver Set” on page 57
- ♦ “Driver” on page 57

### 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 94 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 94 for the parameters.

---

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

---

## 9.1.3 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 58
- ♦ “Windows” on page 58
- ♦ “UNIX” on page 59
- ♦ “iMonitor” on page 59
- ♦ “Remote Loader” on page 60

## NetWare

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

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

## Windows

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

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

## UNIX

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

## iMonitor

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

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



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

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 10.1, “Exporting the Driver in Designer,” on page 63](#)
- ♦ [Section 10.2, “Exporting the Driver in iManager,” on page 63](#)

## 10.1 Exporting the Driver in Designer

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

## 10.2 Exporting the Driver in iManager

- 1 In iManager, select *Identity Manager > Identity Manager Overview*.
- 2 Browse to and select the driver set object, then click *Search*.
- 3 Click the driver icon.
- 4 Select *Export* in the Identity Manager Driver Overview window.
- 5 Browse to and select the driver object 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

# 11

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

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:

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

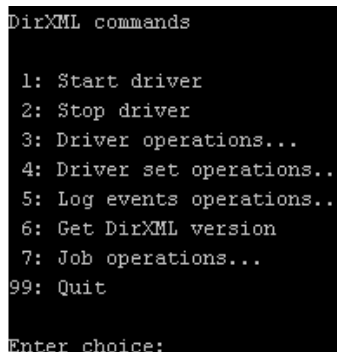
There are two different methods for using the DirXML Command Line utility:

- ♦ [Section A.1, “Interactive Mode,” on page 67](#)
- ♦ [Section A.2, “Command Line Mode,” on page 76](#)

## 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 68](#) contains the list of options and what functionality is available.
- 5 Enter 99 to quit the utility.

---

**NOTE:** If you are running eDirectory<sup>™</sup> 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 69</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 73</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.
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>
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/ndsstd/query.html">Novell nds.dtd (http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/ref/ndsstd/query.html)</a>.</p> <p>Examples:</p> <p>NetWare: <code>sys:\files\query.xml</code></p> <p>Windows: <code>c:\files\query.xml</code></p> <p>Linux: <code>/files/query.xml</code></p>

Options	Description
8: <i>Submit XDS command document to driver</i>	<p>Processes an XDS command document:</p> <p><i>Enter filename of XDS command document:</i></p> <p>Examples:</p> <p>NetWare: 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>
10: <i>Queue event for driver</i>	<p>Adds an event to the driver queue:</p> <p><i>Enter filename of XDS event document:</i></p> <p>Examples:</p> <p>NetWare: sys:\files\add.xml</p> <p>Windows: c:\files\add.xml</p> <p>Linux: /files/add.xml</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 71</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 72</a> for a description of these options.</p>

Options	Description
99: <i>Exit</i>	Exits the driver options.

**Figure A-2** Password Operations

```
Select a password operation

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

Enter choice:
```

**Table A-3** Password Operations

Operation	Description
1: <i>Set shim password</i>	Sets the application password. This is the password of the user account you are using to authenticate into the connected system with.
2: <i>Clear shim password</i>	Clears the application password.
3: <i>Set Remote Loader password</i>	The Remote Loader password is used to control access to the Remote Loader instance.  Enter the Remote Loader password, then confirm the password by typing it again.
4: <i>Clear Remote Loader password</i>	Clears the Remote Loader password so no Remote Loader password is set on the Driver object.
5: <i>Set named password</i>	Allows you to store a password or other pieces of security information on the driver. See <a href="#">Section 8.6, "Storing Driver Passwords Securely with Named Passwords,"</a> on page 46 for more information.  There are four prompts to fill in: <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>

Operation	Description
6: <i>Clear named passwords</i>	<p>Clears a specified Named Password or all named passwords that are stored on the driver object: <i>Do you want to clear all named passwords? (yes/no)</i>.</p> <p>If you enter Yes, all Named Passwords are cleared. If you enter No, you are prompted to specify the password name that you want to clear.</p>
7: <i>List named passwords</i>	Lists all named passwords that are stored on the driver object. It lists the password name and the password description.
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 dxcmd utility allows you to set the Application password and the Remote Loader password. You cannot set the Driver Object password with this utility. It shows if the password has been set or not.</p>
99: <i>Exit</i>	Exits the current menu and takes you back to the Driver options.

**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.



Operation	Description
3: <i>View cached transactions</i>	<p>A text file is created with the events that are stored in cache. You can select the number of transactions to view.</p> <ul style="list-style-type: none"> <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>	<p>Deletes the transactions stored in cache.</p> <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>	<p>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 74</a> for a list of these options.</p> <p>Type the number of the item you want to log. After the items are selected, enter 99 to accept the selections.</p>
2: <i>Reset driver set log events</i>	Resets all of the log event options.
3: <i>Set driver log events</i>	<p>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 74</a> for a list of these options.</p> <p>Type the number of the item you want to log. After the items are selected, enter 99 to accept the selections.</p>

Operation	Description
4: <i>Reset driver log events</i>	Resets all of the log event options.
99: <i>Exit</i>	Exits the log events operations menu.

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

Options
1: Status success
2: Status retry
3: Status warning
4: Status error
5: Status fatal
6: Status other
7: Query elements
8: Add elements
9: Remove elements
10: Modify elements
11: Rename elements
12: Move elements
13: Add-association elements
14: Remove-association elements
15: Query-schema elements
16: Check-password elements
17: Check-object-password elements
18: Modify-password elements
19: Sync elements
20: Pre-transformed XDS document from shim
21: Post input transformation XDS document
22: Post output transformation XDS document
23: Post event transformation XDS document
24: Post placement transformation XDS document
25: Post create transformation XDS document
26: Post mapping transformation <inbound> XDS document
27: Post mapping transformation <outbound> XDS document

---

**Options**

---

28: Post matching transformation XDS document

29: Post command transformation XDS document

30: Post-filtered XDS document <Publisher>

31: User agent XDS command document

32: Driver resync request

33: Driver migrate from application

34: Driver start

35: Driver stop

36: Password sync

37: Password request

38: Engine error

39: Engine warning

40: Add attribute

41: Clear attribute

42: Add value

43: Remove value

44: Merge entire

45: Get named password

46: Reset Attributes

47: Add Value - Add Entry

48: Set SSO Credential

49: Clear SSO Credential

50: Set SSO Passphrase

51: User defined IDs

99: Accept checked items

---

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

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

## A.2 Command Line Mode

The command line mode allows you to use script or batch files. [Table A-8 on page 76](#) 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.
-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.

Option	Description
-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 74</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 79](#) contains other values for specific command line options.

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

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

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



# Schema For Work Order Management

As part of the installation of the Novell® Identity Manager WorkOrder driver, eDirectory™ is extended to include two new object classes. These objects allow the driver to connect to the Identity Vault correctly, perform work orders, and create a process log with the work order status.

Installing Identity Manager 3.5 provides iManager plug-ins to help you create or view these objects in the WorkOrder driver. See [Chapter 5, “Customizing the Driver,”](#) on page 25.

The Publisher Placement rule, and the Subscriber Placement and Create rules are described in the following sections:

- ♦ [Section B.1, “DirXML-WorkOrder Object,”](#) on page 81
- ♦ [Section B.2, “DirXML-WorkToDo Object,”](#) on page 83
- ♦ [Section B.3, “Publisher Placement Rule,”](#) on page 84
- ♦ [Section B.4, “Subscriber Placement Rule,”](#) on page 84
- ♦ [Section B.5, “Subscriber Create Rule,”](#) on page 84

## B.1 DirXML-WorkOrder Object

The DirXML-WorkOrder object (sometimes referred to as the WorkOrder object in this manual) is used to tell the driver what tasks to perform. It delays the work order until a date and time or until another work order is configured. It also repeats work orders at a given interval.

The following table shows the work order attributes you need to specify:

**Table B-1** *WorkOrder Object Attributes*

Work Order Attributes (eDirectory Namespace)	Description	Type
Description	Description of the work order. The driver does not change this attribute. It is passed through to the WorkToDo object when the work order is processed.	Case ignore string
Common Name	The naming attribute for eDirectory™	Case ignore string
DirXML-nwoContact Name	Information about the work order. The driver does not change this attribute. It is passed through to the WorkToDo object when the work order is processed.	Case ignore string
DirXML-nwoContent	This attribute is passed through to the WorkToDo object. It is used by policies to process the work order.	Case ignore string
DirXML-DueDate	The date and time the work order is to be processed.	Time

Work Order Attributes (eDirectory Namespace)	Description	Type
DirXML-nwoDoltNowFlag	If set to True, the subscriber channel sends the work order to the Publisher channel to be processed immediately.	Boolean
DirXML-nwoSendToPublisher	If set to True, the Subscriber channel sends the work order to the Publisher channel to be written to the WorkOrder container. For example, if the work order was created by a policy as a result of an event in the Identity Vault.	Boolean
DirXML-woType	Information about the work order. The driver does not change this attribute. It is passed through to the WorkToDo object when the work order is processed.	User defined
DirXML-nwoCreationDate	Information about the work order. The driver does not change this attribute.	Time
DirXML-nwoDependentWorkOrder	The DN of the dependent work order. The work order is not processed until the dependent work order has a status of Configured. If the attribute is non-existent or empty, it is ignored.	Distinguished Name
DirXML-nwoRepeatInterval	The amount of time, in hours, before the work order is repeated. This value is added to the due date after the work order is processed.	Case ignore string
DirXML-nwoStatus	Status of the work order.  Pending: the work order will be processed on the due date.  Configured: the work order was processed.  Error: An error occurred when processing.  On Hold: the work order is not to be processed.	Case ignore string
DirXML-nwoWorkOrderNumber	Information about the work order. The driver does not change this attribute. It is passed through to the WorkToDo object when the work order is processed.	Case ignore string
DirXML-nwoDeleteOnError	If set to True, the work order is deleted if the status is Error and the DeleteDueDate has expired.	Boolean
DirXML-nwoProcessLog	Contains information relating to the processing of the work order.	Case ignore string
DirXML-nwoDeleteDueDate	The date and time the work order will be deleted. If the status is Pending or Configured.	Time

Work Order Attributes (eDirectory Namespace)	Description	Type
DirXML-Creator Name	Information about the work order. The driver does not change this attribute. It is passed through to the WorkToDo object when the work order is processed.	Case ignore string
DirXML-Other1	Information about the work order. The driver does not change this attribute. It is passed through to the WorkToDo object when the work order is processed.	Case ignore string
DirXML-Other2	Information about the work order. The driver does not change this attribute. It is passed through to the WorkToDo object when the work order is processed.	Case ignore string

## B.2 DirXML-WorkToDo Object

The DirXML-WorkToDo object is created by the driver to attempt processing. It is used by the policy to process the work to be done. All attributes in this object get their values from the work order object that initiated this object.

**Table B-2** *DirXML-WorkToDo Object Attributes*

WorkToDo Attributes	Description	Type
DirXML-CreatorName	Information about the work order. The driver does not change this attribute.	Case ignore string
DirXML-nwoContent	The value of the content attribute in the work order.	Case ignore string
DirXML-nwoDN	DN of the work order.	Distinguished Name
Description	Information about the work order. The driver does not change this attribute.	Case ignore string
DirXML-nwoContactName	Information about the work order. The driver does not change this attribute.	Case ignore string
DirXML-nwoWorkOrderNumber	Information about the work order. The driver does not change this attribute.	Case ignore string
DirXML-woType	Information about the work order. The driver does not change this attribute.	Case ignore string
DirXML-Other1	Information about the work order. The driver does not change this attribute.	Case ignore string
DirXML-Other2	Information about the work order. The driver does not change this attribute.	Case ignore string

## B.3 Publisher Placement Rule

The Publisher Placement rule determines where the work orders are placed in the Identity Vault after they are processed. These containers might be the same or different, depending on how you choose to set up your customized driver. For example, you could have work orders stored in containers depending on the returned status, such as configured, error, warning or on hold.

## B.4 Subscriber Placement Rule

The Subscriber Placement rule determines which container work orders will be created in and sent to the WorkOrder driver.

## B.5 Subscriber Create Rule


To create a work order, the Subscriber Create rule is set up so all new work orders with the necessary information can be sent to the Subscriber channel. The following attributes must be present to pass the Create rule and go to the Subscriber channel:

**Table B-3** *Work Order Attributes for the Subscriber Create Rule*

Required Attributes	Description	Values or Examples
DirXML-nwoSendToPublisher	Send the work order to directly to the Publisher channel.	True or False
DirXML-nwoStatus	State of the work order so the driver knows what to do with the work order.	Pending, Configured, Error, on Hold, Warning
DirXML-nwoDoItNowFlag	When to perform the work order.	True or False
DirXML-nwoContent	Content to be processed by the driver.	XML code

# Properties of the Driver

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 C.1, “Driver Configuration,” on page 85](#)
- ♦ [Section C.2, “Global Configuration Values,” on page 89](#)
- ♦ [Section C.3, “Named Passwords,” on page 90](#)
- ♦ [Section C.4, “Engine Control Values,” on page 90](#)
- ♦ [Section C.5, “Log Level,” on page 92](#)
- ♦ [Section C.6, “Driver Image,” on page 93](#)
- ♦ [Section C.7, “Security Equals,” on page 93](#)
- ♦ [Section C.8, “Filter,” on page 94](#)
- ♦ [Section C.9, “Edit Filter XML,” on page 94](#)
- ♦ [Section C.10, “Misc,” on page 94](#)
- ♦ [Section C.11, “Excluded Users,” on page 95](#)
- ♦ [Section C.12, “Driver Manifest,” on page 95](#)
- ♦ [Section C.13, “Inspector,” on page 96](#)
- ♦ [Section C.14, “Server Variables,” on page 96](#)

## C.1 Driver Configuration

In iManager:

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

In Designer:

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

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

### C.1.1 Driver Module



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

In iManager:

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

In Designer:

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

Option	Description
<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>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.

## C.1.2 Driver Object Password

In iManager:

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

In Designer:

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

Option	Description
<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.

## C.1.3 Authentication










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


In iManager:

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

In Designer:

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

Option	Description
<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 <code>hostname=xxx.xxx.xxx.xxx port=xxxx</code> <code>kmo=certificatename</code> , when the host name is the IP address of the application server running the Remote Loader server and the port is the port the remote loader is listening on. The default port for the Remote Loader is 8090.  The <code>kmo</code> entry is optional. It is only used when there is an SSL connection between the Remote Loader and the Metadirectory engine.  Example: <code>hostname=10.0.0.1 port=8090</code> <code>kmo=IDMCertificate</code>
<i>Driver Cache Limit (kilobytes)</i> or  <i>Cache limit (KB)</i>	Specify the maximum event cache file size (in KB). If it is set to zero, the file size is unlimited.   Click <i>Unlimited</i> to set the file size to unlimited in Designer.
<i>Application Password</i> or  <i>Set Password</i>	Specify the password for the user object listed in the <i>Authentication ID</i> field.

Option	Description
<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>	

## C.1.4 Startup Option


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

In iManager:

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

In Designer:

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

Option	Description
<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.

## C.1.5 Driver Parameters

In iManager:

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

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 Name	Parameter Descriptions
Driver Name	The actual name you want to use for the driver.
WorkOrders Container	The name of the container where work orders are to be stored.
WorkToDo Container	The name of the container to store configured work orders.
Polling Method	Specifies the polling method by interval or time. <i>Interval</i> indicates that the driver will poll at a specified time interval. <i>Polling by time</i> indicates a specific time of day.
Driver Heartbeat	Specifies if the Publisher should emit heartbeat documents. The driver emits heartbeat documents to indicate to the Identity Manager that the driver is still functioning.
Install Driver as Remote or Local	Select <i>Remote</i> to configure the driver for use with the Remote Loader service.  Select <i>Local</i> to configure the driver for local use.
Poll Interval	The polling interval (in minutes) at which the Publisher channel polls the Identity Manager Vault for work orders to be configured.
Poll Time	Time of day the Publisher channel wakes up to check the Identity Manager Vault for work orders to be configured.

## C.2 Global Configuration Values

Global configuration values (GCVs) allow you to specify settings for the Identity Manager features such as 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.

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*.

## C.3 Named Passwords

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

You can also use the Named Passwords feature to store other pieces of information securely, such as a user name. To configured Named Passwords, see [Section 8.6, “Storing Driver Passwords Securely with Named Passwords,” on page 46](#).

In iManager:

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

In Designer:

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

## C.4 Engine Control Values

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

In iManager:

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

This option does not exist in Designer at this time.

**Table C-1** *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.

Option	Description
<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 backward-compatible mode. The backward-compatible mode causes the XSLT processor to use one or more behaviors that are not XPath 1.0 and XSLT 1.0 standards-compliant. This is done in the interest of backward-compatibility with existing DirXML® style sheets that depend on the non-standard behaviors.</p> <p>For example, the behavior of the XPath “!=” operator when one operand is a node-set and the other operand is other than a node-set is incorrect in DirXML releases up to and including Identity Manager 2.0. This behavior has been corrected; however, the corrected behavior is disabled by default through this control in favor of backward-compatibility with existing DirXML style sheets.</p>
<i>Maximum application objects to migrate at once</i>	<p>This control is used to limit the number of application objects that the Metadirectory engine requests from an application during a single query that is performed as part of a Migrate Objects from Application operation.</p> <p>If java.lang.OutOfMemoryError errors are encountered during a Migrate from Application operation, this number should be set lower than the default. The default is 50.</p> <hr/> <p><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>

Option	Description
<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>

## C.5 Log Level

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

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

Option	Description
<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.

## C.6 Driver Image

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

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

---

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

---

In iManager:

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

In Designer:

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

## C.7 Security Equals

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

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

In iManager:

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

- 3 Click *Edit Properties > Security Equals*.

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

## C.8 Filter

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

In iManager:

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

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

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

## C.9 Edit Filter XML

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

In iManager:

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


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

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

## C.10 Misc

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

**Table C-2** *Trace File Settings*

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.

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*.

## C.11 Excluded Users

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

In iManager:

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

Designer does not list the excluded users.

## C.12 Driver Manifest

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

In iManager:

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

In Designer:

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

## C.13 Inspector

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

## C.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 that password that 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 <code>&lt;password&gt;</code> 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 <code>&lt;password&gt;</code> value coming from the connected system is written directly to the Universal Password in the Identity Vault if it is enabled for the user. If the user's password policy does not enable Universal Password, the password is written to the NDS Password.</p>
<i>Use Distribution Password for password synchronization</i>	<p>To use this setting, you must have a version of eDirectory that supports Universal Password, regardless of whether you have enabled Universal Password in your password policies.</p> <p>If this option is enabled, a password value coming from the connected system is written to the Distribution Password. The Distribution Password is reversible, which means that it can be retrieved from the Identity Vault data store for password synchronization. It is used by Identity Manager for bidirectional password synchronization with connected systems. For Identity Manager to distribute passwords to connected systems, this option must be enabled.</p> <p>NMAS and Password policies control whether the Distribution Password is synchronized with other passwords in the Identity Vault. By default, the Distribution Password is the same as the Universal Password in the Identity Vault.</p> <p>If the password in the Identity Vault is to be independent of Password Synchronization, so that Identity Manager is a conduit only for synchronizing passwords among connected systems, change this default setting. In the Universal Password Configuration Options in a Password policy, disable <i>Synchronize Universal Password with Distribution Password</i>. This use of Identity Manager Password Synchronization is also referred to as "tunneling."</p>
<i>Accept password only if it complies with user's Password Policy</i>	<p>To use this setting, users must have a Password policy assigned that has Universal Password enabled, and Advanced Password Rules enabled and configured.</p> <p>If this option is chosen, Identity Manager does not write a password from this connected system to the Distribution Password in the Identity Manager data store or publish it to connected systems unless the password complies with the user's Password policy.</p> <p>By using the notification option that is also on this page, you can inform users when a password is not set because it is not compliant.</p>

Option	Description
<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>
<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>
<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>
<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>

# Documentation Update

# D

The documentation was updated on the following date:

- ♦ Section D.1, “July 30, 2007,” on page 99
- ♦ Section D.2, “June 29, 2007,” on page 99
- ♦ Section D.3, “May 17, 2007,” on page 99
- ♦ Section D.4, “April 11, 2007,” on page 100

## D.1 July 30, 2007

The following section was updated:

### D.1.1 Schema for Work Order Management

Location	Change
“DirXML-DueDate” on page 81	Changed the sched from DirXML-nwoDueDate to DirXML-DueDate.

## D.2 June 29, 2007

The following section was updated:

### D.2.1 Understanding Driver Architecture

Location	Change
Section 3.2, “Subscriber Channel Functions,” on page 17	Added the last two paragraphs on page 18.

## D.3 May 17, 2007

The following section was updated:

### D.3.1 Schema for Work Order Management

Location	Change
Table B-2 on page 83	Changed the attribute from DirXML-nwoType to DirXML-woType.

## D.4 April 11, 2007

The following section was updated:

### D.4.1 Additional Solutions

Location	Change
"Human Resource Driver Policies" on page 27	Fixed the broken link to the sample files <code>hr-drv-cmv-transform.xml</code> and <code>wo-sub-smd-transform.xml</code> .