

## WorkOrder Driver Implementation Guide

# Novell<sup>®</sup> Identity Manager

**3.6.1**

June 05, 2009

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



## 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 © 2008-2010 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>7</b>
<b>1 Overview</b>	<b>9</b>
1.1 The Work Order Process	9
1.1.1 Subscriber Channel Functions	9
1.1.2 Publisher Channel Functions	11
1.2 Key Features	14
1.3 Support for Standard Driver Functions	14
1.4 Terminology	14
<b>2 Implementation Checklist</b>	<b>15</b>
<b>3 Installing Driver Files</b>	<b>17</b>
<b>4 Creating a New Driver</b>	<b>19</b>
4.1 Creating the WorkOrder Container in the Identity Vault	19
4.2 Creating the Driver in Designer	19
4.2.1 Importing the Driver Configuration File	19
4.2.2 Configuring the Driver Settings	20
4.2.3 Deploying the Driver	21
4.2.4 Starting the Driver	21
4.3 Creating the Driver in iManager	22
4.3.1 Importing the Driver Configuration File	22
4.3.2 Configuring the Driver Settings	24
4.3.3 Starting the Driver	24
4.4 Activating the Driver	25
<b>5 Upgrading an Existing Driver</b>	<b>27</b>
5.1 Supported Upgrade Paths	27
5.2 What's New in Version 3.6.1	27
5.3 Upgrade Procedure	27
<b>6 Customizing the Driver</b>	<b>29</b>
6.1 Policies and Rules Used in the Basic Configuration	29
6.1.1 Subscriber Channel	29
6.1.2 Publisher Channel	30
6.2 Human Resource Example Using an HR Driver	31
6.2.1 Human Resource Driver Policies	32
6.2.2 WorkOrder Driver Policy	33
6.3 Human Resource Example without an HR Driver	33
6.3.1 Filter Additions	34
6.3.2 Subscriber Create Rule	34
6.3.3 Subscriber Command Transform	34
6.3.4 Work Order E-Mail Notification of Work Order Completion	34

<b>7</b>	<b>Creating and Managing Work Orders</b>	<b>35</b>
7.1	Using Drivers to Create Work Orders . . . . .	35
7.2	Using iManager . . . . .	35
7.2.1	Creating a New Work Order . . . . .	35
7.2.2	Editing Work Order Properties . . . . .	35
7.2.3	Filtering the Work Order List . . . . .	37
<b>8</b>	<b>Managing the Driver</b>	<b>39</b>
<b>9</b>	<b>Troubleshooting Driver Processes</b>	<b>41</b>
<b>A</b>	<b>Driver Properties</b>	<b>43</b>
A.1	Driver Configuration . . . . .	43
A.1.1	Driver Module . . . . .	44
A.1.2	Driver Object Password (iManager Only) . . . . .	44
A.1.3	Authentication . . . . .	44
A.1.4	Startup Option . . . . .	45
A.1.5	Driver Parameters . . . . .	46
A.1.6	ECMAScript (Designer Only) . . . . .	47
A.2	Global Configuration Values . . . . .	47
<b>B</b>	<b>Objects and Attributes Used</b>	<b>49</b>
B.1	New Objects Used by the Driver . . . . .	49
B.1.1	DirXML-WorkOrder Object . . . . .	49
B.1.2	DirXML-WorkToDo Object . . . . .	49
B.2	DoltNow and SendToPublisher Flags . . . . .	49
B.2.1	DoltNow Flag . . . . .	50
B.2.2	SendToPublisherFlag . . . . .	50
<b>C</b>	<b>Schema and Policy Rules For Work Order Management</b>	<b>51</b>
C.1	DirXML-WorkOrder Object . . . . .	51
C.2	DirXML-WorkToDo Object . . . . .	53
C.3	Publisher Placement Rule . . . . .	54
C.4	Subscriber Placement Rule . . . . .	54
C.5	Subscriber Create Rule . . . . .	54

# About This Guide

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

- ◆ Chapter 1, “Overview,” on page 9
- ◆ Chapter 2, “Implementation Checklist,” on page 15
- ◆ Chapter 3, “Installing Driver Files,” on page 17
- ◆ Chapter 4, “Creating a New Driver,” on page 19
- ◆ Chapter 5, “Upgrading an Existing Driver,” on page 27
- ◆ Chapter 6, “Customizing the Driver,” on page 29
- ◆ Chapter 7, “Creating and Managing Work Orders,” on page 35
- ◆ Chapter 8, “Managing the Driver,” on page 39
- ◆ Chapter 9, “Troubleshooting Driver Processes,” on page 41
- ◆ Appendix A, “Driver Properties,” on page 43
- ◆ Appendix B, “Objects and Attributes Used,” on page 49
- ◆ Appendix C, “Schema and Policy Rules For Work Order Management,” on page 51

## 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 WorkOrder Driver Implementation Guide*, visit the [Identity Manager Documentation Web site \(http://www.novell.com/documentation/idm36drivers\)](http://www.novell.com/documentation/idm36drivers).

## Additional Documentation

For documentation on other Identity Manager drivers, see the [Identity Manager Documentation Web site \(http://www.novell.com/documentation/idm36drivers\)](http://www.novell.com/documentation/idm36drivers).

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



# Overview

# 1

Typically, changes to data in the Identity Vault or a connected application are immediately processed. Work orders enable you to schedule when tasks are to be performed.

For example, a new employee is hired but is not scheduled to start for a month. The employee needs to be added to the HR database but should not be granted access to any corporate resources (e-mail, servers, and so forth) until the start date. In a typical scenario, the user would be granted access as soon as he or she is added to the HR database. With work orders implemented, a work order is created that delays account provisioning to the user's start date.

The WorkOrder driver provides work order functionality. The following sections introduce the WorkOrder driver and the key concepts and terminology associated with the driver:

- ♦ [Section 1.1, “The Work Order Process,” on page 9](#)
- ♦ [Section 1.2, “Key Features,” on page 14](#)
- ♦ [Section 1.3, “Support for Standard Driver Functions,” on page 14](#)
- ♦ [Section 1.4, “Terminology,” on page 14](#)

## 1.1 The Work Order Process

From a high-level perspective, work orders are processed as follows:

1. A work order is created, either through an automated process (another driver) or a manual process (iManager), and is added as a WorkOrder object in a the Identity Vault's work order container.
2. At the scheduled time (as defined in the WorkOrder object), the driver begins processing the work order.
3. The driver applies any policies to the work order (performing any actions associated with the policies) and creates a WorkToDo object in the Identity Vault's work order container.
4. Depending on how you configure the WorkOrder driver and the other drivers in your system, either the WorkOrder driver performs the desired work or other drivers use the information in the the WorkToDo object to perform the work. Because the WorkOrder driver is designed to accommodate a variety of configuration scenarios, sample scenarios are provided in [Chapter 6, “Customizing the Driver,” on page 29](#).

The following sections provide detailed information about the work performed by the driver's Subscriber and Publisher channels. Because the WorkOrder driver channels function differently than with other drivers, you should carefully review the information.

- ♦ [Section 1.1.1, “Subscriber Channel Functions,” on page 9](#)
- ♦ [Section 1.1.2, “Publisher Channel Functions,” on page 11](#)

### 1.1.1 Subscriber Channel Functions

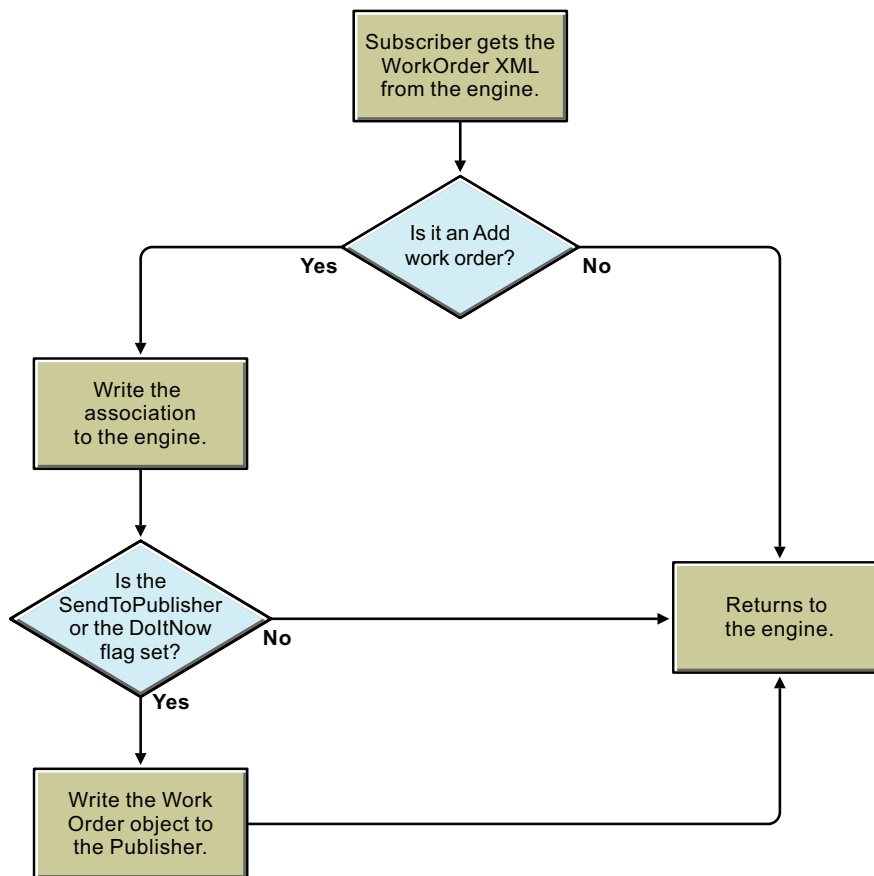
This section provides a basic understanding of the functions the Subscriber channel performs in the WorkOrder driver.

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-woContent
- ◆ DirXML-woStatus
- ◆ DirXML-DoItNow Flag
- ◆ DirXML-SendToPublisher Flag

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

**Figure 1-1** 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.

## 1.1.2 Publisher Channel Functions

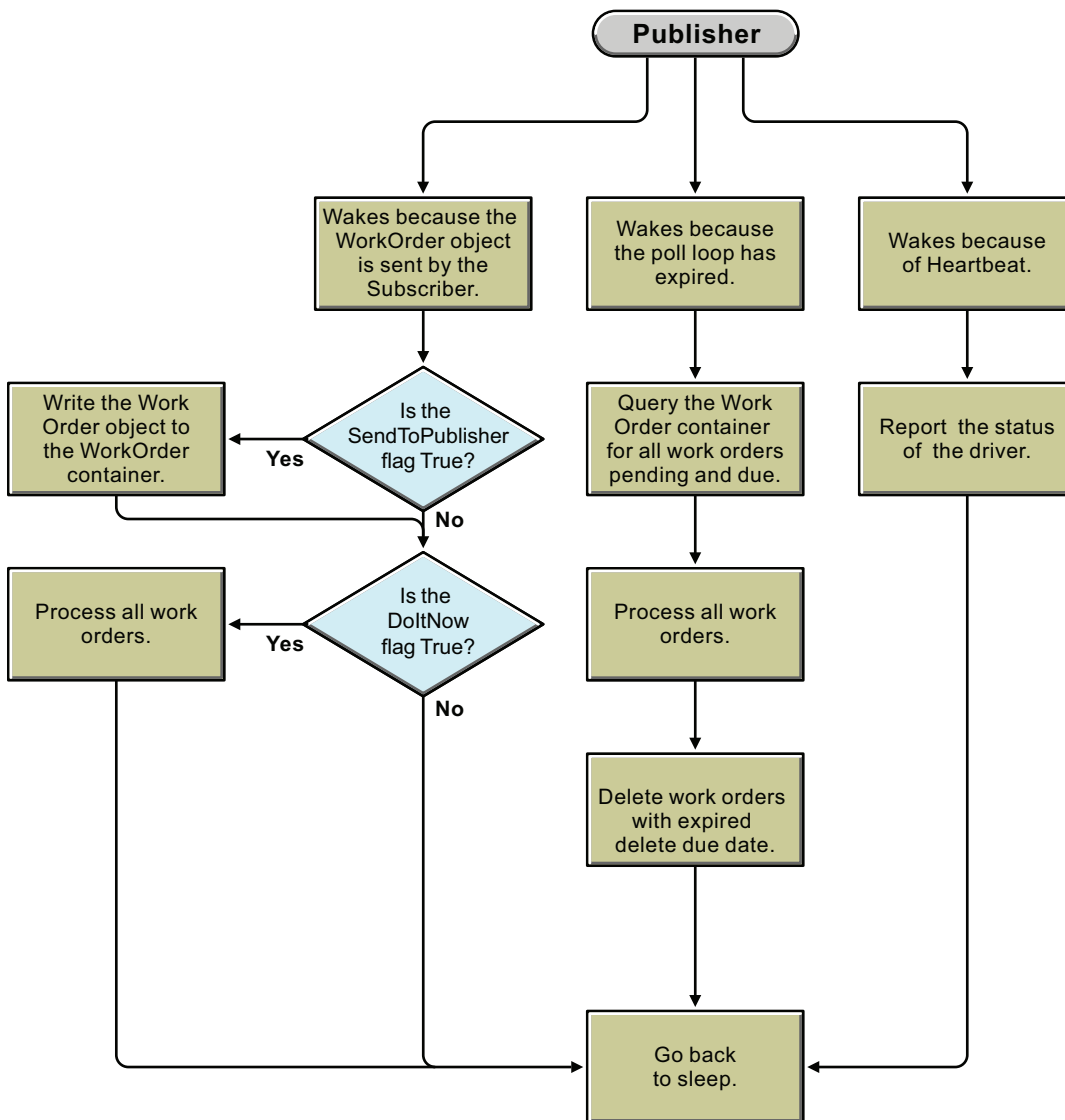
This section reviews the functions of the Publisher channel.

- ♦ “The Publisher Channel Wakes Up” on page 11
- ♦ “How the Publisher Channel Processes Work Orders” on page 12
- ♦ “How the Publisher Channel Deletes Work Orders” on page 13

### The Publisher Channel Wakes Up

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

**Figure 1-2** Publisher Channel Configuration



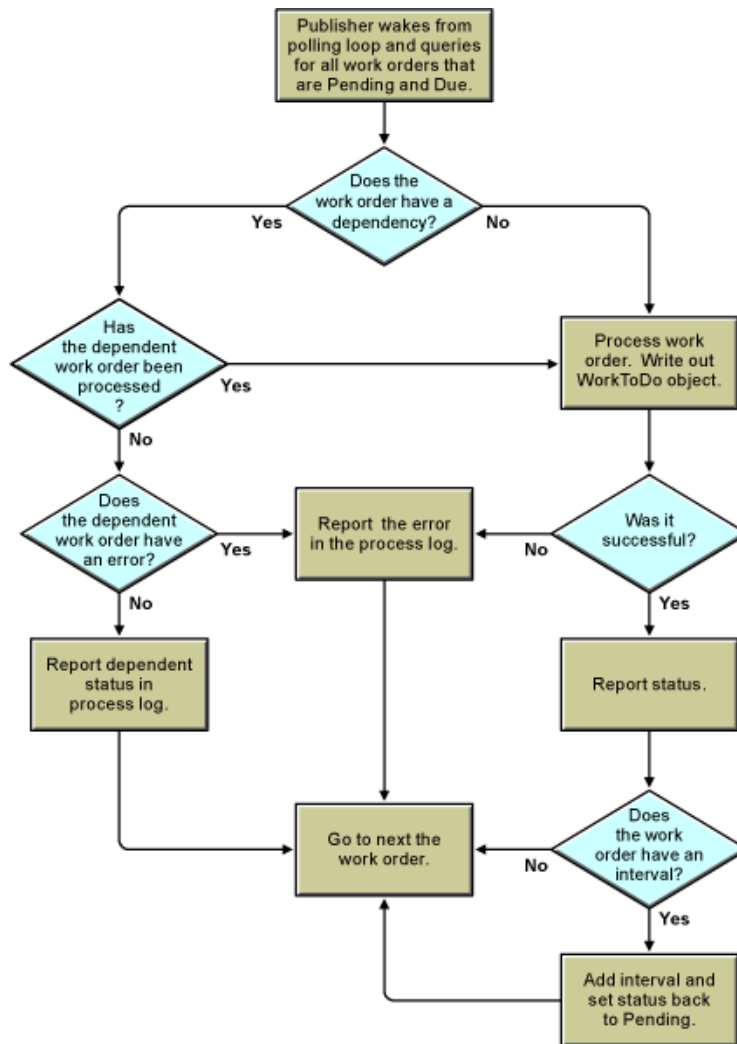
1. The Publisher channel wakes because the Subscriber channel sends a WorkOrder object. If the SendToPublisher flag is set to True, the work order is written out to the work order container. If the DoItNow flag is set to True, the work order is processed immediately.

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 [“How the Publisher Channel Processes Work Orders”](#) on page 12.
  - b. The Publisher channel queries all work orders for expired DeleteDueDates. See [“How the Publisher Channel Deletes Work Orders”](#) on page 13.
3. If the driver heartbeat is configured, the driver wakes to report the driver status.

### 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 1-3** *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

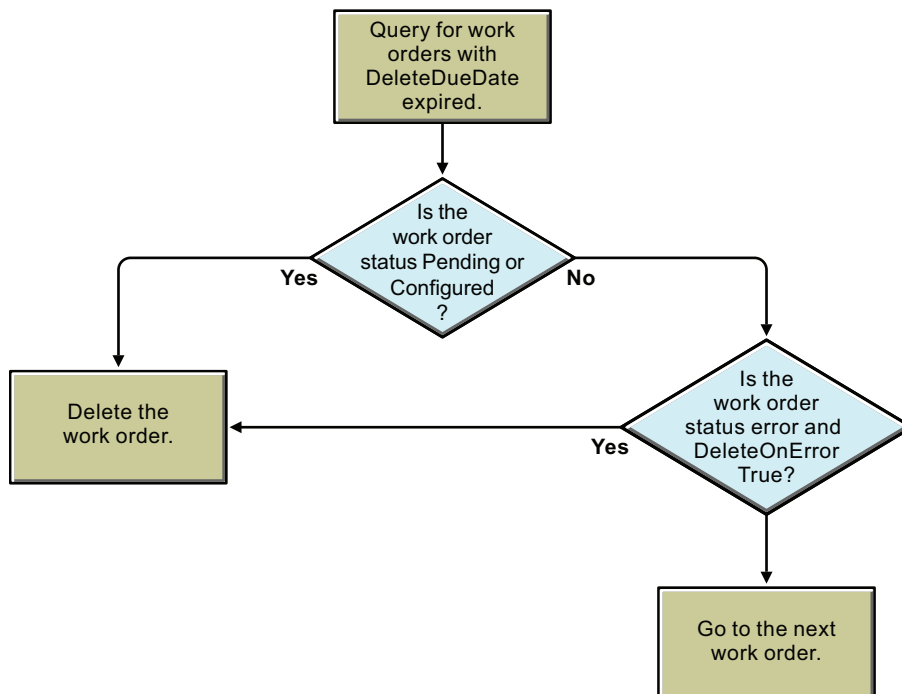
dependent work order status is configured, the Publisher channel processes the work order. If not, the work order waits until the next polling loop to see if the dependent work order has been configured.

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-woContent attribute of the WorkToDo object contains the value of the DirXML-woContent 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 is 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 as many times as specified in the repeat interval count value, or indefinitely if no repeat interval count value is specified. The process log contains the results.

### 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 1-4 The DeleteDueDate Process



## 1.2 Key Features

The following list describes key features of the WorkOrder driver:

- ♦ **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 successfully processed.
- ♦ **Repeats work orders:** The driver allows for work orders to be repeated at a set interval.
- ♦ **Facilitates tracking and accountability for 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.

## 1.3 Support for Standard Driver Functions

There are several functions that most Identity Manager drivers support. The following list explains whether or not the WorkOrder driver supports these standard functions.

- ♦ **Local Platforms:** The WorkOrder driver can run on the same platforms as the Metadirectory engine. See “[Metadirectory Server](#)” in “[System Requirements](#)” in the *Identity Manager 3.6.1 Installation Guide*.
- ♦ **Remote Platforms:** The WorkOrder driver works on all the platforms supported by the Remote Loader. See “[Remote Loader](#)” in “[System Requirements](#)” in the *Identity Manager 3.6.1 Installation Guide*.
- ♦ **Role-Based Entitlements:** The WorkOrder driver does not support Role-Based Entitlements.
- ♦ **Password Synchronization Support:** The WorkOrder driver does not support Password Synchronization.
- ♦ **Synchronized Objects:** The WorkOrder driver only processes events that pertain to work orders. It does not synchronize other objects and attributes within the Identity Vault.

## 1.4 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 any other work orders that must be completed before the current work order.
- ♦ **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.
- ♦ **Delete Due Date:** The date the work order will be deleted from the Identity Vault.
- ♦ **Pending:** A work order that is not yet due.

# Implementation Checklist

# 2

Use the following checklist to ensure that you complete all of the tasks required to set up and use the WorkOrder driver.

**Table 2-1** *WorkOrder Implementation Checklist*

Task	Details
<input type="checkbox"/> Install the WorkOrder driver files	By default, the WorkOrder driver files (driver shim and configuration file) are copied to the Metadirectory server when the Metadirectory engine is installed. If the driver is not on the Metadirectory server (because a custom installation was performed, or you want to run the driver on a server other than the Metadirectory server), see <a href="#">Chapter 3, “Installing Driver Files,” on page 17</a> .
<input type="checkbox"/> Create a new WorkOrder driver  or  <input type="checkbox"/> Upgrade an existing WorkOrder driver to the new version	You need to import the basic configuration file to create the driver. For instructions, see <a href="#">Chapter 4, “Creating a New Driver,” on page 19</a> .  If you have an existing driver, you can upgrade its configuration to this version. For instructions, see <a href="#">Chapter 5, “Upgrading an Existing Driver,” on page 27</a> .
<input type="checkbox"/> Customize the driver	The basic configuration for the WorkOrder driver enables it to create WorkOrder objects and WorkToDo objects. This is the extent of what the WorkOrder driver does when using the base configuration. For any additional work to be done, you must customize the WorkOrder driver or other Identity Manager drivers to perform the desired work.  For instructions, see <a href="#">Chapter 6, “Customizing the Driver,” on page 29</a> .
<input type="checkbox"/> Create work orders	Most work orders are likely created by other drivers as part of the work order process you establish while customizing the driver (see the previous task). However, you can also create work orders manually as well as modify existing work orders.  For instructions, see <a href="#">Chapter 7, “Creating and Managing Work Orders,” on page 35</a> .





# Installing Driver Files

# 3

By default, the WorkOrder driver files are installed on the Metadirectory server at the same time as the Metadirectory engine. The installation program extends the Identity Vault's schema and installs both the driver shim and the driver configuration files. It does not create the driver in the Identity Vault (see [Chapter 4, "Creating a New Driver," on page 19](#)) or upgrade an existing driver's configuration (see [Chapter 5, "Upgrading an Existing Driver," on page 27](#)).

If you performed a custom installation and did not install the WorkOrder driver on the Metadirectory server, you have two options:

- ◆ Install the files on the Metadirectory server, using the instructions in "[Installing the Metadirectory Server](#)" in the *Identity Manager 3.6.1 Installation Guide*.
- ◆ Install the Remote Loader (required to run the driver on a non-Metadirectory server) and the driver files on a non-Metadirectory server where you want to run the driver. See "[Installing the Remote Loader](#)" in the *Identity Manager 3.6.1 Installation Guide*.



# Creating a New Driver

# 4

After the WorkOrder driver files are installed on the server where you want to run the driver (see [Chapter 3, “Installing Driver Files,” on page 17](#)), you can create the driver in the Identity Vault. You do so by importing the basic driver configuration file and then modifying the driver configuration to suit your environment. The following sections provide instructions:

- ♦ [Section 4.1, “Creating the WorkOrder Container in the Identity Vault,” on page 19](#)
- ♦ [Section 4.2, “Creating the Driver in Designer,” on page 19](#)
- ♦ [Section 4.3, “Creating the Driver in iManager,” on page 22](#)
- ♦ [Section 4.4, “Activating the Driver,” on page 25](#)

## 4.1 Creating the WorkOrder Container in the Identity Vault

The WorkOrder driver requires you to specify an Identity Vault container for WorkOrder objects and WorkToDo objects. You can use an existing container for these objects, but we strongly recommend that you create a new container. You can give the container any name you want (for example, WorkOrders). You should restrict rights to the container so that only authorized administrators can change the container or the objects it holds.

## 4.2 Creating the Driver in Designer

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

- ♦ [Section 4.2.1, “Importing the Driver Configuration File,” on page 19](#)
- ♦ [Section 4.2.2, “Configuring the Driver Settings,” on page 20](#)
- ♦ [Section 4.2.3, “Deploying the Driver,” on page 21](#)
- ♦ [Section 4.2.4, “Starting the Driver,” on page 21](#)

### 4.2.1 Importing the Driver Configuration File

- 1 In Designer, open your project.
- 2 In the Modeler, right-click the driver set where you want to create the driver, then select *New > Driver* to display the Driver Configuration Wizard.
- 3 In the *Driver Configuration* list, select *WorkOrder*, then click *Run*.
- 4 On the Import Information Requested page, fill in the following fields:
  - Driver Name:** Specify a name that is unique within the driver set.
  - WorkOrder Container:** Browse for and select the container you’ve created for WorkOrder objects (see [Section 4.1, “Creating the WorkOrder Container in the Identity Vault,” on page 19](#)).

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

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

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

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

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

- 6 Click *Next* to import the driver configuration.

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

- 7 To review or modify the default configuration settings, click *Configure*, then continue with the next section, [Configuring the Driver Settings](#).

or

To skip the configuration settings at this time, click *Close*. When you are ready to configure the settings, continue with the next section, [Configuring the Driver Settings](#).

## 4.2.2 Configuring the Driver Settings

After importing the driver configuration file, the WorkOrder driver will run. However, the basic configuration might not meet the requirements for your environment. For example, you might need to change whether the driver checks for new work orders in the WorkOrder container at a specific interval throughout the day or only at a specific time each day. The default setting is to poll the WorkOrder container every minute.

In addition to the polling setting, there are additional settings that can help you customize and optimize the driver. The settings are divided into categories such as Driver Configuration, Engine Control Values, and Global Configuration Values (GCVs).

The driver configuration settings are explained in [Appendix A, "Driver Properties,"](#) on page 43.


If you do not have the Driver Properties page displayed in Designer:

- 1 Open your project.
- 2 In the Modeler, right-click the driver icon  or the driver line, then select *Properties*.

Although it is important for you to understand all of the settings, your first priority should be to review the [Driver Parameters](#) located on the Driver Configuration page. These settings let you control the method the driver uses to check for new work orders.

## 4.2.3 Deploying the Driver

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

- 1 In Designer, open your project.
- 2 In the Modeler, right-click the driver icon  or the driver line, then select *Live > Deploy*.
- 3 If you are authenticated to the Identity Vault, skip to [Step 5](#); otherwise, specify the following information to authenticate:
  - ♦ **Host:** Specify the IP address or DNS name of the server hosting the Identity Vault.
  - ♦ **Username:** Specify the DN of the user object used to authenticate to the Identity Vault.
  - ♦ **Password:** Specify the user's password.
- 4 Click *OK*.
- 5 Read the deployment summary, then click *Deploy*.
- 6 Read the message, then click *OK*.
- 7 Click *Define Security Equivalence* to assign rights to the driver.

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

  - 7a Click *Add*, then browse to and select the object with the correct rights.
  - 7b Click *OK* twice.
- 8 Click *Exclude Administrative Roles* to exclude users that should not be synchronized.


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

  - 8a Click *Add*, then browse to and select the user object you want to exclude.
  - 8b Click *OK*.
  - 8c Repeat [Step 8a](#) and [Step 8b](#) for each object you want to exclude.
  - 8d Click *OK*.
- 9 Click *OK*.

## 4.2.4 Starting the Driver

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

To start the driver:

- 1 In Designer, open your project.
- 2 In the Modeler, right-click the driver icon  or the driver line, then select *Live > Start Driver*.


For information about management tasks with the driver, see [Chapter 8, "Managing the Driver,"](#) on [page 39](#).

## 4.3 Creating the Driver in iManager

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

- ♦ [Section 4.3.1, "Importing the Driver Configuration File," on page 22](#)
- ♦ [Section 4.3.2, "Configuring the Driver Settings," on page 24](#)
- ♦ [Section 4.3.3, "Starting the Driver," on page 24](#)

### 4.3.1 Importing the Driver Configuration File

- 1 In iManager, click  to display the Identity Manager Administration page.
- 2 In the *Administration* list, click *Import Configuration* to launch the Import Configuration wizard.
- 3 Follow the wizard prompts, filling in the requested information (described below) until you reach the Summary page.

Prompt	Description
Where do you want to place the new driver?	You can add the driver to an existing driver set, or you can create a new driver set and add the driver to the new set. If you choose to create a new driver set, you are prompted to specify the name, context, and server for the driver set.
Import a configuration into this driver set	Use the default option, <i>Import a configuration from the server (.XML file)</i> .  In the <i>Show</i> field, select <i>Identity Manager 3.6.1 configurations</i> .  In the <i>Configurations</i> field, select the WorkOrder file.
Driver name	Type a name for the driver. The name must be unique within the driver set.
WorkOrder Container	Browse for and select the container you created for WorkOrder objects (see <a href="#">Section 4.1, "Creating the WorkOrder Container in the Identity Vault," on page 19</a> ).
Driver is Local/Remote	Select <i>Local</i> if this driver will run on the Metadirectory server without using the Remote Loader service. Select <i>Remote</i> if you want the driver to use the Remote Loader service, either locally on the Metadirectory server or remotely on another server.
Remote Host Name and Port	This applies only if the driver is running remotely.  Specify the hostname or IP address of the server where the driver's Remote Loader service is running.
Driver Password	This applies only if the driver is running remotely.  Specify the driver object password that is defined in the Remote Loader service. The Remote Loader requires this password to authenticate to the Metadirectory server.

Prompt	Description
Remote Password	This applies only if the driver is running remotely.  Specify the Remote Loader's password (as defined on the Remote Loader service). The Metadirectory engine (or Remote Loader shim) requires this password to authenticate to the Remote Loader
Define Security Equivalences	The driver requires rights to objects within the Identity Vault and to the input and output directories on the server. The Admin user object is most often used to supply these rights. However, you might want to create a DriversUser (for example) and assign security equivalence to that user. Whatever rights that the driver needs to have on the server, the DriversUser object must have the same security rights.
Exclude Administrative Roles	You should exclude any administrative User objects (for example, Admin and DriversUser) from synchronization.

When you finish providing the information required by the wizard, a Summary page similar to the following is displayed.

**Import Configuration**

Summary - Current Configuration

**Warning: Drivers May Require Configuration**

Drivers imported from a configuration file may require additional configuration settings to be fully functional. Select the driver's link to edit its configuration settings.

The following summarizes the state of the driver as it currently exists.

- Arrow (NCP Server)
- DS (Driver Set)
- work-order (Drivers May Require Configuration) (Driver)**
  - smp (Schema Mapping Policy)
  - none (Input Transformation Policy)
  - none (Output Transformation Policy)
- Publisher (Publisher)
  - none (Command Transformation Policy)
  - none (Event Transformation Policy)
  - none (Matching Policy)
  - none (Creation Policy)
  - pub-pp (Placement Policy)

<< Back   Next >>   Cancel   Finish

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

- To modify the default configuration settings, click the linked driver name, then continue with the next section, [Configuring the Driver Settings](#).

or


To skip the configuration settings at this time, click *Finish*. When you are ready to configure the settings, continue with the next section, [Configuring the Driver Settings](#).

### 4.3.2 Configuring the Driver Settings

After importing the driver configuration file, the WorkOrder driver will run. However, the basic configuration might not meet the requirements for your environment. For example, you might need to change whether the driver checks for new work orders in the WorkOrder container at a specific interval throughout the day or only at a specific time each day. The default setting is to poll the WorkOrder container every minute.

In addition to the polling setting, there are additional settings that can help you customize and optimize the driver. The settings are divided into categories such as Driver Configuration, Engine Control Values, and Global Configuration Values (GCVs).

To configure the settings:

- 1** Make sure the Modify Object page for the WorkOrder driver is displayed in iManager. If it is not:
  - 1a** In iManager, click  to display the Identity Manager Administration page.
  - 1b** Click *Identity Manager Overview*.
  - 1c** Browse to and select the driver set object that contains the new driver.
  - 1d** Click the driver set name to access the Driver Set Overview page.
  - 1e** Click the upper right corner of the driver, then click *Edit properties*.
- 2** Review the settings on the various pages and modify them as needed for your environment. The configuration settings are explained in “[Driver Parameters](#)” on page 46.

Although it is important for you to understand all of the settings, your first priority should be to review the [Driver Parameters](#) located on the Driver Configuration page. These settings let you control the method the driver uses to check for new work orders.
- 3** After modifying the settings, click *OK* to save the settings and close the Modify Object page.
- 4** (Conditional) If the WorkOrder driver’s Summary page for the Import Configuration Wizard is still displayed, click *Finish*.

---


**WARNING:** Do not click *Cancel* on the Summary page. This removes the driver from the Identity Vault and loses your work.

---

### 4.3.3 Starting the Driver

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

To start the driver:

- 1** In iManager, click  to display the Identity Manager Administration page.
- 2** Click *Identity Manager Overview*.
- 3** Browse to and select the driver set object that contains the driver you want to start.



- 4 Click the driver set name to access the Driver Set Overview page.
- 5 Click the upper right corner of the driver, then click *Start driver*.

For information about management tasks with the driver, see [Chapter 8, “Managing the Driver,” on page 39](#).

## 4.4 Activating the Driver

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

For information on activation, refer to “[Activating Novell Identity Manager Products](#)” in the *Identity Manager 3.6.1 Installation Guide*.



# Upgrading an Existing Driver

# 5

If you are running the driver on the Metadirectory server, the driver shim files are updated when you update the server unless they were not selected during a custom installation. If you are running the driver on another server, the driver shim files are updated when you update the Remote Loader on the server.

The 3.6.1 version of the driver shim supports drivers created by using any 3.x version of the driver configuration file. You can continue to use these driver configurations until you want to upgrade them.

The following sections provide information to help you upgrade an existing driver's configuration to version 3.6.1:

- ♦ [Section 5.1, “Supported Upgrade Paths,” on page 27](#)
- ♦ [Section 5.2, “What’s New in Version 3.6.1,” on page 27](#)
- ♦ [Section 5.3, “Upgrade Procedure,” on page 27](#)

## 5.1 Supported Upgrade Paths

You can upgrade from any 3.x version of the WorkOrder driver. Upgrading a pre-3.x version of the driver directly to version 3.6.1 is not supported.

## 5.2 What’s New in Version 3.6.1

The 3.6.1 version of the driver does not include any new features.

## 5.3 Upgrade Procedure

The process for upgrading the WorkOrder driver is the same as for other Identity Manager drivers. For detailed instructions, see “[Upgrading](#)” in the *Identity Manager 3.6.1 Installation Guide*.



# Customizing the Driver

# 6

After you create a new WorkOrder driver by importing the basic configuration file, the driver processes WorkOrder objects from the Identity Vault to create WorkToDo objects. This is all the WorkOrder driver does when using the basic configuration. For any additional work to be done, you must customize the WorkOrder driver or other Identity Manager drivers to perform the desired work.

The following section describes how the policies and rules are set up in the basic configuration:

- ♦ [Section 6.1, “Policies and Rules Used in the Basic Configuration,” on page 29](#)

The WorkOrder driver is extremely flexible in what you can do with it. The following sections illustrate two possible solutions for customizing your driver:

- ♦ [Section 6.2, “Human Resource Example Using an HR Driver,” on page 31](#)
- ♦ [Section 6.3, “Human Resource Example without an HR Driver,” on page 33](#)

## 6.1 Policies and Rules Used in the Basic Configuration

This section describes policies and rules for the Subscriber and Publisher channels in the WorkOrder driver’s basic configuration. For an overview on how the Subscriber and Publisher channels work, see [Section 1.1.1, “Subscriber Channel Functions,” on page 9](#) and [Section 1.1.2, “Publisher Channel Functions,” on page 11](#).

### 6.1.1 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 6-1** *Configuring the Subscriber Channel*

Rule or Policy	What it does
Subscriber Filter	Allows only events for WorkOrder objects to be processed.
Event Transformation	Not used in the sample configuration.
Matching Rule	Not used in the sample configuration.

Rule or Policy	What it does
Create Rule	<p>Contains rules only for WorkOrder objects.</p> <p>Requires values for the following attributes on a WorkOrder object:</p> <ul style="list-style-type: none"> <li>woStatus</li> <li>woSendToPublisher</li> <li>woDoltNow</li> <li>woContent</li> <li>woType</li> </ul> <p>If the values are not present, the work order is not sent to the Publisher channel and the work order is not updated by the driver.</p> <p>For a description of these attributes, see <a href="#">Appendix C, “Schema and Policy Rules For Work Order Management,” on page 51.</a></p>
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.

## 6.1.2 Publisher Channel

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

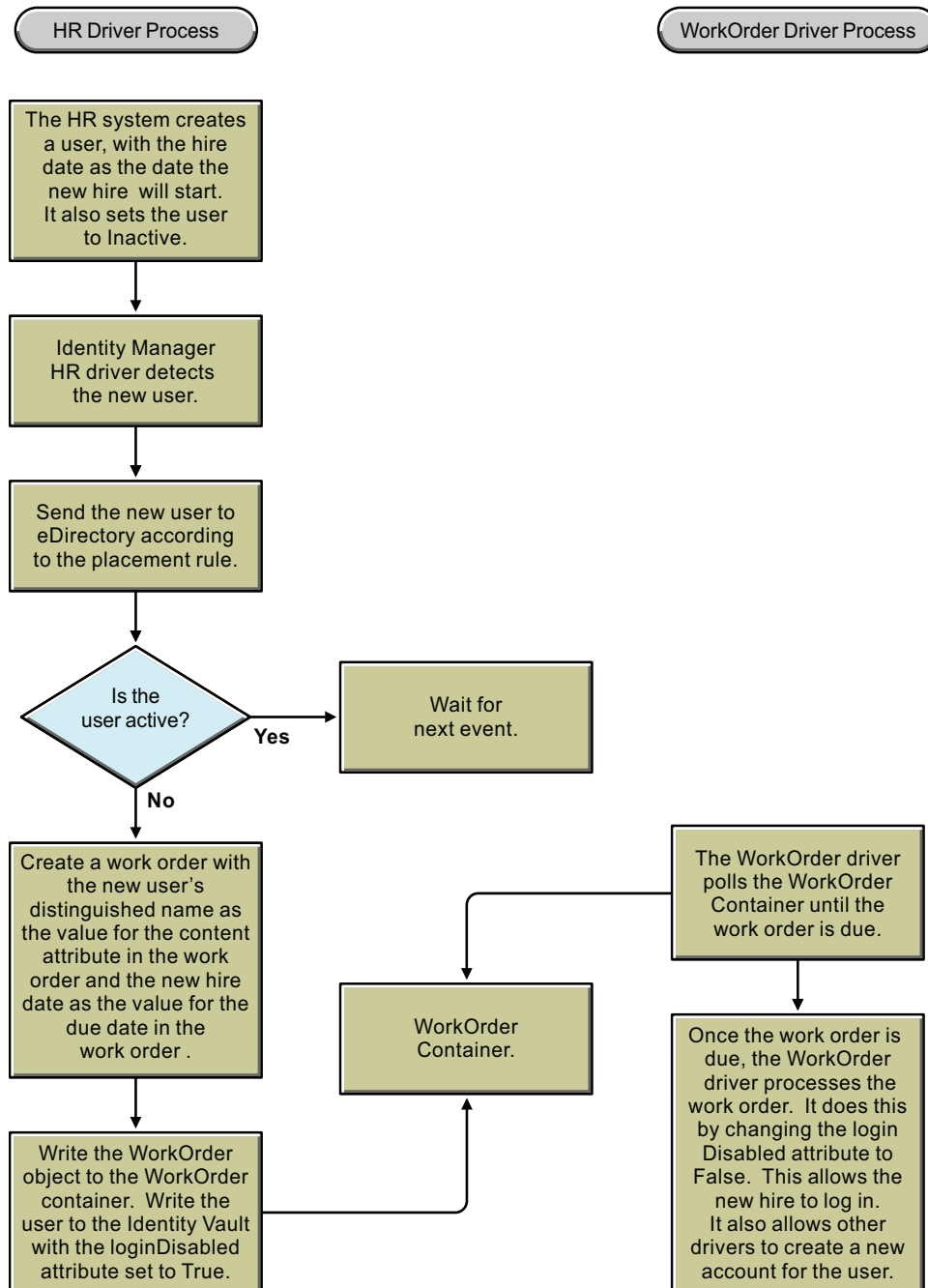
**Table 6-2** *Configuring the Publisher channel*

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

## 6.2 Human Resource Example Using an HR Driver

The following example illustrates 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 6-1](#) illustrates how these drivers work together in the example configuration.

**Figure 6-1** Data Flow with an HR Driver



In this scenario, assume the new employee's name is Albert Hauser. Albert is hired, but does not begin work until a future date and time. He is put into the HR system with the hire date set. Albert is marked as not active and does not have access to the system.

The HR Identity Manager driver writes Albert's 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 Albert's account 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 Albert 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.

## 6.2.1 Human Resource Driver Policies

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

### Mapping Rule

The mapping rule maps the attributes used in the WorkOrder driver to attributes in the Identity Vault. You can view the sample at [hr-drv-schema-map.xml](http://www.novell.com/documentation/idm36drivers/work_order/samples/hr-drv-schema-map.xml) ([http://www.novell.com/documentation/idm36drivers/work\\_order/samples/hr-drv-schema-map.xml](http://www.novell.com/documentation/idm36drivers/work_order/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](http://www.novell.com/documentation/idm36drivers/work_order/samples/hr-drv-schema-map.xml) ([http://www.novell.com/documentation/idm36drivers/work\\_order/samples/hr-drv-filter.xml](http://www.novell.com/documentation/idm36drivers/work_order/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 into the DirXML-woContent attribute. You can view the sample at [hr-drv-cmd-transform.xml](http://www.novell.com/documentation/idm36drivers/work_order/samples/hr-drv-cmd-transform.xml) ([http://www.novell.com/documentation/idm36drivers/work\\_order/samples/hr-drv-cmd-transform.xml](http://www.novell.com/documentation/idm36drivers/work_order/samples/hr-drv-cmd-transform.xml)).

A second policy 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-transform2.xml](http://www.novell.com/documentation/idm36drivers/work_order/samples/hr-drv-cmd-transform2.xml) ([http://www.novell.com/documentation/idm36drivers/work\\_order/samples/hr-drv-cmd-transform2.xml](http://www.novell.com/documentation/idm36drivers/work_order/samples/hr-drv-cmd-transform2.xml)).



## 6.2.2 WorkOrder Driver Policy

The WorkOrder driver policy uses only the Publisher Command Transformation policy. The 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-woContent attribute. It then sets the user's Login Disabled 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.

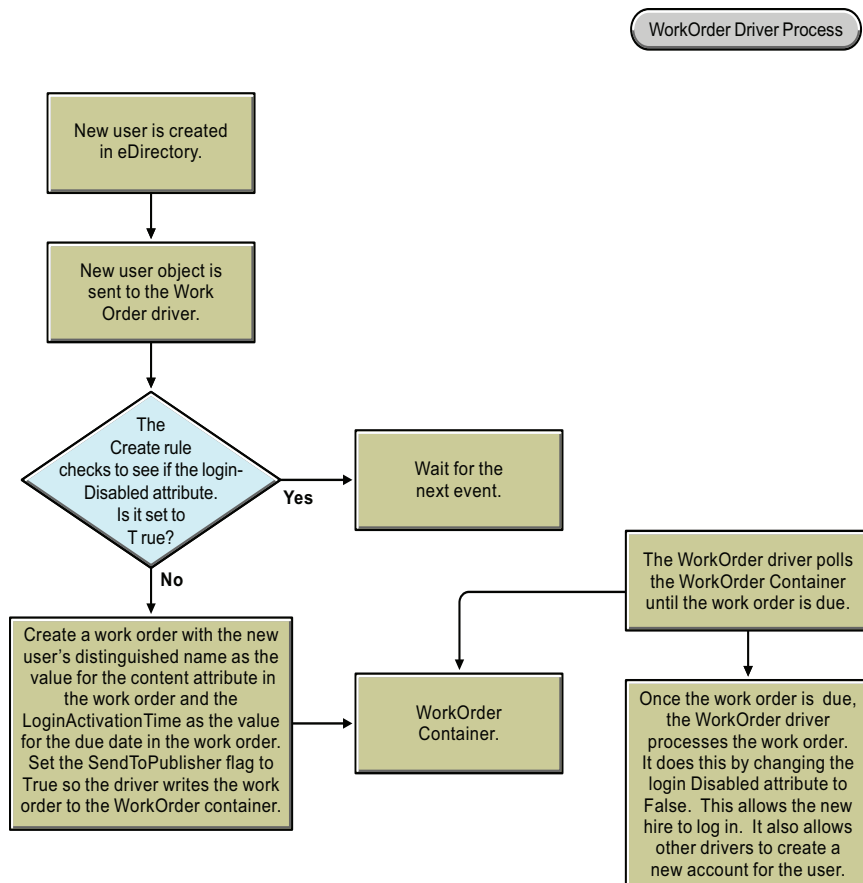
---

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](http://www.novell.com/documentation/idm36drivers/work_order/samples/hr-wo-drv-pub-cmd-transform.xml) ([http://www.novell.com/documentation/idm36drivers/work\\_order/samples/hr-wo-drv-pub-cmd-transform.xml](http://www.novell.com/documentation/idm36drivers/work_order/samples/hr-wo-drv-pub-cmd-transform.xml)).

## 6.3 Human Resource Example without an HR Driver

This example 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 6-2](#) illustrates this sample configuration.

**Figure 6-2** Data Flow without an HR Driver



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.

The following policies show how to implement the sample configuration:

### 6.3.1 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](http://www.novell.com/documentation/idm36drivers/work_order/samples/wo-filter.xml) ([http://www.novell.com/documentation/idm36drivers/work\\_order/samples/wo-filter.xml](http://www.novell.com/documentation/idm36drivers/work_order/samples/wo-filter.xml)).

### 6.3.2 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](http://www.novell.com/documentation/idm36drivers/work_order/samples/wo-create.xml) ([http://www.novell.com/documentation/idm36drivers/work\\_order/samples/wo-create.xml](http://www.novell.com/documentation/idm36drivers/work_order/samples/wo-create.xml)).

### 6.3.3 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-woContent attribute. The DirXML-DueDate is set to the loginActivationTime. The DirXML-woStatus is set to pending. The DirXML-woSendToPublisher attribute is set to True.

This work order has not yet been created in the Identity Vault, so the 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](http://www.novell.com/documentation/idm36drivers/work_order/samples/wo-sub-cmd-transform.xml) ([http://www.novell.com/documentation/idm36drivers/work\\_order/samples/wo-sub-cmd-transform.xml](http://www.novell.com/documentation/idm36drivers/work_order/samples/wo-sub-cmd-transform.xml)).

### 6.3.4 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](http://www.novell.com/documentation/idm36drivers/work_order/samples/wo-pub-cmd-transform.xml) ([http://www.novell.com/documentation/idm36drivers/work\\_order/samples/wo-pub-cmd-transform.xml](http://www.novell.com/documentation/idm36drivers/work_order/samples/wo-pub-cmd-transform.xml)).

# Creating and Managing Work Orders

# 7

There are two ways to create work orders. The following sections review how this is accomplished:

- [Section 7.1, “Using Drivers to Create Work Orders,” on page 35](#)
- [Section 7.2, “Using iManager,” on page 35](#)

## 7.1 Using Drivers to Create Work Orders


Identity Manager drivers can create work orders as a result of events processed by the drivers. For example, if you use a Human Resource driver (SAP\*, Peoplesoft\*, and so forth), you can have the driver generate a work order whenever a new user is added. [Chapter 6, “Customizing the Driver,” on page 29](#) provides examples how to use a driver to create a work order.

## 7.2 Using iManager

You can use iManager to manually create and maintain work orders:


- [Section 7.2.1, “Creating a New Work Order,” on page 35](#)
- [Section 7.2.2, “Editing Work Order Properties,” on page 35](#)
- [Section 7.2.3, “Filtering the Work Order List,” on page 37](#)

### 7.2.1 Creating a New Work Order

- 1 In iManager, click  to display the Identity Manager Administration page.
- 2 In the *Features* list, click *Work Order Management* to display the Work Order Management page.
- 3 In the *WorkOrder Driver* field, browse for and select the WorkOrder driver for which you are creating the work order.
- 4 Click *New*, specify a name for the work order, then click *OK*.  
The name is used for the WorkOrder object’s name in the Identity Vault.
- 5 Fill in the fields on the WorkOrder page. For information about the fields, see the next section, [Editing Work Order Properties](#).

### 7.2.2 Editing Work Order Properties

The Work Order page lets you configure a new work order or edit an existing work order.

- 1 If you are editing an existing work order and the Work Order page is not already open, open the Work Order page:
  - 1a In iManager, click  to display the Identity Manager Administration page.
  - 1b In the Features list, click *Work Order Management* to display the Work Order Management page.

**1c** In the WorkOrder Driver field, browse for and select the WorkOrder driver associated with the work order you want to edit.

After you select the appropriate WorkOrder driver, all work orders associated with the driver are listed.

**1d** Click the work order you want to edit.



**2** Fill in the following fields:

**Status:** The status of a new work order can be either *Pending* or *On Hold*. Normally, work order status is *Pending*. You can stop a work order by selecting *On Hold*. After a work order has been processed, the resulting work order status appears in this field.

**Due Date:** You can choose to have the driver do the work order immediately or schedule the work order. To schedule a due date, click the calendar icon. Use the calendar to choose the date. Use the arrows to select the month, year, and time.

**Repeat Work Order:** Select this option to have the work order processed multiple times. 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 date unless it is manually deleted, edited, or the driver sends back an error message.

**Delete Date:** Use the calendar control to select a date to delete work orders that have been configured. Work orders with an error status are not deleted unless you select *Delete Work Order Even if the Work Order Has an Error*.

**Dependent Work Orders:** When creating a new work order, you can make it dependent on one or more work orders. Click  to browse for and select dependent work orders. To remove a work order from the list, select the work order, then click .

**Type:** Use this field to specify a work order type. The driver does not change this attribute. The attribute is passed through to the WorkToDo object when the work order is processed.

**Work Order Number:** A unique work order number. This value can be assigned by a corporate work order system other than Novell® eDirectory™, such as a work order database.

**Contact Information:** Contact information for the person responsible for the work order.

**Work Order Processing Log:** After a work order has been processed, the driver logs the results of the work order, including the status, in this field. This allows you to check the work order's current status and identify any problems the driver encountered while attempting to configure the work order.

The work order's status attribute remains pending until the work order is processed. The work order is processed when the due date has expired or the Do It Now flag is set. The driver reports the processing results by setting the status attribute to Configured, Warning, or Error. If the work order is On Hold, it ignores the work order.

- ♦ **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 sends a warning.

**Description:** The work order description.

**Work Order Content:** The data in this field is used by the driver's rules to process the work order. For example, it might be the XML that the Command Transformation uses to process the work order.

- 3 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.

### 7.2.3 Filtering 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 orders that are not yet due are listed.



# Managing the Driver

# 8

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

- ◆ Starting and stopping the driver
- ◆ Viewing driver version information
- ◆ Using Named Passwords to securely store passwords associated with the driver
- ◆ Monitoring the driver's health status
- ◆ Backing up the driver
- ◆ Inspecting the driver's cache files
- ◆ Viewing the driver's statistics
- ◆ Using the DirXML<sup>®</sup> Command Line utility to perform management tasks through scripts
- ◆ Securing the driver and its information

Because these tasks, as well as several others, are common to all Identity Manager drivers, they are included in one reference, the [Identity Manager 3.6.1 Common Driver Administration Guide](#).





# Troubleshooting Driver Processes

# 9


Viewing driver processes is necessary to analyze unexpected behavior. To view the driver processing events, use DTrace. You should only use it during testing and troubleshooting the driver. Running DTrace while the drivers are in production increases the utilization on the Identity Manager server and can cause events to process very slowly. For more information, see “[Viewing Identity Manager Processes](#)” in the *Identity Manager 3.6.1 Common Driver Administration Guide*.



# Driver Properties

# A


This section provides information about the Driver Configuration and Global Configuration Values properties for the WorkOrder driver. These are the only unique properties for drivers. All other driver properties (Named Password, Engine Control Values, Log Level, and so forth) are common to all drivers. Refer to “[Driver Properties](#)” in the *Identity Manager 3.6.1 Common Driver Administration Guide* for information about the common properties.

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

- ♦ [Section A.1, “Driver Configuration,” on page 43](#)
- ♦ [Section A.2, “Global Configuration Values,” on page 47](#)

## A.1 Driver Configuration

In iManager:

- 1 Click  to display the Identity Manager Administration page.
- 2 Open the driver set that contains the driver whose properties you want to edit:
  - 2a In the *Administration* list, click *Identity Manager Overview*.
  - 2b If the driver set is not listed on the *Driver Sets* tab, use the *Search In* field to search for and display the driver set.
  - 2c Click the driver set to open the Driver Set Overview page.
- 3 Locate the WorkOrder driver icon, then click the upper right corner of the driver icon to display the *Actions* menu.
- 4 Click *Edit Properties* to display the driver’s properties page.

By default, the Driver Configuration page is displayed.

In Designer:

- 1 Open a project in the Modeler.
- 2 Right-click the driver icon or line, then select click *Properties > Driver Configuration*.



The Driver Configuration options are divided into the following sections:

- ♦ [Section A.1.1, “Driver Module,” on page 44](#)
- ♦ [Section A.1.2, “Driver Object Password \(iManager Only\),” on page 44](#)
- ♦ [Section A.1.3, “Authentication,” on page 44](#)
- ♦ [Section A.1.4, “Startup Option,” on page 45](#)
- ♦ [Section A.1.5, “Driver Parameters,” on page 46](#)
- ♦ [Section A.1.6, “ECMAScript \(Designer Only\),” on page 47](#)

## A.1.1 Driver Module

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

**Table A-1** *Driver Modules*

Option	Description
<i>Java</i>	<p>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.</p> <p>The name of the Java class is:</p> <pre>com.novell.nds.dirxml.driver.workorder.WorkOrderDriverShim</pre>
<i>Connect to Remote Loader</i>	<p>Used when the driver is connecting remotely to the connected system. Designer includes two suboptions:</p> <ul style="list-style-type: none"><li>◆  <i>Driver Object Password</i>: Specifies a password for the Driver object. If you are using the Remote Loader, you must enter a password on this page. Otherwise, the remote driver does not run. The Remote Loader uses this password to authenticate itself to the remote driver shim.</li><li>◆  <i>Remote Loader Client Configuration for Documentation</i>: Includes information on the Remote Loader client configuration when Designer generates documentation for the driver.</li></ul>

## A.1.2 Driver Object Password (iManager Only)











**Table A-2** *Driver Object Password*

Option	Description
<i>Driver Object Password</i>	<p>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.</p>

## A.1.3 Authentication

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

**Table A-3** *Authentication Options*


Option	Description
<p><i>Authentication ID</i></p> <p>or</p> <p> <i>User ID</i></p>	<p>Specify a user application ID. This ID is used to pass Identity Vault subscription information to the application.</p> <p>Example: Administrator</p>
<p><i>Authentication Context</i></p> <p>or</p> <p> <i>Connection Information</i></p>	<p>Specify the IP address or name of the server the application shim should communicate with.</p>
<p><i>Remote Loader Connection Parameters</i></p> <p>or</p> <p> <i>Host name</i></p> <p> <i>Port</i></p> <p> <i>KMO</i></p> <p> <i>Other parameters</i></p>	<p>Used only if the driver is connecting to the application through the Remote Loader. The parameter to enter is</p> <p><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.</p> <p>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.</p> <p>Example: <code>hostname=10.0.0.1 port=8090</code>  <code>kmo=IDMCertificate</code></p>
<p><i>Driver Cache Limit (kilobytes)</i></p> <p>or</p> <p> <i>Cache limit (KB)</i></p>	<p>Specify the maximum event cache file size (in KB). If it is set to zero, the file size is unlimited.</p> <p> Click <i>Unlimited</i> to set the file size to unlimited in Designer.</p>
<p><i>Application Password</i></p> <p>or</p> <p> <i>Set Password</i></p>	<p>Specify the password for the user object listed in the <i>Authentication ID</i> field.</p>
<p><i>Remote Loader Password</i></p> <p>or</p> <p> <i>Set Password</i></p>	<p>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.</p>

## A.1.4 Startup Option

The startup options allow you to set the driver state when the Identity Manager server is started.

**Table A-4** *Startup Options*

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.

Option	Description
<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.

## A.1.5 Driver Parameters

The Driver Parameters section lets you configure the driver-specific parameters. When you change driver parameters, you tune driver behavior to align with your network environment. For example, you might find the polling interval to be shorter than you need. Making the interval longer could improve network performance while still maintaining your performance expectations for work order processing.

**Table A-5** *Driver Parameters*

Option	Description
<i>Driver Name</i>	The actual name you want to use for the driver.
<i>WorkOrders Container</i>	The name of the container where WorkOrder objects and WorkToDo objects are to be stored.
<i>Poll Interval</i>	How often the Publisher channel polls the WorkOrder container for work orders to be configured. The default is one minute. You can use this setting, not use this setting, or use it with the Poll Time setting. If you don't want to use this setting, leave the field empty.
<i>Poll Time</i>	Time of day the Publisher channel checks the WorkOrder container for work orders to be configured. By default, this setting is disabled ( <i>No poll time</i> ) so that only the Poll Interval setting is used. However, you can use this setting instead of the Poll Interval setting, or you can use it with the Poll Interval setting.  The poll times are available in half-hour increments. If you need a more granular poll time (for example, 1:15 PM rather than 1:00 PM or 1:30 PM), click the <i>Edit XML</i> button, locate the <code>&lt;definition display-name="Poll Time" id="112" name="polling-time" type="enum"&gt;</code> entry, and change the type from <code>enum</code> to <code>string</code> . Click <i>OK</i> to save the change, then enter the desired time in the Poll Time field. Use the <code>HH:MM AM/PM</code> format (for example, 1:15 PM).

Option	Description
<i>Publisher Heartbeat every Poll Interval</i>	Specifies if the Publisher should emit heartbeat documents. The driver emits heartbeat documents to indicate to the Identity Manager engine that the driver is still functioning.  If you don't use the Poll Interval setting, this setting is automatically disabled.

## A.1.6 ECMAScript (Designer Only)

Enables you to add ECMAScript resource files. The resources extend the driver's functionality when Identity Manager starts the driver.

## A.2 Global Configuration Values


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

The WorkOrder driver includes one predefined GCV.

**Table A-6** *Global Configuration Values*

Option	Description
WorkOrder Container	This is the WorkOrder container that is specified by the <a href="#">WorkOrder Container</a> setting on the Driver Configuration page. You can change the setting on the Driver Configuration page or on the GCV page.  The GCV is included as a driver set GCV (not a driver GCV) so that it can be used by other drivers as they create work orders to be placed in the WorkOrder container.

To modify the driver's GCVs in iManager:


- 1** Click  to display the Identity Manager Administration page.
- 2** Open the driver set that contains the driver whose properties you want to edit.
  - 2a** In the *Administration* list, click *Identity Manager Overview*.
  - 2b** If the driver set is not listed on the *Driver Sets* tab, use the *Search In* field to search for and display the driver set.
  - 2c** Click the driver set to open the Driver Set Overview page.
- 3** To add a GCV to the WorkOrder driver, locate the WorkOrder driver icon, click the upper right corner of the driver icon to display the *Actions* menu, then click *Edit Properties*.

or


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

To modify the driver's GCVs in Designer:

- 1** Open a project in the Modeler.

**2** To add a GCV to the WorkOrder driver, right-click the driver icon  or line, then select *Properties > Global Configuration Values*.

or

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



# Objects and Attributes Used

# B

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

- ◆ [Section B.1, “New Objects Used by the Driver,” on page 49](#)
- ◆ [Section B.2, “DoItNow and SendToPublisher Flags,” on page 49](#)

## B.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 C, “Schema and Policy Rules For Work Order Management,” on page 51](#).

- ◆ [Section B.1.1, “DirXML-WorkOrder Object,” on page 49](#)
- ◆ [Section B.1.2, “DirXML-WorkToDo Object,” on page 49](#)

### B.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 B.2, “DoItNow and SendToPublisher Flags,” on page 49](#).

An iManager plug-in is provided to help you create and maintain work orders. To learn how to use the plug-in, see [Chapter 7, “Creating and Managing Work Orders,” on page 35](#).

### B.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, 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 is 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.

## B.2 DoItNow and SendToPublisher Flags

The WorkOrder driver has two flags to initiate a work order event.

- ◆ [Section B.2.1, “DoItNow Flag,” on page 50](#)
- ◆ [Section B.2.2, “SendToPublisherFlag,” on page 50](#)

## **B.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.

## **B.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 initiated by a policy in response to an event in the Identity Vault, setting the flag to True enables the WorkOrder driver to create the WorkOrder object and add it to the Identity Vault's work order container. The WorkOrder object can then be processed like any other WorkOrder object added to the container by another driver.

# Schema and Policy Rules For Work Order Management

# C

As part of the installation of the WorkOrder driver, Novell® 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.

You can use iManager to create or view these objects in the Identity Vault. See [Chapter 6, “Customizing the Driver,”](#) on page 29.

- ♦ [Section C.1, “DirXML-WorkOrder Object,”](#) on page 51
- ♦ [Section C.2, “DirXML-WorkToDo Object,”](#) on page 53
- ♦ [Section C.3, “Publisher Placement Rule,”](#) on page 54
- ♦ [Section C.4, “Subscriber Placement Rule,”](#) on page 54
- ♦ [Section C.5, “Subscriber Create Rule,”](#) on page 54

## C.1 DirXML-WorkOrder Object

The DirXML-WorkOrder object (sometimes referred to as the WorkOrder object in this documentation) 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 C-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-woContact 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-woContent	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-woDoItNowFlag	If set to True, the Subscriber channel sends the work order to the Publisher channel to be processed immediately.	Boolean
DirXML-woSendToPublisher	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-woCreationDate	Information about the work order. The driver does not change this attribute.	Time
DirXML-woDependentWorkOrder	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-woRepeatInterval	The amount of time, in minutes, before the work order is repeated. This value is added to the due date after the work order is processed.	Case ignore string
DirXML-woRepeatCount	Repeats the work order as many times as the number specifies. Use this attribute in association with the DirXML-woRepeatInterval attribute.	Case ignore string
DirXML-woStatus	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-woWorkOrderNumber	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-woDeleteOnError	If set to True, the work order is deleted if the status is Error and the DeleteDueDate has expired.	Boolean
DirXML-woProcessLog	Contains information relating to the processing of the work order.	Case ignore string

Work Order Attributes (eDirectory Namespace)	Description	Type
DirXML-woDeleteDueDate	If the status is Pending or Configured, this attribute shows the date and time the work order will be deleted.	Time
DirXML-CreatorName	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.	Distinguished Name
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

## C.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 C-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-woContent	The value of the content attribute in the work order.	Case ignore string
DirXML-woDN	DN of the work order.	Distinguished Name
Description	Information about the work order. The driver does not change this attribute.	Case ignore string
DirXML-woContactName	Information about the work order. The driver does not change this attribute.	Case ignore string
DirXML-woWorkOrderNumber	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

## C.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.

## C.4 Subscriber Placement Rule

The Subscriber Placement rule determines the container that work orders are created in and sent to the WorkOrder driver.

## C.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, otherwise the event cannot be processed further:

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

Required Attributes	Description	Values or Examples
DirXML-woSendToPublisher	Send the work order directly to the Publisher channel.	True or False
DirXML-woStatus	State of the work order so the driver knows what to do with the work order.	Pending, Configured, Error, on Hold, Warning
DirXML-woDoItNowFlag	When to perform the work order.	True or False
DirXML-woContent	Content to be processed by the driver.	XML code
DirXML-woType	Information about the work order. The driver does not change this attribute.	Case ignore string