



# NetIQ® Identity Manager™

## WorkOrder Driver Implementation Guide

October 2014

## Legal Notices

THIS DOCUMENT AND THE SOFTWARE DESCRIBED IN THIS DOCUMENT ARE FURNISHED UNDER AND ARE SUBJECT TO THE TERMS OF A LICENSE AGREEMENT OR A NON-DISCLOSURE AGREEMENT. EXCEPT AS EXPRESSLY SET FORTH IN SUCH LICENSE AGREEMENT OR NON-DISCLOSURE AGREEMENT, NETIQ CORPORATION PROVIDES THIS DOCUMENT AND THE SOFTWARE DESCRIBED IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW DISCLAIMERS OF EXPRESS OR IMPLIED WARRANTIES IN CERTAIN TRANSACTIONS; THEREFORE, THIS STATEMENT MAY NOT APPLY TO YOU.

For purposes of clarity, any module, adapter or other similar material ("Module") is licensed under the terms and conditions of the End User License Agreement for the applicable version of the NetIQ product or software to which it relates or interoperates with, and by accessing, copying or using a Module you agree to be bound by such terms. If you do not agree to the terms of the End User License Agreement you are not authorized to use, access or copy a Module and you must destroy all copies of the Module and contact NetIQ for further instructions.

This document and the software described in this document may not be lent, sold, or given away without the prior written permission of NetIQ Corporation, except as otherwise permitted by law. Except as expressly set forth in such license agreement or non-disclosure agreement, no part of this document or the software described in this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, or otherwise, without the prior written consent of NetIQ Corporation. Some companies, names, and data in this document are used for illustration purposes and may not represent real companies, individuals, or data.

This document could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein. These changes may be incorporated in new editions of this document. NetIQ Corporation may make improvements in or changes to the software described in this document at any time.

U.S. Government Restricted Rights: If the software and documentation are being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), in accordance with 48 C.F.R. 227.7202-4 (for Department of Defense (DOD) acquisitions) and 48 C.F.R. 2.101 and 12.212 (for non-DOD acquisitions), the government's rights in the software and documentation, including its rights to use, modify, reproduce, release, perform, display or disclose the software or documentation, will be subject in all respects to the commercial license rights and restrictions provided in the license agreement.

**© 2014 NetIQ Corporation. All Rights Reserved.**

For information about NetIQ trademarks, see <https://www.netiq.com/company/legal/>.

---

# Contents

|   |           |
|---|-----------|
| <b>About this Book and the Library</b>                        | <b>5</b>  |
| <b>About NetIQ Corporation</b>                                | <b>7</b>  |
| <b>1 Understanding the WorkOrder Driver</b>                   | <b>9</b>  |
| 1.1 The Work Order Process                                    | 9         |
| 1.1.1 Subscriber Channel Functions                            | 10        |
| 1.1.2 Publisher Channel Functions                             | 11        |
| 1.2 Key Features  | 14        |
| 1.3 Support for Standard Driver Functions                     | 15        |
| 1.4 Terminology   | 15        |
| <b>2 Implementation Checklist</b>                             | <b>17</b> |
| <b>3 Installing Driver Files</b>                              | <b>19</b> |
| <b>4 Creating a New Driver Object</b>                         | <b>21</b> |
| 4.1 Creating the WorkOrder Container in the Identity Vault    | 21        |
| 4.2 Creating the Driver Object in Designer                    | 21        |
| 4.2.1 Importing the Current Driver Packages                   | 22        |
| 4.2.2 Installing the Driver Packages                          | 22        |
| 4.2.3 Configuring the Driver Settings                         | 23        |
| 4.2.4 Deploying the Driver Object                             | 24        |
| 4.2.5 Starting the Driver                                     | 24        |
| 4.3 Activating the Driver                                     | 25        |
| 4.4 Adding Packages to an Existing 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 4.5                                 | 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                 | 30        |
| 6.2.1 Human Resource Driver Policies                          | 32        |
| 6.2.2 WorkOrder Driver Policy                                 | 32        |
| 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 to Create Work Orders . . . . .           | 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 the Driver</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 Options . . . . .                                | 45        |
| A.1.5    | Driver Parameters . . . . .                              | 45        |
| A.1.6    | ECMAScript . . . . .                                     | 46        |
| A.1.7    | Global Configuration . . . . .                           | 46        |
| A.2      | Global Configuration Values . . . . .                    | 46        |
| <b>B</b> | <b>Objects and Attributes</b>                            | <b>47</b> |
| B.1      | New Objects Used by the Driver . . . . .                 | 47        |
| B.1.1    | DirXML-WorkOrder Object . . . . .                        | 47        |
| B.1.2    | DirXML-WorkToDo Object . . . . .                         | 47        |
| B.2      | DoltNow and SendToPublisher Flags . . . . .              | 47        |
| B.2.1    | DoltNow Flag . . . . .                                   | 48        |
| B.2.2    | SendToPublisherFlag . . . . .                            | 48        |
| <b>C</b> | <b>Schema and Policy Rules For Work Order Management</b> | <b>49</b> |
| C.1      | DirXML-WorkOrder Object . . . . .                        | 49        |
| C.2      | DirXML-WorkToDo Object . . . . .                         | 51        |
| C.3      | Publisher Placement Rule . . . . .                       | 52        |
| C.4      | Subscriber Placement Rule . . . . .                      | 52        |
| C.5      | Subscriber Create Rule . . . . .                         | 52        |

---

# About this Book and the Library

The *Identity Manager WorkOrder Driver Implementation Guide* explains how to install, configure, and manage the Identity Manager WorkOrder driver.

## Intended Audience

This book provides information for developers and administrators using Identity Manager and the WorkOrder driver.

## Other Information in the Library

The library provides the following information resources:

### **Identity Manager Setup Guide**

Provides overview of Identity Manager and its components. This book also provides detailed planning and installation information for Identity Manager.

### **Designer Administration Guide**

Provides information about designing, testing, documenting, and deploying Identity Manager solutions in a highly productive environment.

### **User Application: Administration Guide**

Describes how to administer the Identity Manager User Application.

### **User Application: User Guide**

Describes the user interface of the Identity Manager User Application and how you can use the features it offers, including identity self-service, the Work Dashboard, role and resource management, and compliance management.

### **User Application: Design Guide**

Describes how to use the Designer to create User Application components, including how to work with the Provisioning view, the directory abstraction layer editor, the provisioning request definition editor, the provisioning team editor, and the role catalog.

### **Identity Reporting Module Guide**

Describes the Identity Reporting Module for Identity Manager and how you can use the features it offers, including the Reporting Module user interface and custom report definitions, as well as providing installation instructions.

### **Analyzer Administration Guide**

Describes how to administer Analyzer for Identity Manager.

### **Identity Manager Common Driver Administration Guide**

Provides information about administration tasks that are common to all Identity Manager drivers.

## **Identity Manager Driver Guides**

Provides implementation information about Identity Manager drivers.

---

# About NetIQ Corporation

We are a global, enterprise software company, with a focus on the three persistent challenges in your environment: Change, complexity and risk—and how we can help you control them.

## Our Viewpoint

### **Adapting to change and managing complexity and risk are nothing new**

In fact, of all the challenges you face, these are perhaps the most prominent variables that deny you the control you need to securely measure, monitor, and manage your physical, virtual, and cloud computing environments.

### **Enabling critical business services, better and faster**

We believe that providing as much control as possible to IT organizations is the only way to enable timelier and cost effective delivery of services. Persistent pressures like change and complexity will only continue to increase as organizations continue to change and the technologies needed to manage them become inherently more complex.

## Our Philosophy

### **Selling intelligent solutions, not just software**

In order to provide reliable control, we first make sure we understand the real-world scenarios in which IT organizations like yours operate—day in and day out. That's the only way we can develop practical, intelligent IT solutions that successfully yield proven, measurable results. And that's so much more rewarding than simply selling software.

### **Driving your success is our passion**

We place your success at the heart of how we do business. From product inception to deployment, we understand that you need IT solutions that work well and integrate seamlessly with your existing investments; you need ongoing support and training post-deployment; and you need someone that is truly easy to work with—for a change. Ultimately, when you succeed, we all succeed.

## Our Solutions

- ◆ Identity & Access Governance
- ◆ Access Management
- ◆ Security Management
- ◆ Systems & Application Management
- ◆ Workload Management
- ◆ Service Management

## Contacting Sales Support

For questions about products, pricing, and capabilities, contact your local partner. If you cannot contact your partner, contact our Sales Support team.

|                                  |  |
|----------------------------------|--|
| <b>Worldwide:</b>                | <a href="http://www.netiq.com/about_netiq/officelocations.asp">www.netiq.com/about_netiq/officelocations.asp</a> |
| <b>United States and Canada:</b> | 1-888-323-6768   |
| <b>Email:</b>                    | <a href="mailto:info@netiq.com">info@netiq.com</a>   |
| <b>Web Site:</b>                 | <a href="http://www.netiq.com">www.netiq.com</a>   |

## Contacting Technical Support

For specific product issues, contact our Technical Support team.

|   |  |
|---|--|
| <b>Worldwide:</b>                       | <a href="http://www.netiq.com/support/contactinfo.asp">www.netiq.com/support/contactinfo.asp</a> |
| <b>North and South America:</b>         | 1-713-418-5555   |
| <b>Europe, Middle East, and Africa:</b> | +353 (0) 91-782 677  |
| <b>Email:</b>                           | <a href="mailto:support@netiq.com">support@netiq.com</a>   |
| <b>Web Site:</b>                        | <a href="http://www.netiq.com/support">www.netiq.com/support</a>                                 |

## Contacting Documentation Support

Our goal is to provide documentation that meets your needs. The documentation for this product is available on the NetIQ Web site in HTML and PDF formats on a page that does not require you to log in. If you have suggestions for documentation improvements, click **Add Comment** at the bottom of any page in the HTML version of the documentation posted at [www.netiq.com/documentation](http://www.netiq.com/documentation). You can also email [Documentation-Feedback@netiq.com](mailto:Documentation-Feedback@netiq.com). We value your input and look forward to hearing from you.

## Contacting the Online User Community

NetIQ Communities, the NetIQ online community, is a collaborative network connecting you to your peers and NetIQ experts. By providing more immediate information, useful links to helpful resources, and access to NetIQ experts, NetIQ Communities helps ensure you are mastering the knowledge you need to realize the full potential of IT investments upon which you rely. For more information, visit [community.netiq.com](http://community.netiq.com).



---

# 1 Understanding the WorkOrder Driver

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 15](#)
- ♦ [Section 1.4, "Terminology," on page 15](#)

## 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 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 10](#)
- ♦ [Section 1.1.2, "Publisher Channel Functions," on page 11](#)

## 1.1.1 Subscriber Channel Functions

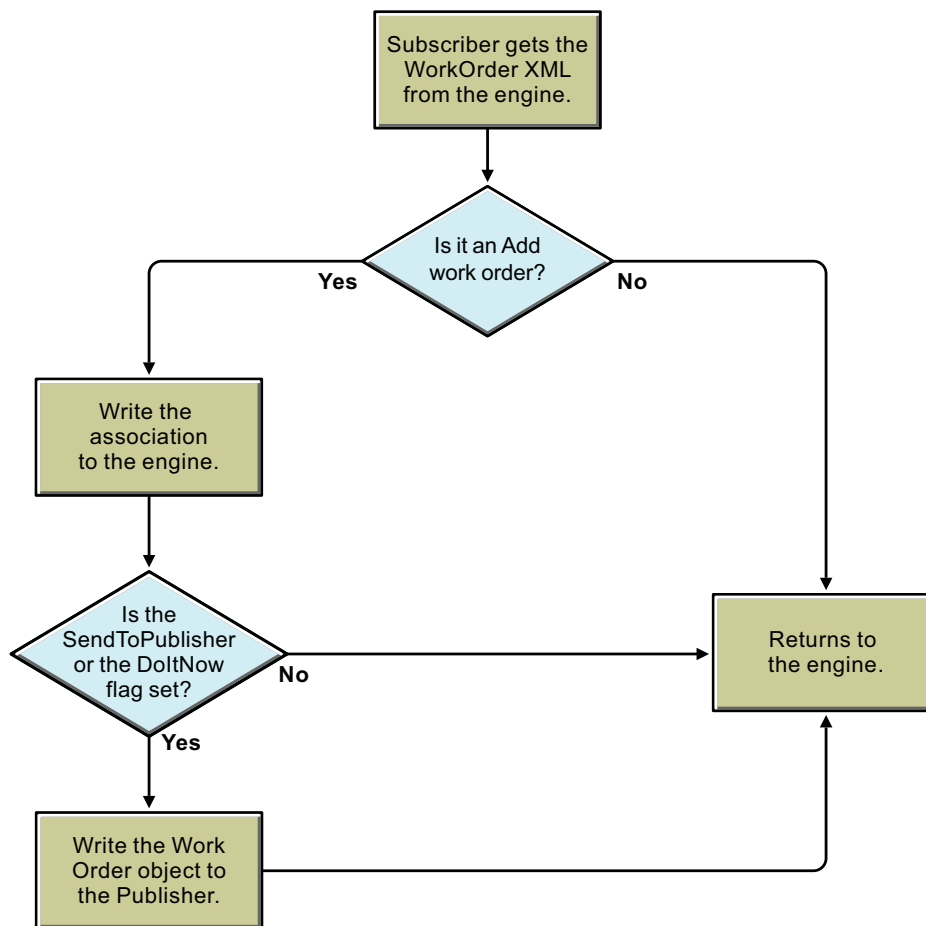
This section provides a basic understanding of the functions the Subscriber channel performs in the WorkOrder driver.

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-nwoContent
- ♦ DirXML-nwoStatus
- ♦ DirXML-DoltNow 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

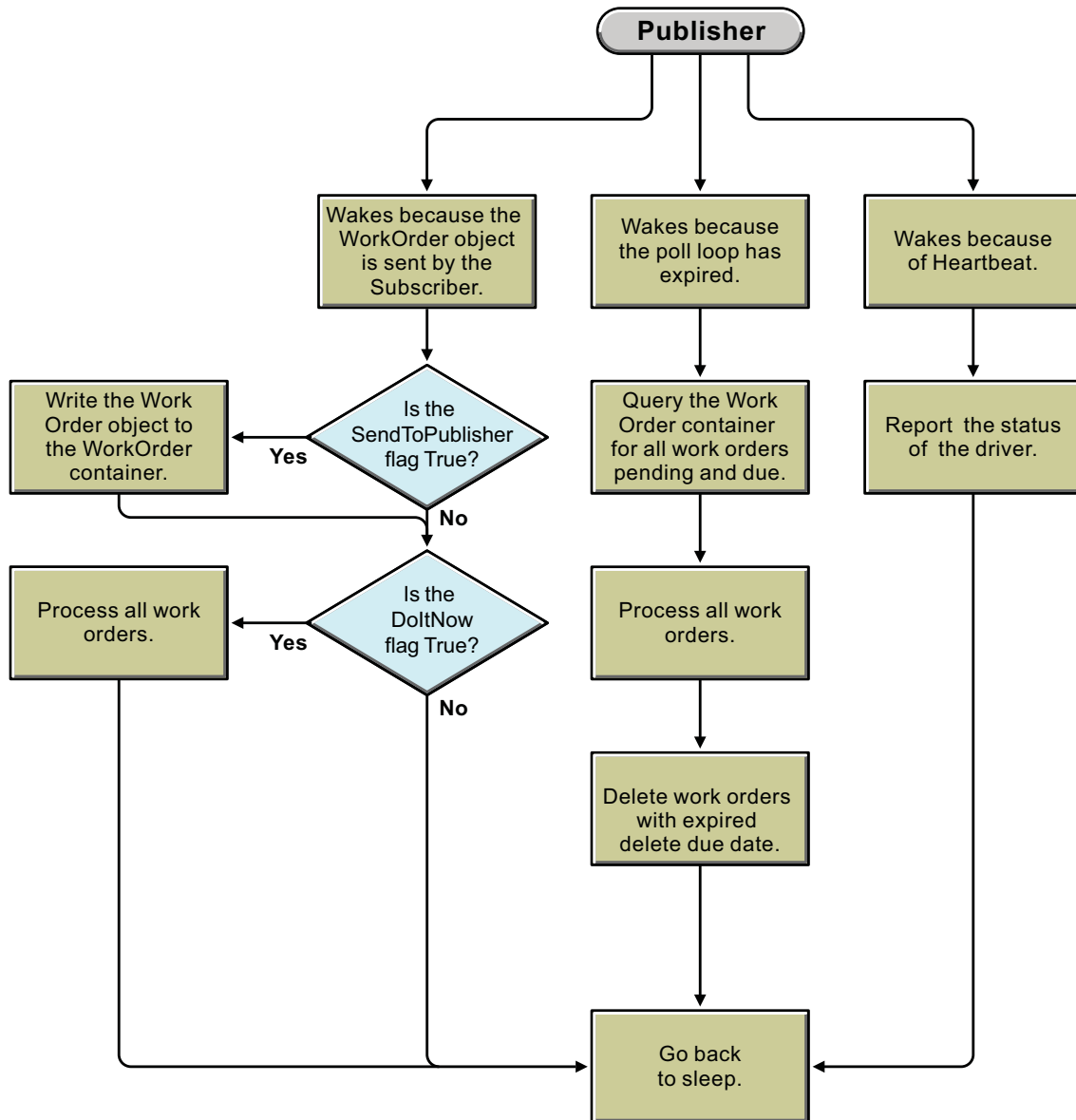
This section reviews the functions of the Publisher channel.

- ♦ [“The Publisher Channel Wakes Up” on page 11](#)
- ♦ [“How the Publisher Channel Processes Work Orders” on page 13](#)
- ♦ [“How the Publisher Channel Deletes Work Orders” on page 14](#)

### The Publisher Channel Wakes Up

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

Figure 1-2 Publisher Channel Configuration



- ♦ 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 DoltNow flag is set to True, the work order is processed immediately.
- ♦ 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.
  1. 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 13.
  2. The Publisher channel queries all work orders for expired DeleteDueDates. See [“How the Publisher Channel Deletes Work Orders”](#) on page 14.
- ♦ 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-nwoContent` attribute of the `WorkToDo` object contains the value of the `DirXML-nwoContent` attribute of the `WorkOrder` object. The default configuration does not do

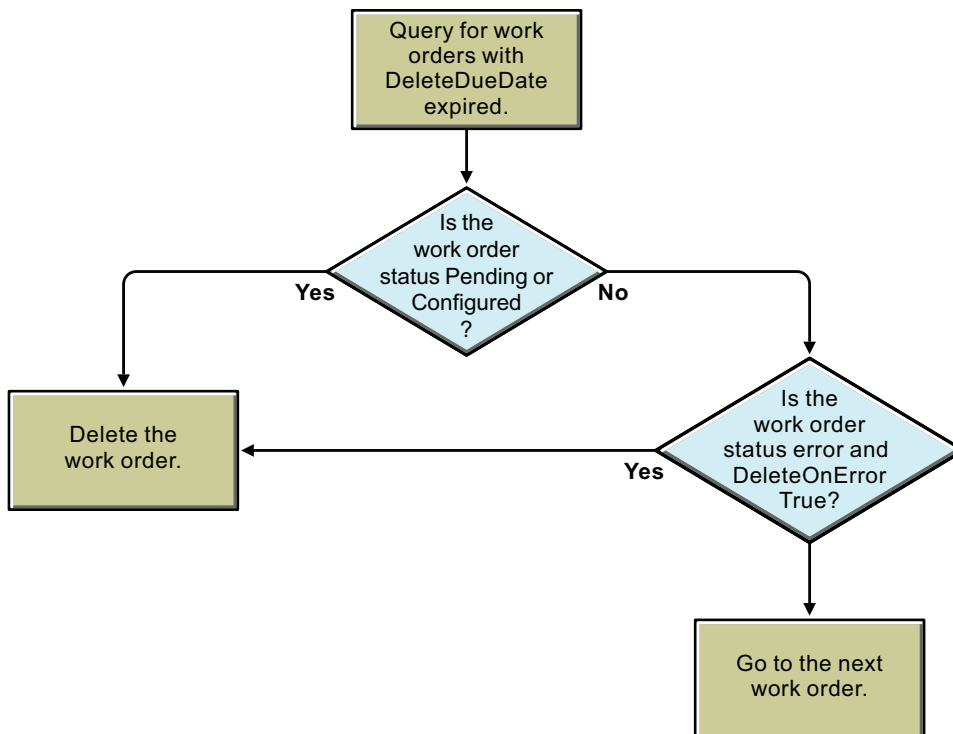
anything else with the WorkToDo object. A policy could use the WorkToDo object to process the work order. For example, the content attribute might contain the DN of a user object whose LogOnDisabled flag should be changed from True to False at the due date.

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 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 [System Requirements for the Identity Manager Engine](#) in the *NetIQ Identity Manager Setup Guide*.

**Remote Platforms:** The WorkOrder driver works on all the platforms supported by the Remote Loader. See [System Requirements for the Remote Loader](#) in the *NetIQ Identity Manager Setup Guide*.

**Entitlements:** The WorkOrder driver does not support 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.





---

# 2 Implementation Checklist

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 19</a> .                            |
| <input type="checkbox"/> Create a new WorkOrder driver<br><br>or<br><br><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 Object," on page 21</a> .<br><br>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.<br><br>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.<br><br>For instructions, see <a href="#">Chapter 7, "Creating and Managing Work Orders," on page 35</a> .  |

---



---

# 3 Installing Driver Files

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 packages. It does not create the driver in the Identity Vault (see [Chapter 4, "Creating a New Driver Object," on page 21](#)) 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 [Preparing to Install the Engine, Drivers, and Plug-ins](#) in the [NetIQ Identity Manager Setup 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 Remote Loader](#) in the [NetIQ Identity Manager Setup Guide](#).



---

# 4 Creating a New Driver Object

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 19](#)), you can create the driver in the Identity Vault. You do so by installing the driver packages 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 21](#)
- ♦ [Section 4.2, “Creating the Driver Object in Designer,” on page 21](#)
- ♦ [Section 4.3, “Activating the Driver,” on page 25](#)
- ♦ [Section 4.4, “Adding Packages to an Existing 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 Object in Designer

You create the WorkOrder driver object 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 Current Driver Packages,” on page 22](#)
- ♦ [Section 4.2.2, “Installing the Driver Packages,” on page 22](#)
- ♦ [Section 4.2.3, “Configuring the Driver Settings,” on page 23](#)
- ♦ [Section 4.2.4, “Deploying the Driver Object,” on page 24](#)
- ♦ [Section 4.2.5, “Starting the Driver,” on page 24](#)

---

**NOTE:** You should not create driver objects by using the new Identity Manager 4.0 and later configuration files through iManager. This method of creating driver objects is no longer supported. To create drivers, you now need to use the new package management features provided in Designer.

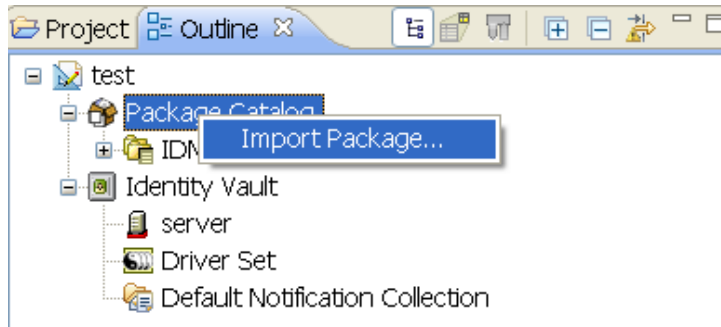
---

## 4.2.1 Importing the Current Driver Packages

The driver packages contain the items required to create a driver, such as policies, entitlements, filters, and Schema Mapping policies. These packages are only available in Designer and can be updated after they are initially installed. You must have the most current version of the packages in the Package Catalog before you can create a new driver object.

To verify that you have the most recent version of the driver packages in the Package Catalog:

- 1 Open Designer.
- 2 In the toolbar, click *Help > Check for Package Updates*.
- 3 Click *OK* to update the packages  
or  
Click *OK* if the packages are up-to-date.
- 4 In the Outline view, right-click the Package Catalog.
- 5 Click *Import Package*.



- 6 Select any WorkOrder driver packages  
or  
Click *Select All* to import all of the packages displayed.  
By default, only the base packages are displayed. Deselect *Show Base Packages Only* to display all packages.
- 7 Click *OK* to import the selected packages, then click *OK* in the successfully imported packages message.
- 8 After the current packages are imported, continue with [Section 4.2.2, "Installing the Driver Packages,"](#) on page 22.

## 4.2.2 Installing the Driver Packages

After you have imported the current driver packages into the Package Catalog, you can install the driver packages to create a new driver object.

- 1 In Designer, open your project.
- 2 In the Modeler, right-click the driver set where you want to create the driver, then click *New > Driver*.
- 3 Select *WorkOrder Base*, then click *Next*.
- 4 On the Driver Information page, specify a name for the driver, then click *Next*.

- 5 On the Install WorkOrder Base page, specify the name of the container that holds the WorkOrder objects, then click *Next*.
- 6 Fill in the following fields for Remote Loader information:
  - Connect To Remote Loader:** Select *Yes* or *No* to determine if the driver will use the Remote Loader. For more information, see [Configuring the Remote Loader and Drivers](#) in the *NetIQ Identity Manager Setup Guide*.
  - If you select *No*, skip to [Step 7](#). If you select *Yes*, use the following information to complete the configuration of the Remote Loader, then click *Next*.
  - Host Name:** Specify the IP address or DNS name of the server where the Remote Loader is installed and running.
  - Port:** Specify the port number for this driver. Each driver connects to the Remote Loader on a separate port. The default value is 8090.
  - Remote Loader Password:** Specify a password to control access to the Remote Loader. It must be the same password that is specified as the Remote Loader password on the Remote Loader.
  - Driver Password:** Specify a password for the driver to authenticate to the Metadirectory server. It must be the same password that is specified as the Driver Object Password on the Remote Loader.
- 7 Review the summary of tasks that will be completed to create the driver, then click *Finish*.
- 8 After you have installed the driver, you must change the configuration for your environment. Proceed to [Section 4.2.3, "Configuring the Driver Settings,"](#) on page 23.


## 4.2.3 Configuring the Driver Settings

After installing the driver packages, 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.
- 3 After you configure the driver, continue with [Section 4.2.4, "Deploying the Driver Object,"](#) on page 24.

## 4.2.4 Deploying the Driver Object

After the driver object 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 4](#); otherwise, specify the following information to authenticate, then click *OK*:

**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 Read the deployment summary, then click *Deploy*.
- 5 Read the message, then click *OK*.
- 6 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.

**6a** Click *Add*, then browse to and select the object with the correct rights.

**6b** Click *OK* twice.

- 7 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.

**7a** Click *Add*, then browse to and select the user object you want to exclude, then click *OK*.

**7b** Repeat [Step 7a](#) for each object you want to exclude.


**7c** Click *OK*.

- 8 Click *OK*.

## 4.2.5 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 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.

If driver activation has expired, the following error message is displayed in the ndstrace window:


```
DirXML Log Event -----
Driver: \META-RHEL6\system\DriverSet\edirDriver-BulkOperations
Channel: Subscriber
Status: Error
Message: Code(-9075) Shutting down because DirXML engine evaluation period
```

To use the driver, you must reactivate it.

For information on activation, refer to “[Activating Identity Manager](#)” in the *NetIQ Identity Manager Setup Guide*.

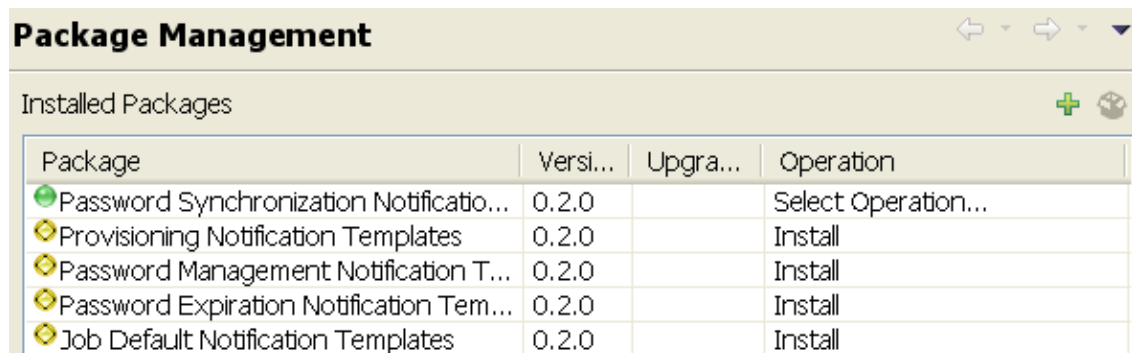
## 4.4 Adding Packages to an Existing Driver

You can add new functionality to an existing driver by adding new packages to an existing driver.

- 1 Right-click the driver, then click *Properties*.
- 2 Click *Packages*, then click the *Add Packages* icon .
- 3 Select the packages to install. If the list is empty, there are no available packages to install.
- 4 (Optional) Deselect the *Show only applicable package versions* option, if you want to see all available packages for the driver, then click *OK*.

This option is only displayed on drivers. By default, only the packages that can be installed on the selected driver are displayed.

- 5 Click *Apply* to install all of the packages listed with the *Install* operation.



- 6 (Conditional) Fill in the fields with appropriate information to install the package you selected for the driver, then click *Next*.
- 7 Read the summary of the installation, then click *Finish*.
- 8 Click *OK* to close the Package Management page after you have reviewed the installed packages.

**Package Management** ← - → - ▾

Installed Packages + 🗑️

| Package                                   | Versi... | Upgra... | Operation           |
|---|----------|----------|---------------------|
| 🟢 Job Default Notification Templates      | 0.2.0    |          | Select Operation... |
| 🟢 Password Expiration Notification Tem... | 0.2.0    |          | Select Operation... |
| 🟢 Password Management Notification T...   | 0.2.0    |          | Select Operation... |
| 🟢 Password Synchronization Notificatio... | 0.2.0    |          | Select Operation... |
| 🟢 Provisioning Notification Templates     | 0.2.0    |          | Select Operation... |

- 9 Repeat [Step 1](#) through [Step 8](#) for each driver where you want to add the new packages.

---

# 5 Upgrading an Existing Driver

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

- ♦ [Section 5.1, "Supported Upgrade Paths," on page 27](#)
- ♦ [Section 5.2, "What's New in Version 4.5," 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 4.0 or later is not supported.

## 5.2 What's New in Version 4.5

Version 4.5 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 Installed Packages](#)" in the *NetIQ Designer for Identity Manager Administration Guide*.



---

# 6 Customizing the Driver

After you create a new WorkOrder driver by installing the driver packages, the driver processes WorkOrder objects from the Identity Vault to create WorkToDo objects. This is all the WorkOrder driver does when it uses 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. The following sections illustrate two possible solutions for customizing your driver:

- ◆ [Section 6.2, “Human Resource Example Using an HR Driver,” on page 30](#)
- ◆ [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 10](#) and [Section 1.1.2, “Publisher Channel Functions,” on page 11](#).

- ◆ [Section 6.1.1, “Subscriber Channel,” on page 29](#)
- ◆ [Section 6.1.2, “Publisher Channel,” on page 30](#)

### 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:

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

| 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>◆ nwoStatus</li> <li>◆ nwoSendToPublisher</li> <li>◆ nwoDoltNow</li> <li>◆ nwoContent</li> <li>◆ nwoType</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,”</a> on page 49.</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 packages.   |
| Schema Mapping         | Maps the eDirectory namespace to the Work Order namespace.  |
| Output Transformation  | Not used in the packages.   |

## 6.1.2 Publisher Channel

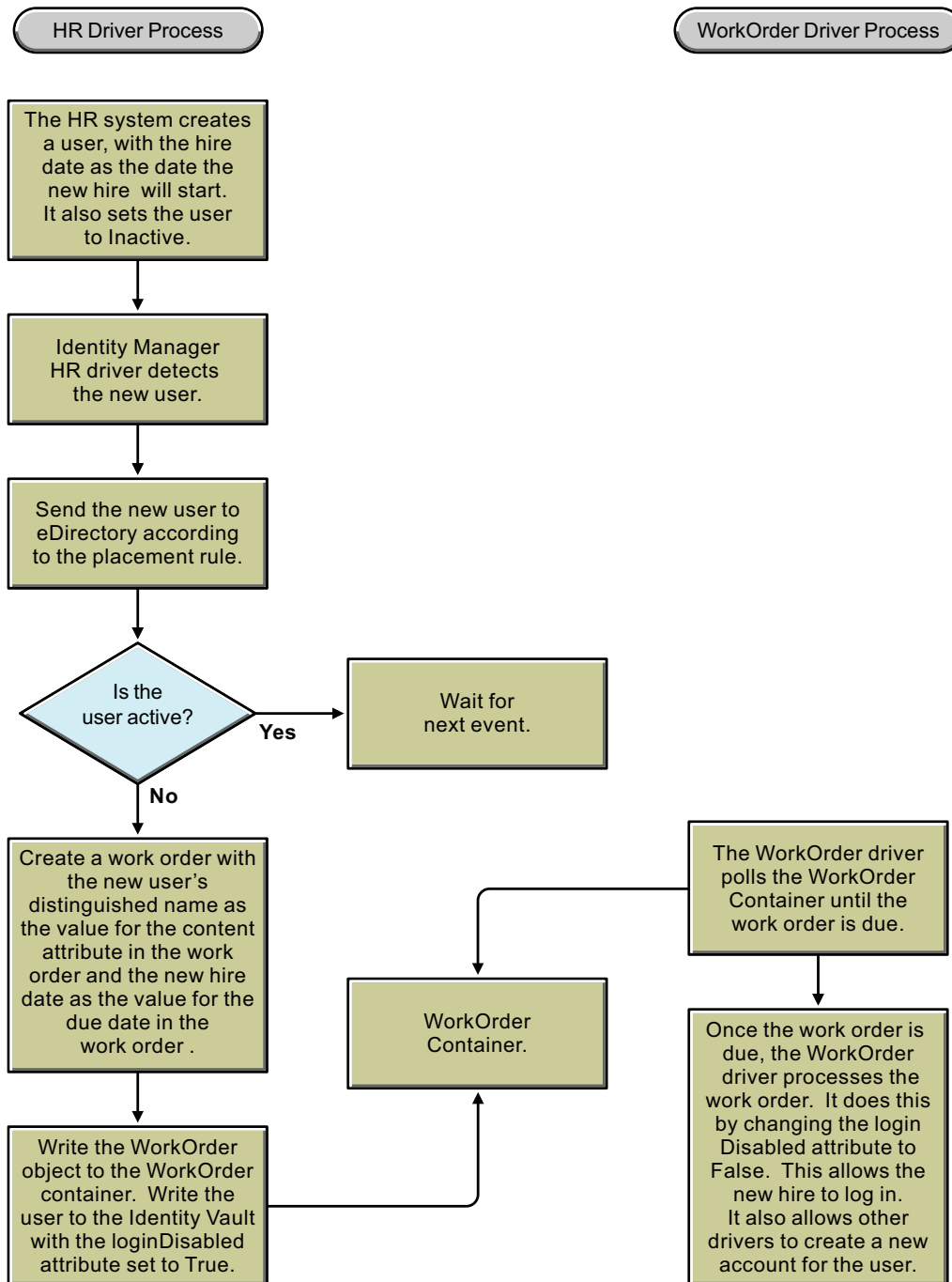
The following table lists the rules and policies used to configure the Publisher channel:

| Rule or Policy         | What it does  |
|------------------------|---|
| Schema Mapping         | Maps the Work Order driver namespace to the eDirectory namespace.   |
| Event Transformation   | Not used in the packages.   |
| Publisher Filter       | Allows only events for WorkOrder objects to be processed.   |
| Matching Rule          | Not used in the packages.   |
| 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 packages.   |

## 6.2 Human Resource Example Using an HR Driver

The following example illustrates how the WorkOrder driver can be used with an HR driver (SAP, PeopleSoft, and so forth) 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 that 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 as 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 or rules 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” on page 32
- ◆ “Filter” on page 32
- ◆ “Command Transformation Policy” on page 32

### 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/idm40drivers/work_order/samples/hr-drv-schema-map.xml) ([http://www.novell.com/documentation/idm40drivers/work\\_order/samples/hr-drv-schema-map.xml](http://www.novell.com/documentation/idm40drivers/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-filter.xml](http://www.novell.com/documentation/idm40drivers/work_order/samples/hr-drv-filter.xml) ([http://www.novell.com/documentation/idm40drivers/work\\_order/samples/hr-drv-filter.xml](http://www.novell.com/documentation/idm40drivers/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-nwoContent attribute. You can view the sample at [hr-drv-cmd-transform.xml](http://www.novell.com/documentation/idm40drivers/work_order/samples/hr-drv-cmd-transform.xml) ([http://www.novell.com/documentation/idm40drivers/work\\_order/samples/hr-drv-cmd-transform.xml](http://www.novell.com/documentation/idm40drivers/work_order/samples/hr-drv-cmd-transform.xml)).

The policy also puts the DirXML-DueDate from the user into the WorkOrder object DirXML-DueDate and then sets the work order status to Pending.

There is a second policy that sets the status of the work order. You can view the sample at [hr-drv-cmd-transform2.xml](http://www.novell.com/documentation/idm40drivers/work_order/samples/hr-drv-cmd-transform2.xml) ([http://www.novell.com/documentation/idm40drivers/work\\_order/samples/hr-drv-cmd-transform2.xml](http://www.novell.com/documentation/idm40drivers/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-nwoContent attribute. It then sets the user's 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.

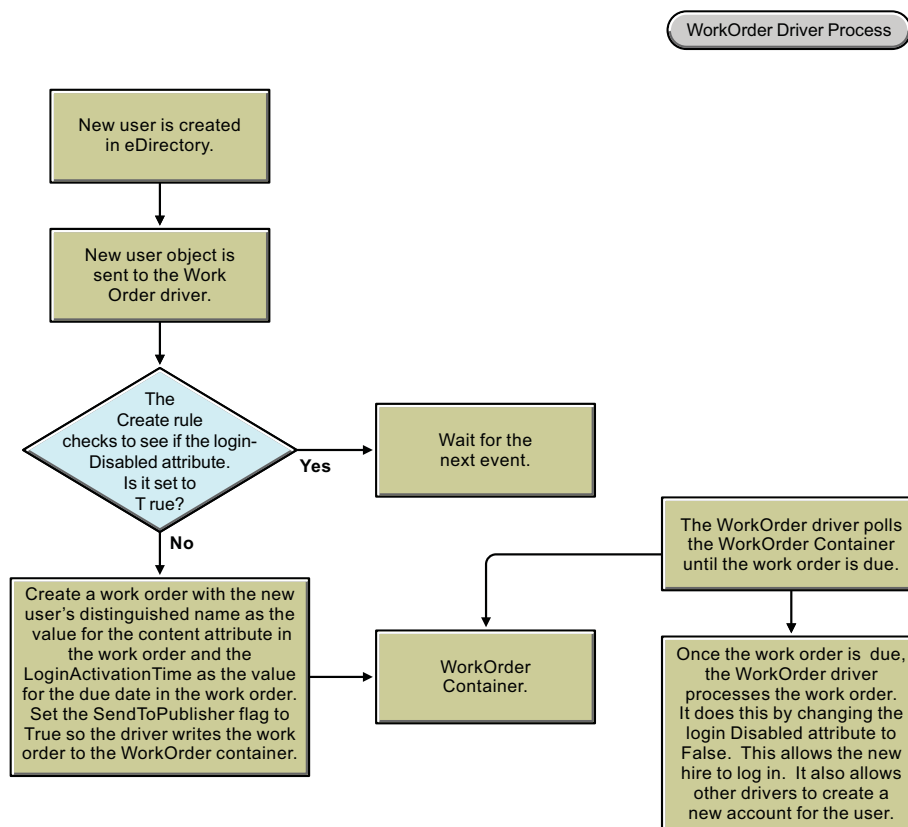
---

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 correctly report the results of the write. You can view the sample at [hr-wo-drv-pub-cmd-transform.xml](http://www.novell.com/documentation/idm40drivers/work_order/samples/hr-wo-drv-pub-cmd-transform.xml) ([http://www.novell.com/documentation/idm40drivers/work\\_order/samples/hr-wo-drv-pub-cmd-transform.xml](http://www.novell.com/documentation/idm40drivers/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 into 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 or rules show how to implement the sample configuration:

- ◆ [Section 6.3.1, "Filter Additions,"](#) on page 34
- ◆ [Section 6.3.2, "Subscriber Create Rule,"](#) on page 34

- ♦ [Section 6.3.3, “Subscriber Command Transform,”](#) on page 34
- ♦ [Section 6.3.4, “Work Order E-Mail Notification of Work Order Completion,”](#) on page 34

### 6.3.1 Filter Additions

These 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/idm40drivers/work\\_order/samples/wo-filter.xml](http://www.novell.com/documentation/idm40drivers/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/idm40drivers/work\\_order/samples/wo-create.xml](http://www.novell.com/documentation/idm40drivers/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-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, 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/idm40drivers/work\\_order/samples/wo-sub-cmd-transform.xml](http://www.novell.com/documentation/idm40drivers/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/idm40drivers/work\\_order/samples/wo-pub-cmd-transform.xml](http://www.novell.com/documentation/idm40drivers/work_order/samples/wo-pub-cmd-transform.xml)).

---

# 7 Creating and Managing Work Orders

There are two ways to create work orders.

- ♦ [Section 7.1, “Using Drivers to Create Work Orders,” on page 35](#)
- ♦ [Section 7.2, “Using iManager to Create Work Orders,” 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 of how to use a driver to create a work order.

## 7.2 Using iManager to Create Work Orders

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 WorkOrder 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 *WorkOrder Management* to display the WorkOrder 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 you create 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 NetIQ 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.



---

# 8 Managing the Driver

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 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 [NetIQ Identity Manager Entitlements Guide](#).





---

# 9 Troubleshooting the Driver

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



---

# A Driver Properties


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 *NetIQ Identity Manager 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 a Designer icon.

- ♦ [Section A.1, “Driver Configuration,” on page 43](#)
- ♦ [Section A.2, “Global Configuration Values,” on page 46](#)

## 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 Options,” on page 45](#)
- ♦ [Section A.1.5, “Driver Parameters,” on page 45](#)
- ♦ [Section A.1.6, “ECMAScript,” on page 46](#)
- ♦ [Section A.1.7, “Global Configuration,” on page 46](#)

## A.1.1 Driver Module

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

**Java:** Used to specify the name of the Java class that is instantiated for the shim component of the driver. This class can be located in the `classes` directory as a class file, or in the `lib` directory as a `.jar` file. If this option is selected, the driver is running locally.

The java class name is:

```
com.novell.nds.dirxml.driver.workorder.WorkOrderDriverShim
```

**Native:** This option is not used with the WorkOrder driver.

**Connect to Remote Loader:** Used when the driver is connecting remotely to the connected system. Designer includes two suboptions:

- ♦ *Remote Loader Client Configuration for Documentation:* Includes information on the Remote Loader client configuration when Designer generates documentation for the driver.
- ♦ *Driver Object Password:* Specifies a password for the Driver object. If you are using the Remote Loader, you must enter a password on this page. Otherwise, the remote driver does not run. The Remote Loader uses this password to authenticate itself to the remote driver shim.

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

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

## A.1.3 Authentication

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

**Authentication ID:** Specify a user application ID. This ID is used to pass Identity Vault subscription information to the application.

Example: `Administrator`

**Authentication Context:** Specify the IP address or name of the server the application shim should communicate with.

**Remote Loader Connection Parameters:** Used only if the driver is connecting to the application through the Remote Loader. The parameter to enter is `hostname=xxx.xxx.xxx.xxx port=xxxx kmo=certificatename`, when the host name is the IP address of the application server running the Remote Loader server and the port is the port the Remote Loader is listening on. The default port for the Remote Loader is 8090.

The `kmo` entry is optional. It is only used when there is an SSL connection between the Remote Loader and the Metadirectory engine.

Example: `hostname=10.0.0.1 port=8090 kmo=IDMCertificate`

**Application Password:** Specify the password for the user object listed in the *Authentication ID* field.

**Remote Loader Password:** Used only if the driver is connecting to the application through the Remote Loader. The password is used to control access to the Remote Loader instance. It must be the same password specified during the configuration of the Remote Loader on the connected system.

**Cache limit (KB):** Specify the maximum event cache file size (in KB). If it is set to zero, the file size is unlimited.

Click *Unlimited* to set the file size to unlimited in Designer.

## A.1.4 Startup Options

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

**Auto start:** The driver starts every time the Identity Manager server is started.

**Manual:** The driver does not start when the Identity Manager server is started. The driver must be started through Designer or iManager.

**Disabled:** The driver has a cache file that stores all of the events. When the driver is set to Disabled, this file is deleted and no new events are stored in the file until the driver state is changed to Manual or Auto Start.

**Do not automatically synchronize the driver:** This option only applies if the driver is deployed and was previously disabled. If this is not selected, the driver re-synchronizes the next time it is started.

## A.1.5 Driver Parameters

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

**Driver Name:** The actual name you want to use for the driver.

**WorkOrders Container:** The name of the container where WorkOrder objects and WorkToDo objects are to be stored.

**Poll Interval:** 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.

**Poll Time:** Time of day the Publisher channel checks the WorkOrder container for work orders to be configured. By default, this setting is disabled (*No poll time*) 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 *Edit XML* button, locate the `<definition display-name="Poll Time" id="112" name="polling-time" type="enum">` entry, and change the type from `enum` to `string`. Click *OK* to save the change, then enter the desired time in the Poll Time field. Use the `HH:MM AM/PM` format (for example, 1:15 PM).

**Publisher Heartbeat every Poll Interval:** 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

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

## A.1.7 Global Configuration

Displays an ordered list of Global Configuration objects. The objects contain extension GCV definitions for the driver that Identity Manager loads when the driver is started. You can add or remove the Global Configuration objects, and you can change the order in which the objects are executed.

## 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:

**WorkOrder Container:** This is the WorkOrder container that is specified by the [WorkOrder Container](#) 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*.

---

# B Objects and Attributes

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

- ♦ [Section B.1, “New Objects Used by the Driver,” on page 47](#)
- ♦ [Section B.2, “DoltNow and SendToPublisher Flags,” on page 47](#)

## B.1 New Objects Used by the Driver

The Identity Manager WorkOrder driver uses two new object classes in the Identity Vault to configure work orders and record the results. For a description of a schema for these objects, see [Appendix C, “Schema and Policy Rules For Work Order Management,” on page 49](#).

- ♦ [Section B.1.1, “DirXML-WorkOrder Object,” on page 47](#)
- ♦ [Section B.1.2, “DirXML-WorkToDo Object,” on page 47](#)

### 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 DoltNow, the driver performs it immediately and does not wait for a polling time or time of day. To learn how to use the DoltNow and SendToPublisher flags, see [Section B.2, “DoltNow and SendToPublisher Flags,” on page 47](#).

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 DoltNow and SendToPublisher Flags

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

- ♦ [Section B.2.1, “DoltNow Flag,” on page 48](#)
- ♦ [Section B.2.2, “SendToPublisherFlag,” on page 48](#)

## 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 you can use policies in an automated solution 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.



---

# C Schema and Policy Rules For Work Order Management

As part of the installation of the WorkOrder driver, NetIQ 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 49
- ◆ [Section C.2, “DirXML-WorkToDo Object,”](#) on page 51
- ◆ [Section C.3, “Publisher Placement Rule,”](#) on page 52
- ◆ [Section C.4, “Subscriber Placement Rule,”](#) on page 52
- ◆ [Section C.5, “Subscriber Create Rule,”](#) on page 52

## 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:

| 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               |
| DirXML-nwoDoltNowFlag                        | If this attribute is set to True, the Subscriber channel sends the work order to the Publisher channel to be processed immediately.                        | Boolean            |

| <b>Work Order Attributes (eDirectory Namespace)</b> | <b>Description</b>   | <b>Type</b>        |
|---|--|--------------------|
| DirXML-nwoSendToPublisher                           | If this attribute is 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 nonexistent or empty, it is ignored.  | Distinguished Name |
| DirXML-nwoRepeatInterval                            | 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-nwoRepeatCount                               | Repeats the work order as many times as the number specifies. Use this attribute in association with the DirXML-nwoRepeatInterval attribute.   | Case ignore string |
| DirXML-nwoStatus                                    | Status of the work order.<br><br>Pending: The work order will be processed on the due date.<br><br>Configured: The work order was processed.<br><br>Error: An error occurred when processing.<br><br>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 this attribute is 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                             | If the status is Pending or Configured, this attribute shows the date and time the work order will be deleted.   | Time               |

| <b>Work Order Attributes (eDirectory Namespace)</b> | <b>Description</b>   | <b>Type</b>        |
|---|--|--------------------|
| 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 WorkOrder object that initiated this object.

| <b>WorkToDo Attributes</b> | <b>Description</b>   | <b>Type</b>        |
|----------------------------|--|--------------------|
| 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 |

## 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, or the event cannot be processed further:

| Required Attributes       | Description  | Values or Examples                           |
|---------------------------|--|--|
| DirXML-nwoSendToPublisher | Send the work order 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-nwoDoltNowFlag     | When to perform the work order.  | True or False                                |
| DirXML-nwoContent         | 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                           |