

---

# NetIQ® Identity Manager

## Driver for eDirectory Implementation Guide

March 2018

## **Legal Notice**

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

**Copyright (C) 2018 NetIQ Corporation. All rights reserved.**

---

# Contents

<b>About this Book and the Library</b>	<b>5</b>
<b>About NetIQ Corporation</b>	<b>7</b>
<b>1 Understanding the eDirectory Driver</b>	<b>9</b>
Driver Concepts	9
Key Terms	9
How the eDirectory Driver Works	9
Driver Features	11
Local Platforms	11
Remote Platforms	11
Entitlements	11
Password Synchronization	11
Synchronizing Data	12
<b>2 Installing the Driver Files</b>	<b>13</b>
<b>3 Creating a New Driver Object</b>	<b>15</b>
Creating the Driver Object in Designer	15
Importing the Current Driver Packages	15
Installing the Driver Packages	16
Configuring the Driver	19
Deploying the Driver Object	19
Starting the Driver	20
Activating the Driver	20
Adding Packages to an Existing Driver	21
<b>4 Upgrading an Existing Driver</b>	<b>23</b>
Supported Upgrade Paths	23
What's New in Version 4.7	23
Upgrade Procedure	23
<b>5 Securing Driver Communication</b>	<b>25</b>
Configuring Secure Data Transfers	25
Understanding Secure Connections via the eDirectory Driver	25
Creating Certificates Using Designer	26
Establishing Secure Connections Using KMO	29
Establishing Secure Connections Using Keystore	33
Configuring Authentication Between Drivers	35

<b>6</b>	<b>Synchronizing Passwords</b>	<b>37</b>
<b>7</b>	<b>Managing the Driver</b>	<b>39</b>
<b>8</b>	<b>Troubleshooting</b>	<b>41</b>
	Troubleshooting Driver Processes . . . . .	41
	Synchronizing eDirectory Objects in a Linux High Availability Setup . . . . .	41
	JCException while Synchronizing a Password . . . . .	41
<b>9</b>	<b>Known Issues</b>	<b>43</b>
	Mutual Authentication Feature is not Working. . . . .	43
<b>A</b>	<b>Driver Properties</b>	<b>45</b>
	Driver Configuration . . . . .	45
	Driver Module. . . . .	45
	Driver Object Password . . . . .	46
	Authentication . . . . .	46
	Startup Option . . . . .	47
	Driver Parameters . . . . .	48
	ECMAScript . . . . .	49
	Global Configurations. . . . .	49
	Global Configuration Values . . . . .	50
	Default Configuration . . . . .	50
	Entitlements . . . . .	51
	Password Synchronization . . . . .	53
	Account Tracking . . . . .	54
	Managed System Information . . . . .	54
<b>B</b>	<b>Synchronized Attributes</b>	<b>57</b>
<b>C</b>	<b>Trace Levels</b>	<b>59</b>

# About this Book and the Library

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

## Intended Audience

This book provides information for individuals responsible for understanding administration concepts and implementing a secure, distributed administration model.

## 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).

## Other Information in the Library

For more information about the library for Identity Manager, see the [Identity Manager documentation website](#).



# 1 Understanding the eDirectory Driver

The Identity Manager Driver for eDirectory synchronizes objects and attributes between different eDirectory trees.

This driver is unique among all other Identity Manager drivers. Because you are synchronizing data between eDirectory trees, you will always have two drivers installed, each in its own tree. The driver in one tree communicates with the driver in the other tree.

- ♦ [“Driver Concepts” on page 9](#)
- ♦ [“Driver Features” on page 11](#)

## Driver Concepts

- ♦ [“Key Terms” on page 9](#)
- ♦ [“How the eDirectory Driver Works” on page 9](#)

## Key Terms

**Driver:** A set of policies, filters, and objects that act as the connector between an Identity Vault and the driver shim.

This software enables an application to publish events from an application to the directory, enables an application to subscribe to events from the directory, and synchronizes data between the directory and applications.

To establish a connection between the Metadirectory engine and an Identity Vault, you specify the driver's configuration and connection parameters, policies, and filter values.

**Driver object:** A collection of channels, policies, rules, and filters that connect an application to an Identity Vault that is running Identity Manager.

Each driver performs different tasks. Policies, rules, and filters tell the driver how to manipulate the data to perform those tasks.

The Driver object displays information about the driver's configuration, policies, and filters. This object enables you to manage the driver and provide eDirectory management of the driver shim parameters.

**Driver shim:** A Java file (`NdsToNds.jar`) loaded directly by Identity Manager. Communicates event changes to be sent from the Identity Manager Driver for eDirectory to an Identity Vault, communicates changes from the Identity Vault to the Identity Manager Driver for eDirectory, and operates as the link that connects the Identity Vault and the Identity Vault Driver object.

**Identity Vault.** A hub, with applications and directories publishing their changes to it. The Identity Vault then sends changes to the applications and directories that have subscribed for them. This results in two main flows of data: the Publisher channel and the Subscriber channel.

## How the eDirectory Driver Works

Channels, filters, and policies control data flow.

**Publisher and Subscriber Channels:** The eDirectory driver is installed and configured in two trees. The driver's Publisher channel in TreeA communicates with the driver's Subscriber channel in TreeB. Conversely, the driver's Publisher channel in TreeB communicates with the driver's Subscriber channel in TreeA.

**Filters:** Identity Manager uses filters to control which objects and attributes are shared. The default filter configurations for the eDirectory driver allow objects and attributes to be shared. For a list of synchronized attributes, see [Appendix B, "Synchronized Attributes,"](#) on page 57.

**Policies:** Identity Manager uses policies to control data synchronization between the eDirectory driver and the Identity Vaults.

The driver does not have any direct communication with eDirectory. It communicates with eDirectory via the Identity Manager engine. The driver is usually configured in pairs with an instance of the driver configured in each of the two trees being synchronized. The Subscriber of the driver instance in each tree connects to the Publisher of the driver instance in the other tree via a TCP connection. The connection is formed on demand when a driver instance needs to communicate with the other driver instance. When the connection is established, it remains intact until the connection is broken or one of the driver instances is stopped. If there is a broken connection or an attempt to connect to the other instance fails, the Subscriber channel issues a retry status to the Identity Manager engine and then attempts to reconnect when the Identity Manager engine resends the event. The port used for the connections is configurable and can be different for Subscriber to Publisher pair.

The Subscriber of each driver instance acts primarily as an event source for the Publisher of the other driver instance which in turn is the event source for the Publisher channel of the Identity Manager engine. Events provided by the Subscriber channel of one instance of the driver are passed across the TCP connection unchanged to the Publisher channel of the other instance of the driver which is then passed, mostly unchanged, to the subscriber channel of the Identity Manager engine.

Because the Subscriber channel is primarily an event source for the Publisher channel of the other instance, policies are usually not in place on the Publisher channel of either instance. The primary exception to this rule is that if any custom event filtering is to be done, it is usually more efficient to do that filtering as early in the dataflow as possible, which usually means in the Subscriber Event Transformation policy. The filters are also usually configured such that the filter of the Subscriber filter of one instance is identical (with the exception of the GUID attribute) to the Publisher filter of the other instance.

There are two different protocols used to transport the XML documents between the driver instances.

To guard against undetected loss of connectivity, such as when the network is physically unavailable or the other server crashes without gracefully closing all of its connections, a keep-alive message is periodically sent in each direction across an open connection. This causes an I/O error if the connection breaks in such a way which may be undetectable otherwise. The frequency of the keep-alive packets is configurable. You can tune it to either minimize the network traffic or minimize the amount of time it takes to detect recover from a broken connection.

The driver uses a value generated from the GUID attribute of one of the instances of the driver. The GUID attribute is required to be in the Subscriber filter for each class that is being synchronized. The GUID attribute is not present in the Publisher filter because it is usually not desirable to synchronize a value that is supposed to be globally unique. The GUID attribute is used to generate an association key only when the Publisher channel of a driver passes through an add event for an object that doesn't already have an association key.

The rights to objects are granted to the driver object by standard eDirectory rights management. Sufficient rights must be granted to the objects to perform the desired operations on the desired objects and attributes in each tree.

The connections between the two driver instances can be either clear or secured by SSL. Configuration of SSL requires the creation of a server certificate for each publisher that is signed by a certificate authority that is trusted by the Subscriber channel. For more information, see [Chapter 5, “Securing Driver Communication,” on page 25](#).

## Driver Features

- ◆ [“Local Platforms” on page 11](#)
- ◆ [“Remote Platforms” on page 11](#)
- ◆ [“Entitlements” on page 11](#)
- ◆ [“Password Synchronization” on page 11](#)
- ◆ [“Synchronizing Data” on page 12](#)

### Local Platforms

The eDirectory driver runs in any Identity Manager installation. See [“Implementation Checklist” in the \*NetIQ Identity Manager Setup Guide for Linux\*](#) or [“Implementation Checklist” in the \*NetIQ Identity Manager Setup Guide for Windows\*](#).

### Remote Platforms

The eDirectory driver supports remote connections without the Remote Loader. The driver does not use the Remote Loader because the driver in one tree communicates directly with the driver in the other tree.

### Entitlements

The basic driver configuration supports entitlements. When entitlements are enabled, the driver does the following actions by default:

- ◆ Adds User object accounts
- ◆ Removes User object accounts
- ◆ Adds members of the distribution list
- ◆ Removes members of the distribution list

The driver support entitlements you create if supporting policies are provided for implementing them. For more information about entitlements, see the [NetIQ Identity Manager Entitlements Guide](#).

---

**IMPORTANT:** In the driver filter, select the **Application** option in **Merge Authority** for the loginDisabled attribute in the eDirectory driver that does not have an entitlement.

---

### Password Synchronization

The eDirectory driver supports password synchronization via Universal Password. If desired, you can also use the older form of password synchronization (Public/Private key pair or NDS password). For more information, see [Chapter 6, “Synchronizing Passwords,” on page 37](#).

## Synchronizing Data

The eDirectory driver synchronizes data between two Identity Vaults or trees. The driver can run anywhere that a Identity Manager server is running.

# 2 Installing the Driver Files

If you are synchronizing information between TreeA and TreeB, you must install the Metadirectory engine and eDirectory driver on eDirectory servers in both trees. Therefore, the installation must be completed twice—once for the Metadirectory engine and eDirectory driver in TreeA and once in TreeB.

The eDirectory servers where you install the driver must hold master or read/write replicas of the objects you want synchronized between the two trees.

The installation program extends the Identity Vault (eDirectory) schema and installs the driver shim. It does not create the driver in the Identity Vault (see [Chapter 3, “Creating a New Driver Object,”](#) on page 15) or upgrade an existing driver’s configuration (see [Chapter 4, “Upgrading an Existing Driver,”](#) on page 23).

The eDirectory driver does not use the Remote Loader because the driver in one tree communicates directly with the driver in the other tree.

The eDirectory driver requires the following:

- ♦ NetIQ Certificate Server running on each server that hosts the driver.
- ♦ A certificate authority (CA) to support SSL encryption between drivers.

NetIQ Certificate Server and the certificate authority are discussed more in [Chapter 5, “Securing Driver Communication,”](#) on page 25.



# 3 Creating a New Driver Object

After the eDirectory driver files are installed on the server where you want to run the driver (see [Chapter 2, “Installing the Driver Files,” on page 13](#)), 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:

- ♦ [“Creating the Driver Object in Designer” on page 15](#)
- ♦ [“Activating the Driver” on page 20](#)
- ♦ [“Adding Packages to an Existing Driver” on page 21](#)

## Creating the Driver Object in Designer

To create the eDirectory driver object, install the driver packages and then modify the configuration to suit your environment. After you create and configure the driver object, you need to deploy it to the Identity Vault and start it.

To connect two trees, you need to complete the following procedures for the drivers that are installed in each Identity Vault.

- ♦ [“Importing the Current Driver Packages” on page 15](#)
- ♦ [“Installing the Driver Packages” on page 16](#)
- ♦ [“Configuring the Driver” on page 19](#)
- ♦ [“Deploying the Driver Object” on page 19](#)
- ♦ [“Starting the Driver” on page 20](#)

---

**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.

---

## Importing the Current Driver Packages

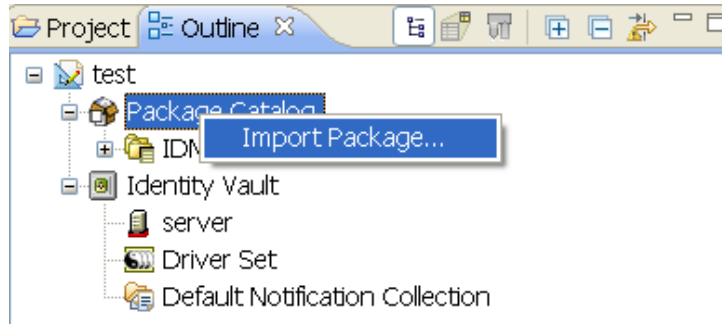
The driver packages contain the items required to create a driver, such as policies, entitlements, filters, and Schema Mapping policies. These packages are only available in Designer. You can upgrade any package that is installed if there is a newer version of the package available. It is recommended to have the latest packages in the Package Catalog before creating a new driver object.

Before creating a driver object in Designer, it is recommended to have all the required packages already imported in the Package Catalog of Designer. Designer prompts you for importing the required packages when it creates the driver object.

To verify you have the most recent version of the driver packages imported into 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 eDirectory 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 [“Installing the Driver Packages” on page 16](#).

## 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.

- 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 **eDirectory Base**, then click **Next**.
- 4 Select the optional features to install for the eDirectory driver. All options are selected by default. The options are:

**Default Configuration:** These packages contain the default configuration information for the eDirectory driver. Always leave this option selected.

**Entitlements:** These packages contain the policies and entitlements required to enable the driver for account creation and management with entitlements. For more information, see the [NetIQ Identity Manager Entitlements Guide](#).

**Password Synchronization:** These packages contain the policies required to enable password synchronization. Leave this option selected if you want to synchronize passwords between the Identity Vaults.



**Data Collection:** These packages contain the policies that enable the driver to collect data for reports. If you are using the Identity Reporting Module, verify that this option is selected. For more information, see the [Administrator Guide to NetIQ Identity Reporting](#).

**Account Tracking:** This group of packages contain the policies that enable account tracking information for reports. If you are using the Identity Reporting Module, verify that this option is selected. For more information, see the [Administrator Guide to NetIQ Identity Reporting](#).

- 5 After selecting the optional packages, click **Next**.
- 6 (Conditional) If there are package dependencies for the packages you selected to install, you must install these dependencies to install the selected packages. Click **OK** to install the Password Synchronization Notification package dependency.
- 7 (Conditional) Click **OK** to install the Common Settings package, if you have not installed any other packages into the selected driver set.
- 8 Click **OK** to install the Advanced Java Class package if you have not installed any other packages into the selected driver set.
- 9 (Conditional) Fill in the following fields on the Common Settings page:

The Common Settings page is displayed only if the Common Settings package is installed as a dependency.

**User Container:** Select the Identity Vault container where the users are added if they don't already exist in the Identity Vault. This value becomes the default value for all drivers in the driver set.

If you want a unique location for this driver, set the value for all drivers on this page. After the driver is created, change the value on the driver's Global Configuration Values page.

**Group Container:** Select the Identity Vault container where the groups are added if they don't already exist in the Identity Vault. This value becomes the default value for all drivers in the driver set.

If you want a unique location for this driver, set the value for all drivers on this page. After the driver is created, change the value on the driver's Global Configuration Values page.

- 10 Click **Next**.
- 11 On the Driver Information page, specify a name for the driver, then click **Next**.
- 12 Fill in the following field to configure the driver:  
**Remote Tree Address and Port:** Specify the hostname or IP address, and port of the server in the remote Identity Vault.
- 13 Click **Next**.
- 14 Fill in the following fields on the eDirectory Default Configuration page:

**eDirectory Publisher Placement type:** Select how the objects are placed in the remote Identity Vault and the local Identity Vault. The options are:

- ◆ **Mirrored:** Mirrors the structure between the remote Identity Vault and the local Identity Vault.

If you choose this option, use the same option for configuring both eDirectory trees you are synchronizing.

This option in the driver configuration synchronizes User, Group, Organization, Country, and Organizational Unit objects. It also mirrors the structure of a subtree in the other tree.

- ◆ **Flat:** All of the objects are placed into a single container.

This option synchronizes User and Group objects and places all users in one container and all groups in another container.

This option is typically used in conjunction with the Department option (or a similar configuration) in the other tree.

This option doesn't create the containers that hold the users and groups. You must create those manually.

- ◆ **Department:** Users are placed in containers named after the department.

This option synchronizes User and Group objects and places all users and groups in a container based on the **Department** field in your management console.

This configuration is typically used in conjunction with the Flat option (or a similar configuration) in the other tree.

This option doesn't create the containers for each department. You must create those manually. They must be the same as the container specified during import.

**Remote Tree Base User Container:** Specify the source container of the user objects in the remote Identity Vault.

**Remote Tree Base Groups Container:** Specify the source container of the group objects in the remote Identity Vault.

15 Click **Next**.

16 (Conditional) Fill in the following fields on the eDirectory Managed System Information page. This page is displayed only if you selected to install the Data Collection and Account Tracking groups of packages.

**Name:** Specify a descriptive name for this Identity Vault. The name is displayed in the reports.

**Description:** Specify a brief description of the this Identity Vault. The description is displayed in the reports.

**Location:** Specify the physical location of this Identity Vault. The location is displayed in the reports.

**Vendor:** Select NetIQ as the vendor of this system. The vendor information is displayed in the reports.

**Version:** Specify the version of this Identity Vault. The version is displayed in the reports.

17 Click **Next**.

18 (Conditional) Fill in the following fields to define the ownership of this Identity Vault. This page is displayed only if you selected to install the Data Collection and Account Tracking groups of packages.

**Business Owner:** Select a user object in the Identity Vault that is the business owner of this Identity Vault. This can only be a user object, not a role, group, or container.

**Application Owner:** Select a user object in the Identity Vault that is the application owner for this Identity Vault. This can only be a user object, not a role, group, or container.

19 Click **Next**.

20 (Conditional) Fill in the following fields to define the classification of the Identity Vault. This page is only displayed if you selected to install the Data Collection and Account Tracking groups of packages.

**Classification:** Select the classification of the Identity Vault. This information is displayed in the reports. The options are:

- ◆ Mission-Critical
- ◆ Vital
- ◆ Not-Critical

- ◆ Other

If you select **Other**, you must specify a custom classification for the Identity Vault.

**Environment:** Select the type of environment the Identity Vault provides. The options are:

- ◆ Development
- ◆ Test
- ◆ Staging
- ◆ Production
- ◆ Other

If you select **Other**, you must specify a custom classification for the Identity Vault.

21 Click **Next**.

22 Review the summary of tasks that will be completed to create the driver, then click **Finish**.

23 After the driver packages are installed, there is additional configuration required for the eDirectory driver. Continue to [“Configuring the Driver” on page 19](#) to configure the driver.

## Configuring the Driver


After installing the driver packages, the eDirectory driver will run. However, the basic configuration might not meet the requirements for your environment. You should complete the following tasks to configure the driver:

- ◆ **Secure the driver connection:** eDirectory drivers communicate via SSL using digital certificates for authentication. You need to set up this secure connection. See [Chapter 5, “Securing Driver Communication,” on page 25](#).
- ◆ **Configure the driver filter:** Modify the driver filter to include the object classes and attributes you want synchronized between the two eDirectory trees. For information about the classes and attributes include in the filter for the basic configuration, see [Appendix B, “Synchronized Attributes,” on page 57](#).
- ◆ **Configure policies:** Modify the policies as needed. Policies should generally be placed only on the Publisher channel, not on the Subscriber channel. The Matching and Placement policies cannot operate correctly on the Subscriber channel because the Subscriber channel is acting primarily as a source of events for the Publisher channel of the other tree.  
You might consider placing an Event Transform or Create Policy on the Subscriber channel to prevent sending unnecessary data across the channel.
- ◆ **Configure password synchronization:** The basic driver configuration is set up to support bidirectional password synchronization through Universal Password. If you don’t want this setup, see [Chapter 6, “Synchronizing Passwords,” on page 37](#).

After completing the configuration tasks, continue with the next section, [Deploying the Driver Object](#).

## Deploying the Driver Object

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:

**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 through the deployment summary, then click **Deploy**.

6 Read the successful 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.

**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**.

## 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 won'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 7, "Managing the Driver,"](#) on [page 39](#).


## Activating the Driver

The Identity Manager driver for eDirectory does not need a separate activation. If you create the driver in a driver set where you have already activated the Identity Manager server and service drivers, the driver inherits the activation from the driver set.

If you create the driver in a driver set that has not been previously activated, the driver will run in the evaluation mode for 90 days. You must activate the driver during the evaluation period; otherwise, the driver will be disabled. If you try to run the driver, `ndstrace` displays an error message indicating that you need to reactivate the driver to use it. For information on activation, refer to [Activating Identity Manager](#) in the *NetIQ Identity Manager Overview and Planning Guide* or [Activating Identity Manager](#) in the *NetIQ Identity Manager Setup Guide for Windows*.

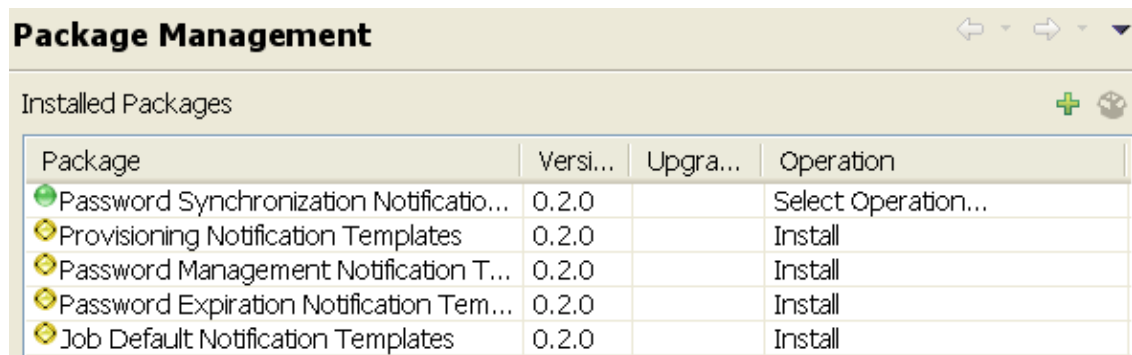
# Adding Packages to an Existing Driver

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

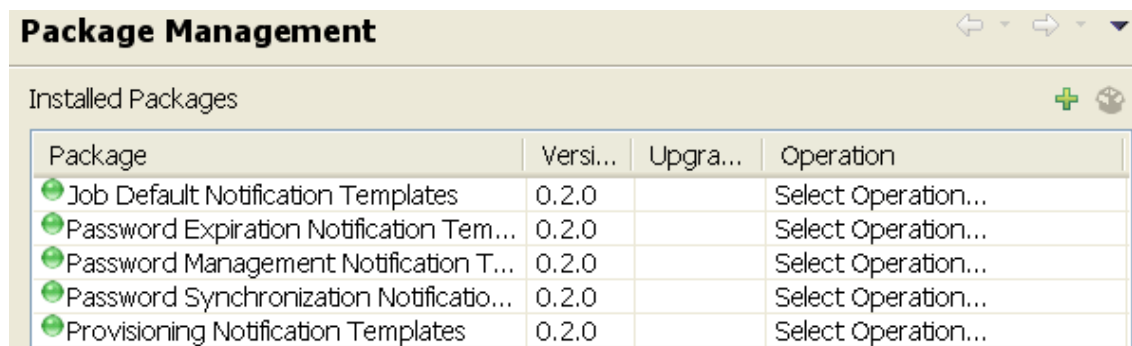
- 1 Right-click the driver, then click **Properties**.
- 2 Click **Packages**, then click the **Add Packages** icon .
- 3 Select the packages to install. If the list is empty, there are no available packages to install.
- 4 (Optional) Deselect the **Show only applicable package versions** option, if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



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



# 4 Upgrading an Existing Driver

The following sections provide information to help you upgrade an existing driver:

- ♦ “Supported Upgrade Paths” on page 23
- ♦ “What’s New in Version 4.7” on page 23
- ♦ “Upgrade Procedure” on page 23

## Supported Upgrade Paths

You can upgrade from any 3.x version of the eDirectory driver. Upgrading a pre-3.x version of the driver directly to version 4.0.3 is not supported.

## What’s New in Version 4.7

This version of the driver does not provide any new features.

## Upgrade Procedure

There is no separate procedure for upgrading an eDirectory driver shim. The driver shim is upgraded with a new version of Identity Manager engine.

To upgrade the installed packages for the driver, perform the following actions in Designer:

- 1 Download the latest available packages.

To configure Designer to automatically read the package updates when a new version of a package is available, click **Windows > Preferences > NetIQ > Package Manager > Online Updates** in Designer. However, if you need to add a custom package to the Package Catalog, you can import the package .jar file. For more information about creating custom packages, see [Developing Packages](#) in the *NetIQ Designer for Identity Manager Administration Guide*.

- 2 Upgrade the installed packages.

**2a** Open the project containing the driver.

**2b** Right-click the driver for which you want to upgrade an installed package, then click **Driver > Properties**.

**2c** Click **Packages**.

If there is a newer version of a package, there is check mark displayed in the Upgrades column.

**2d** Click **Select Operation** for the package that indicates there is an upgrade available.

**2e** From the drop-down list, click **Upgrade**.

**2f** Select the version that you want to upgrade to, then click **OK**.

---

**NOTE:** Designer lists all versions available for upgrade.

---

**2g** Click **Apply**.

**2h** (Conditional) Fill in the fields with appropriate information to upgrade the package, then click **Next**.

Depending on which package you selected to upgrade, you must fill in the required information to upgrade the package.

**2i** Read the summary of the packages that will be installed, then click **Finish**.

**2j** Review the upgraded package, then click **OK** to close the Package Management page.

For detailed information, see the [Upgrading Installed Packages](#) in the *NetIQ Designer for Identity Manager Administration Guide*.

---

**IMPORTANT:** It is a good practice to upgrade one eDirectory driver at a time. However, both drivers must run the same version to ensure feature compatibility.

---



# 5 Securing Driver Communication

To provide security while transmitting information between two Identity Vaults, you must configure the eDirectory driver to communicate with the destination eDirectory driver through an SSL connection.

In addition, you can provide additional security by requiring the two eDirectory drivers to authenticate to one another. Although this is optional, it is strongly recommended.

The following sections explain how to set up SSL and configure driver authentication:

- ♦ “Configuring Secure Data Transfers” on page 25
- ♦ “Configuring Authentication Between Drivers” on page 35

## Configuring Secure Data Transfers

All eDirectory driver communication is secured through SSL. To configure your eDirectory drivers to handle secure data transfers, run the NDS-to-NDS Driver Certificate Wizard in iManager.

You can also configure your eDirectory drivers through Designer. For information about configuring eDirectory drivers through Designer, see Designer for Identity Manager Administration Guide

- ♦ “Understanding Secure Connections via the eDirectory Driver” on page 25
- ♦ “Creating Certificates Using Designer” on page 26
- ♦ “Establishing Secure Connections Using KMO” on page 29
- ♦ “Establishing Secure Connections Using Keystore” on page 33

## Understanding Secure Connections via the eDirectory Driver

The following items can help you understand how secure connections are established when using the eDirectory driver:

- ♦ The driver uses SSL sockets to provide authentication and a secure connection. SSL uses digital certificates to allow the parties to an SSL connection to authenticate one another. Identity Manager in turn uses NetIQ Certificate Server certificates for secure management of sensitive data.
- ♦ To use the driver, you must have the NetIQ Certificate Server running in each tree. We recommend that you use the certificate authority from one of the trees containing the driver to issue the certificates used for SSL. If your tree does not have a certificate authority, you need to create one. You can use an external certificate authority. For information about NetIQ Certificate Server, see the [NetIQ Certificate Server 3.3 Documentation Web site \(http://www.NetIQ.com/documentation/crt33/\)](http://www.NetIQ.com/documentation/crt33/).
- ♦ The NetIQ implementation of SSL that the driver uses is based on NetIQ Secure Authentication Services (SAS) and NTLS for eDirectory. These must be installed and configured on the server where the driver runs. eDirectory usually does this automatically.

- ♦ To configure driver security, it is necessary to create and reference certificates in the eDirectory trees that will be connected using the driver. The two SSL types for securing the connection are Key Material Objects (KMOs) and Keystore.
- ♦ Certificate objects in eDirectory are called KMO because they securely contain both the certificate data including the public key and the private key associated with the certificate.

A minimum of two KMOs (one KMO per tree) must be created for use with the eDirectory drivers. This section explains using a single KMO per tree.

The NDS-to-NDS Driver Certificate Wizard sets up the KMOs.

## Creating Certificates Using Designer

To provide security while transmitting information between two Identity Vaults, you must configure the eDirectory driver to communicate with the destination eDirectory driver through an SSL connection.

In addition, you can provide additional security by requiring the two eDirectory drivers to authenticate to one another. Although this is optional, it is strongly recommended.

The following section explains how to set up SSL and configure driver authentication using designer.

### Establishing Secure Connection and Creating Certificates Using Designer

Perform the following actions to configure two eDirectory drivers and communicate with each other over a secure channel.

1. Open your project in Designer.
2. Select two Identity Vaults from the palette between which you wish to secure a connection.
3. The Driver Configuration Wizard appears. The purpose of the Driver Configuration Wizard is to help you install drivers. For more information on creating a driver, see [Chapter 3, "Creating a New Driver Object," on page 15](#).

To create a driver with packages, select the available base package listed. If there are no packages listed, then the packages are not imported into the package catalog. For more information about importing and installing packages, see [Installing or Upgrading Packages](#).

4. Configure the drivers. For more information on configuring the driver, see ["Configuring the Driver" on page 19](#).

To create a driver with a driver configuration file, click Import Driver Configuration. All of the driver configurations files for the version of your Identity Manager server are listed. For more information about importing a driver configuration file, see [Importing a Driver Configuration File](#).

5. Deploy the driver.
6. Configure the driver settings to communicate the two eDirectory drivers. For more information, see [Configure TLS for eDir-to-eDir drivers](#).

To configure the driver settings, navigate to **Preferences > NetIQ > Identity Manager > > Configuration > eDir-to-eDir SSL/TLS tab settings**.

*Table 5-1 Preferences: NetIQ > Identity Manager > Configuration > eDir-to-eDir SSL/TLS Tab Settings*

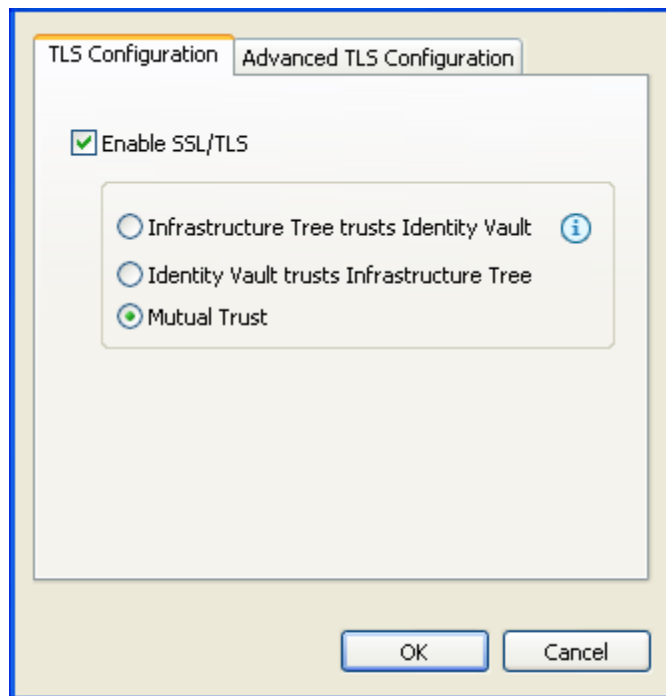
Setting	Description
Preferred key size	Specifies the preferred key size that is generated when drivers are encrypted and stored in eDirectory: 256, 384, 512, 768, 1024, or 2048 bytes.

Setting	Description
<b>Preferred secure hash algorithm</b>	<p>Specifies the preferred hash algorithm to use when encrypting drivers: SHA1-RSA, MD2-RSA, MD5-RSA, SHA256-ECDSA, or SHA384-ECDSA.</p> <ul style="list-style-type: none"> <li>◆ For 256 key size, Designer provides SHA256-ECDSA algorithm.</li> <li>◆ For 384 key size, Designer provides SHA384-ECDSA algorithm.</li> </ul> <p>SHA256-ECDSA and SHA384-ECDSA are Suite B-compliant algorithms.</p>
<b>Preferred validity period</b>	<p>Specifies the validity period for a driver certificate, ranging from 6 months to 10 years.</p>
<b>Always overwrite existing certificates</b>	<p>Specifies that existing driver certificates are overwritten with each deployment. If you select this option, Designer deletes existing certificates and creates new ones. The new certificates are then good for another two years (assuming the default value is two years, as defined in the <b>Preferred Validity Period</b> field.) If you select <b>Live &gt; Create eDir-to-eDir Certificates</b>, Designer deletes old certificates and creates new ones.</p>
<b>Overwrite certificates only if they have expired</b>	<p>Specifies that only expired driver certificates are overwritten with each deployment. This is the default setting. The default expiration length is two years. If a certificate expires, SSL/TLS stops working. If a certificate is expired, Designer deletes it and creates a new one.</p>
<b>Never overwrite existing certificates</b>	<p>Never overwrites driver certificates.</p>
<b>Restart drivers after building certificates</b>	<p>Restarts drivers after certificates have been updated or created.</p>

## 7. Enable TLS.

Perform the following actions to enable TLS:

- a. Right-click **eDir-to-eDir** in the Outline view, then click **Secure Connection Settings**.
- b. Right-click an eDir-to-eDir driver, click **Properties > Driver Configuration > Authentication**, then click **Configure TLS**. The **Configure TLS** icon displays only on eDir-to-eDir driver pages.
- c. Click **Enable SSL/TLS**.



- d. Select a direction of trust.

These options apply to certificates that NetIQ creates for eDirectory. The options do not apply to third-party security certificates.

The default is Mutual Trust, which is considered to be the most secure.

Unless you want to use the certificate for authentication, the option that you select doesn't matter. If only encryption is important, you can select any one of the three options.

If authentication is important, select the option that gives you the appropriate trust.

**Scenario: JJ Infrastructure Tree Trusts JT ID Vault.** JJ Infrastructure Tree is the organizational certificate authority. JJ Infrastructure Tree signed a certificate and placed it in JT IDVault. JT ID Vault trusts JJ Infrastructure Tree. The two vaults synchronize data through a secure connection.

If the two vaults break their trusted relationship, JJ Infrastructure Tree can prevent sensitive data from being synchronized by revoking its certificate.

**Scenario: JT ID Vault Trusts JJ Infrastructure Tree.** JJ Infrastructure Tree creates two certificates. One is placed in JJ Infrastructure Tree, and the other is placed in JT ID Vault. The two vaults synchronize data through a secure connection.

If the two vaults break their trusted relationship, JJ Infrastructure Tree can prevent sensitive data from being synchronized by revoking its certificate.

**Scenario: Mutual Trust.** JT ID Vault and JJ Infrastructure Tree both sign certificates.

- e. (Optional) Use the **Advanced TLS Configuration** to select key size, hash algorithm, and validity period.

The validity period is important for when a certificate has expired and you need to overwrite or create a new one.

8. Click **OK**.

You can enable or configure TLS without immediately deploying the drivers. You can turn the settings on. However, you can't create SSL/TLS certificates unless the drivers have been deployed into their respective Identity Vaults. If you enable SSL/TLS but want to create certificates later, you can do so. When you later deploy the eDir-to-eDir drivers, Designer guides you through steps to automatically create certificates.

9. Perform the following actions to create certificate.

The first time you enable and configure SSL/TLS on driver's **Authentication** tab, click **OK**, then follow the prompts. A Create Certificates dialog box appears. Click **Yes**.

You can also create certificate by right-clicking the eDir2eDir application.

Click **Live > Create eDir-to-eDir Certificates**.

For more information, see [Configuration](#) in Designer Administrator guide.

---

**NOTE:** If the settings are not configured, the certificate is created using the default settings.

---

To view the details of the certificate in iManager, navigate to **iManager > NetIQ > Certificate access**. Click the name of the certificate to view the summary of the certificate. Click **Validate** on top of the screen to view if the certificate is valid or if it has expired.

## Establishing Secure Connections Using KMO

To configure your Identity Vault system to handle secure Identity Manager data transfers:

- 1 Find out the tree name or IP address of the destination server.
- 2 Launch iManager and authenticate to your first tree.
- 3 Click the Identity Manager Administration tab.
- 4 In the **Administration** list, click **eDir-to-eDir Driver Certificates** to launch the wizard.
- 5 At the Welcome page, enter the requested information for the first tree.

Default values are provided by using objects in the tree that you authenticated to when you launched iManager. You must enter or confirm the following information:

**Driver DN:** Specify the distinguished name of the eDirectory driver (for example, eDirectoryDriver.DriverSet1.Services.Novell).

**Tree:** Verify the name of the current tree; if it is not correct, enter the correct name.

**Username:** Specify the username for an account with Admin privileges in the current tree (for example, Admin).

**Password:** Specify the password for the user.

**Context:** Specify the user's context (for example Services.Novell).

**Enable Suite B:** Select this option to enable the driver communication to use Suite B. The driver uses TLS 1.2 as a communication protocol and encrypts the data using the cryptographic algorithms mandated by Suite B. For more information about Suite B, see [Suite B Cryptography](#).

Select one of the following certificates for securing your communication:

- ◆ **ECDSA 256:** Enables Suite B mode operation by using 128-bit level of security (Suite B Cipher 128-bit only).
- ◆ **ECDSA 384:** Enables Suite B mode operation by using 192-bit level of security (Suite B Cipher 192-bit).

- 6 Click **Next**.

The wizard uses the information you entered to authenticate to the first tree, verify the driver DN, and verify that the driver is associated with a server.

**7** Specify the requested information for the second tree:

**Driver DN:** Specify the distinguished name of the eDirectory driver (for example, eDirectoryDriver.DriverSet2.Novell).

**Tree:** Specify the name of the second tree.

Alternatively, specify the IP address of the server in the second tree running the eDirectory driver. This server must hold partition replicas of the Username (specified in Step 5), the driver set, and the server object.

**Username:** Specify the username for an account with Admin privileges in the second tree (for example, Admin).

**Password:** Specify the password for the user.

**Context:** Specify the user's context (for example Users.Novell).

**8** Click **Next**.

The wizard uses the information you entered to authenticate to the second tree, verify the driver DN, and verify that the driver is associated with a server.

**9** Review the information on the Summary Page, then click **Finish**.

If KMOs already existed for these trees, the wizard deletes them and then does the following:

- ◆ Exports the trusted root of the CA in the first tree.
- ◆ Creates KMO objects.
- ◆ Issues a certificate signing request.
- ◆ Places certificate key pair names in the drivers' Authentication IDs (see ["Authentication" on page 46](#)).

## Creating the Certificate Manually

The eDirectory to eDirectory certificate wizard rarely fails to create the certificate properly. In such scenarios, perform the following actions to manually create the certificate and establish a secure communication between the two eDirectory drivers.

The following example explains how to manually create the certificate.

### Prerequisites

- ◆ Two Linux servers

---

**NOTE:** 1 server is a SLES 12.2 which is the main IDV (vault) tree and the other is a Red Hat 7.2 server.

---

- ◆ Both servers have eDirectory installed
- ◆ Both servers have Identity Manager 4.6 installed
- ◆ Both servers have eDirectory driver installed

---

**NOTE:** Since both eDirectory trees have certificate authority, either of the trees can be used as the source tree. The main tree is used as the vault tree which has iManager certificates as well. This ensures all certificates are on the same server (for convenience).

---

Perform the following actions to create the certificate using KMO method:

- 1 Create a normal server certificate by navigating to **Roles and Tasks > NetIQ certificates server > Create server certificate**.
- 2 Browse for the **Server**. Provide a **Nickname**. Choose the **Creation method** as **Custom** as we would manually make some changes. Click **Next**.

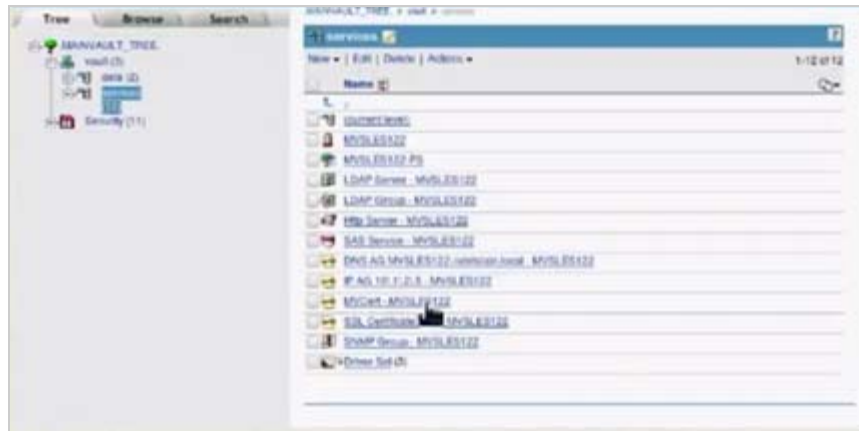


- 3 Select **Organizational certificate authority** (default) and click **Next**.
- 4 Select **SSL or TLS** as **Key type**. Select **Server** under **Extended key type**. Click **Next**.



- 5 Select **SHA 256** as the **Signature algorithm** as it is considered more secure. Change the **validity period** as **Maximum** (as per your choice). Click **Next**.
- 6 Select **Your organization's certificate** (default). Click **Next**.
- 7 The summary of the certificate information is listed. Click **Finish**.

- The KMO object is successfully created and listed under **Objects>server**.



Log into the Red Hat server and repeat the above procedure to create the certificate. However, the following changes need to be made:

- In step 3, select **External certificate authority**.
- After step 7, click **Save Certificate Signing Request** and save it in a preferred location.

---

**NOTE:** This is a .csr file

---

Exit the Red Hat server and log into the main server (SLES).

- Navigate to **Roles and Tasks >Certificate server >Issue certificate**. Browse for the saved .csr file and click **Next**.
- The **Key Type** and **Extended Key Type** remains same as mentioned in step 4. Click **Next**.  
The **Certificate Type** and **Path length** remains with default settings. Click **Next**. **Validity period** is set to **Maximum**. Click **Next**.
- Save the File in **Base 64** format and click **Next**.
- Click **Finish**.



- Save the file to a preferred location. This file is saved in a .b64 format.
- Navigate to **View Objects**, under **Tree**. Navigate to **Security** container and click the **CA**. In the **Certificate** tab, check the **Self Signed Certificate** box and click **Export**.
- Uncheck the **Export private key** checkbox. Select the format as **Base64** and click **Next**.

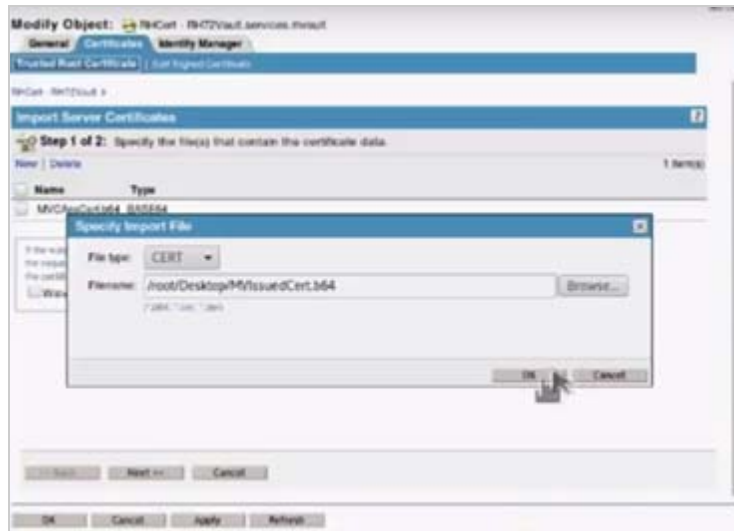


Save the exported certificate to a preferred location. The saved file is in .b64 format.

Exit from the main vault.

Login to the Red Hat server and perform the following actions to synchronize data:

1. Navigate to **View Objects** and browse for the KMO object. Click the **KMO** object. Click the **Certificate** tab > **Trusted Root certificate** tab and click **Import**.
2. Browse and select the self signed certificate. Click **OK**.
3. Click **New** on the top left hand corner and browse for the issued certificate. Click **OK**.



4. Two certificates to import are listed. Click **Next**, **Finish** and **Ok**.  
For KMO method, configure and start the eDirectory driver.
5. Exit from the Red Hat server and login to the main vault server.  
Configure and start the driver on the vault. Both the connections are secure.
6. Navigate to **View objects** and make a change (edit) to an object.  
For example: changing the description of a user.

Exit from the main server and login to the Red Hat server. The changes made to the user can be noticed here. The data is successfully synchronized.

## Establishing Secure Connections Using Keystore

To establish a secure connection between two eDirectory servers using eDirectory driver, you need to import the trusted root certificate into keystore of connected eDirectory server and vice versa.

- 1 Create a server certificate in iManager.
  - 1a In the **Roles and Tasks** view, click **NetIQ Certificate Server** > **Create Server Certificate**.
  - 1b Browse to and select the server object where the eDirectory driver is installed.
  - 1c Specify a certificate nickname. For example `maincert`.

---

**NOTE:** NetIQ recommends that you avoid using spaces in the certificate nickname. For example, use `maincert` instead of `main cert`.

Also, make a note of the certificate nickname. This nickname is used for the KMO name in the driver properties.

---

- 1d Select **Custom** in the certificate creation method, then click **Next**.
- 1e Keep the default **Organizational Certificate Authority** selection, then click **Next**.
- 1f Uncheck **Enable extended key usage** check box.
- 1g Accept the default settings and review the summary, click **Finish**, then click **Close**.
- 2 Export the certificate.
  - 2a In the **Roles and Tasks** view, click **NetIQ Certificate Access > server certificates**.
  - 2b Select the KMO object that you created in [Step 1](#) and click **Validate**. For example, `maincert`.
  - 2c Select the validated KMO object and click **Export**.
  - 2d Select the KMO object from the **Certificates** list.
  - 2e Ensure the **Export private key** check box is checked.
  - 2f Provide a password and click **Next**.  
Use this as a source password while generating a keystore.
  - 2g Save the certificate to a file in `.pfx` format, click **Close**.
- 3 Generate a keystore from the exported certificate using the following command at the command line:

```
keytool -importkeystore -srckeystore <file saved in step 2> -srcstoretype PKCS12 -destkeystore <name of the keystore> -alias <kmo name provided in Step 1>
```

For example:

```
keytool -importkeystore -srckeystore maincert.pfx -srcstoretype PKCS12 -destkeystore new.keystore -alias maincert
```

After executing this command, specify the destination password, then provide the source password which you want to use as a **Keystore password** while configuring the driver.

- 4 Import the trusted root certificate from the connected eDirectory server and save it to a file in `der` format.
  - 4a In iManager, log in to the connected eDirectory server with administrator rights.
  - 4b In the left pane of the **Roles and Tasks** tab, select **NetIQ Certificate Access > Server Certificates**.
  - 4c Select any server KMO object and click **Validate**. For example, `SSL CertificateDNS`.
  - 4d Select the validated KMO object and click **Export**.
  - 4e Select **OU=Organizational CA** certificate from drop down menu for the **Certificate** option.
  - 4f Select **DER as the Export** format, then click **Next**.
  - 4g Save the file to a local file system in `der` format. For example `PublicKeyCert.der`.
- 5 Add the **DER** file to the keystore created in [Step 3](#) by using the following command at the command line:

```
keytool -import -file <Certificate name> -keystore KEYSTOERPATH\new.keystore -storepass <keystorepass>
```

For example,

```
keytool -import -file PATH_OF_DERFile\PublicKeyCert.der -keystore
KEYSTOERPATH\new.keystore -storepass keystorepass
```

In this command, `storepass` value is same as the destination password that you have provided in [Step 3](#).

NetIQ recommends that you use Java 1.6 or higher version `keytool`. This command might not work with versions earlier than Java 1.6.

- 6 When you are asked to trust this certificate, type `YES`, then press **Enter**.
- 7 Copy the `new.keystore` file to any directory on the same file system that has the Identity Vault files.
- 8 In iManager, select **Identity Manager > Identity Manager Overview**.
- 9 Search the eDirectory driver.
- 10 In the **Driver Settings**, select the SSL Type as **Keystore**.
- 11 In the **Keystore Path** parameter, enter the complete path to the `keystore` file.
- 12 Enable the driver's SSL parameters and configure the other SSL parameters as needed.  
For information, see ["Driver Parameters" on page 48](#).
- 13 Repeat this procedure for the eDirectory driver deployed in the connected eDirectory server.

## Configuring Authentication Between Drivers

In addition to providing the mandatory certificates needed to use SSL, you can set up additional security by configuring the Subscriber channel on one eDirectory driver to authenticate to the Publisher channel on the other driver.

Set a driver object password and application password on each driver. Make sure the driver object password of the first driver matches the application password of the second driver, and that the driver object password of the second driver matches the application password of the first driver. For example:

*Table 5-2 Driver Object and Application Passwords*

	<b>Driver Object Password</b>	<b>Application Password</b>
<b>Driver 1</b>	Provo	Cambridge
<b>Driver 2</b>	Cambridge	Provo

For information about setting the passwords, see ["Driver Object Password" on page 46](#) and ["Authentication" on page 46](#).



# 6 Synchronizing Passwords

To use the eDirectory driver to set up password synchronization between the two Identity Vaults, follow the instructions in the [NetIQ Identity Manager Password Management Guide](#).

The following list contains information that is specific to setting up password synchronization with the eDirectory driver. Use it to supplement the information in the [NetIQ Identity Manager Password Management Guide](#).

- ◆ Universal Password is the standard method to synchronize passwords with Identity Manager. The eDirectory driver's policies and filters (in the basic configuration file) are set up to support this method. However, you can use the older method of synchronizing passwords through the NDS password. This method is also known as synchronizing the public key and private key. If you choose to use the NDS password method, make sure you follow the instructions in "Password Synchronization Scenarios" in the [NetIQ Identity Manager Password Management Guide](#).
- ◆ If you decide to enforce password policies in multiple trees, make sure that the Advanced Password Rules in the password policies are compatible in each tree, so that password synchronization can be successful.

If you enforce incompatible password policies in multiple eDirectory trees, and choose to reset a password back to the distribution password if it does not comply (with the option **If password does not comply, enforce Password Policy on the connected system by resetting user's password to the Distribution Password**), you could encounter a loop in which each Identity Vault server tries to change a noncompliant password.

Information about password policies is in "Managing Passwords by Using Password Policies" in the [NetIQ Identity Manager Password Management Guide](#).

- ◆ The Check Password Status task in iManager does not work for a connected system if the Password policy has Universal Password enabled and does not have the setting selected for synchronizing Universal Password with NDS Password.

The Check Password Status task lets you see whether a user's password in Identity Manager is synchronized with the password on connected systems.

If you are using the eDirectory driver and the password policy for a user specifies in the **Configuration Options** tab that the NDS Password should not be updated when the Universal Password is updated, then the Check Password Status task for that user always shows that the password is not synchronized. The password status is shown as not synchronized, even if the Identity Manager Distribution Password and the Universal Password on the connected system are in fact the same.

This is because the Identity Vault check-password functionality is checking the NDS Password at this time, instead of going through NMAS to refer to the Universal Password.

By default, the NDS Password is updated when the Universal Password is updated in the password policy. If you select this option, Check Password Status should be accurate for the connected system.



# 7 Managing the Driver

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

- ♦ Starting, stopping, and restarting 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
- ♦ Synchronizing objects
- ♦ Migrating and resynchronizing data
- ♦ Activating the driver

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 Setup Guide for Linux](#) or [NetIQ Identity Manager Setup Guide for Windows](#).





# 8 Troubleshooting

Refer to the following sections for information about troubleshooting problems you might encounter with the eDirectory driver:

- ♦ [“Troubleshooting Driver Processes” on page 41](#)
- ♦ [“Synchronizing eDirectory Objects in a Linux High Availability Setup” on page 41](#)
- ♦ [“JCEException while Synchronizing a Password” on page 41](#)

## Troubleshooting Driver Processes

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

## Synchronizing eDirectory Objects in a Linux High Availability Setup

To start the user synchronization immediately after a failover in the Linux High Availability cluster, change the eDirectory driver configuration:

- 1 Set the **Receive timeout in minutes** option in Publisher options to a smaller value.
- 2 Delete the port number from the **Authentication Context** and specify two different ports in the Subscriber and Publisher settings.
- 3 In the Subscriber settings, go to the **Advanced options**, select the **Socket local bind** option and specify the IP address in the **Local bind address for the subscriber socket** option.  
This is the IP address where eDirectory is listening. You must specify the IP address if there are multiple IP addresses in the high availability setup.
- 4 Specify the same IP address that you specified in the **Local bind address for the subscriber socket** option in the Publisher settings.

## JCEException while Synchronizing a Password

The driver reports JCEException while synchronizing a password.

It is safe to ignore the exception. It does not impact the functionality of the driver.



# 9 Known Issues

The following known issues exist for this version of the driver:

## Mutual Authentication Feature is not Working

**Issue:** If you enable the Mutual Authentication option on eDirectory driver, it fails to establish a connection between eDirectory servers.

**Workaround:** There is no workaround at this time.



# A Driver Properties

This section provides information about the Driver Configuration and Global Configuration Values properties for the eDirectory driver. These are the only unique properties for the eDirectory driver. 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 for Identity Manager, it is marked with a Designer icon.

- ♦ “[Driver Configuration](#)” on page 45
- ♦ “[Global Configuration Values](#)” on page 50

## Driver Configuration

In iManager:

- 1 Click the Identity Manager Administration tab.
- 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 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.

In Designer:

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

The Driver Configuration options are divided into the following sections:

- ♦ “[Driver Module](#)” on page 45
- ♦ “[Driver Object Password](#)” on page 46
- ♦ “[Authentication](#)” on page 46
- ♦ “[Startup Option](#)” on page 47
- ♦ “[Driver Parameters](#)” on page 48
- ♦ “[ECMAScript](#)” on page 49
- ♦ “[Global Configurations](#)” on page 49

## 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.

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

**Connect to Remote Loader:** The Remote Loader is not used with the eDirectory driver. However, Designer includes two suboptions, one (Driver Object Password) of which is required to set up authentication between two eDirectory drivers. If you use a driver object password, you need to select the **Connect to Remote Loader** option, set the password, click **Apply** to save the password, then select the **Java** option again.

- ♦ **Remote Loader Client Configuration for Documentation:** This option is not used with the eDirectory driver.
- ♦ **Driver Object Password:** Specifies a password for the eDirectory driver. This password must match the “[Application Password:](#)” on page 47 set for the destination eDirectory driver.

## Driver Object Password

The driver object password is used to enable the eDirectory driver’s Subscriber channel to authenticate to the Publisher channel of the destination eDirectory driver. This authentication, although it is optional, provides an extra layer of security between the two drivers.

In Designer, this setting is located under the [Connect to Remote Loader:](#) option.

For additional information about setting up authentication between the two drivers, see [Chapter 5, “Securing Driver Communication,”](#) on page 25.

**Driver Object Password:** Specifies a password for the eDirectory driver. This password must match the “[Application Password:](#)” on page 47 set for the destination eDirectory driver.

## Authentication

The Authentication section stores the information required to authenticate to the connected system. For the eDirectory driver, it stores the information required to authenticate to the connected eDirectory driver and tree.

**Authentication information for server:** Displays or specifies the server that the driver is associated with.

**Authentication ID:** This ID is used by the driver to authenticate to the destination eDirectory driver. The ID is automatically generated and stored in this field when you run the NDS-to-NDS Driver Certificates Wizard. Authentication ID is used for establishing the secure connection. Format of the data in the Authentication ID field is the name of the local KMO object to use for the connection. This KMO object should be present in the local directory tree. For example, if eDirectory driver is configured between Server1 and Server2:

- ♦ Authentication ID on server1 is : eDirectory(Server1)
  - ♦ Authentication ID on Server2 is : eDirectory(Server2)
- Both KMOs eDirectory(Server1) and eDirectory(Server2) should be signed by the same certificate authority (CA).

For information, see [Chapter 5, “Securing Driver Communication,”](#) on page 25.

**Authentication Context:** Specify the hostname or IP address of the destination server as well as the decimal port number (for example, 187.168.1.1:8196).

You can specify a separate port for Subscriber and Publisher channels by specifying a second port number following a second colon. If a second port number is specified, the Publisher channel uses the second port number rather than using the same port number as the Subscriber channel (for example, 255.255.255.255:2000:2001).

If your server has multiple IP addresses, you can specify the IP address you want the Publisher channel to use. This requires specifying the remote IP address, the Subscriber channel port, the local IP address, and the Publisher channel port. For example, 137.65.134.81:2000:137.65.134.83:2000 specifies that the Subscriber channel communicates with the remote tree on 137.65.134.81, port 2000, and that the Publisher channel listens on 137.65.134.83, port 2000.

If you see `java.net.ConnectException: Connection Refused`, no port connection is available in the other eDirectory tree. This error might be caused by one of the following:

- ◆ The driver in the other eDirectory tree is not running.
- ◆ The driver is running but is configured to use a different port.

**Remote Loader Connection Parameters:** The eDirectory driver does not support the use of the Remote Loader. These options do not apply.

**Application Password:** The application password, when used in conjunction with the driver object password, enables the eDirectory driver's Subscriber channel to authenticate to the Publisher channel of the destination eDirectory driver. This authentication, although it is optional, provides an extra layer of security between the two drivers.

This password be the same as the driver object password for the destination eDirectory driver.

For more information, see [Chapter 5, "Securing Driver Communication," on page 25](#).

**Remote Loader Password:** The eDirectory driver does not support the use of the Remote Loader. These options do not apply.

**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.

## Startup Option

The Startup Option section enables 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 applies only if the driver is deployed and was previously disabled. If this is not selected, the driver re-synchronizes the next time it is started.

## 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.

The parameters are divided into the following categories:

- ◆ [“Driver Settings” on page 48](#)
- ◆ [“Subscriber Settings” on page 48](#)
- ◆ [“Publisher Settings” on page 49](#)

## Driver Settings

**SSL type:** Specifies whether to use a Key Material Object (KMO) for SSL or use a Java keystore file to secure the eDirectory driver communication. If you select **keystore**, provide the following mandatory parameters:

- ◆ **Name of the keystore file:** Specify the name of the Java keystore file. If the file path is not specified, the file must be available in the eDirectory DIB file directory.
- ◆ **Keystore password:** Specify the password to access the Java keystore file that contains the SSL certificates.
- ◆ **Reenter Keystore password:** Specify the password again.
- ◆ **Remove existing password:** Enable this option if you do not want to specify the keystore password. If you select this option, the **Keystore password** option is automatically disabled.
- ◆ **Name of certificate (key alias):** Specify the name of the key and certificate used when creating the keystore. The Java keytool program refers to this parameter as the alias.
- ◆ **Certificate password (key password):** Specify the password for the key created in the keystore.
- ◆ **Reenter Certificate password (key password):** Specify the key password again.
- ◆ **Remove existing password:** Enable this option if you do not want to specify the key password. If you select this option, the **Certificate password** option is automatically disabled.
- ◆ **Advanced options:** Select **Show** to display the advanced options.
- ◆ **Subscriber acts as server for SSL handshake:** Ideally, the SSL handshake protocol has the subscriber acting as the client side of the SSL handshake. Select **Yes** to reverse the protocol and set the subscriber as the server side of the SSL handshake.
- ◆ **Disable mutual authentication - only used if acting as server:** Select **Yes** to disable the SSL mutual authentication. This option is applicable only if you set subscriber as the server side of the SSL handshake.

**Secure Protocol:** Specifies the version of the TLS protocol that is used to establish a connection between eDirectory drivers. Identity Manager supports TLSv1, TLSv1\_1, and TLSv1\_2.

## Subscriber Settings

**Address or host name of remote publisher:** Specifies the IP address or DNS name of the server hosting the remote eDirectory driver that the local subscriber connects to.

**TCP port of remote publisher:** If the remote publisher options specify a TCP port, this must be set to **specify** and the value from the remote Publisher channel entered into the **Port number** field. (These two fields must match what is set in the remote Publisher channel's options, which have corresponding fields).



**Port number:** Specifies the port number that the remote publisher is configured to run on. Displays only if you select **specify** in the **TCP port of remote publisher** field.

**Advanced options:** Displays additional fields when you select **show**.

**Socket local bind:** The **local bind** fields specify which IP address the Subscriber channel's socket will be bound to. This is generally only useful if the server has more than one IP address and it is important to bind to a particular address because of firewall settings.

**Local bind address for subscriber socket:** The **local bind** fields specify which IP address the Subscriber channel's socket will be bound to. This is generally only useful if the server has more than one IP address and it is important to bind to a particular address because of firewall settings.

**Receive timeout in minutes:** In order to detect a lost TCP/IP connection, the eDir-to-eDir driver periodically sends small packets. This value determines how long after entering a receive-wait condition the Subscriber channel waits until sending a keep-alive packet to determine if the TCP/IP connection has been lost. Generally, do not change this value except under instruction from NetIQ.

The default value for the Subscriber channel is one minute.

## Publisher Settings

**Publisher heartbeat interval:** Specifies how often you want the driver to send a status message along the Publisher channel when there has not been any traffic during the interval time.

**Local bind address for publisher socket:** Specifies which IP address the Subscriber channel's socket will be bound to. This is generally only useful if the server has more than one IP address and it is important to bind to a particular address because of firewall settings. This setting applies to the local publisher's "server" socket on which the local publisher listens for connections from the remote Subscriber channel.

**Receive timeout in minutes:** In order to detect a lost TCP/IP connection, the eDirectory driver periodically sends small packets. This value determines how long after entering a receive-wait condition the Publisher channel waits until sending a keep-alive packet to determine if the TCP/IP connection has been lost. Generally, do not change this value except under instruction from NetIQ.

The default value for the Publisher channel is ten minutes.

## ECMAScript

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


## Global Configurations

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.

# 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 eDirectory driver includes several GCVs that are created from information supplied during importing the driver configuration file (see [Chapter 3, “Creating a New Driver Object,” on page 15](#)) and one that is not.

The driver also includes the GCVs that are used with password synchronization. In Designer, you must click the  icon next to a password synchronization GCV to edit it. This displays the Password Synchronization Options dialog box that has a better view of the relationship between the different settings. In iManager, navigate to **Driver Properties > Global Configuration Values** and edit the password synchronization settings in your password synchronization policy tab.

You can add your own GCVs if you discover you need additional ones as you implement policies in the driver.


To access the driver's GCVs in iManager:

- 1 Click the Identity Manager Administration tab.
- 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 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 access the driver's GCVs in Designer:

- 1 Open a project in the Modeler.
- 2 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**.

The Global Configuration Values are divided into categories:

- ♦ [“Default Configuration” on page 50](#)
- ♦ [“Entitlements” on page 51](#)
- ♦ [“Password Synchronization” on page 53](#)
- ♦ [“Account Tracking” on page 54](#)
- ♦ [“Managed System Information” on page 54](#)

## Default Configuration

The following GCVs define control the default configuration of the eDirectory driver:

**eDirectory Publisher Placement type:** Controls how the objects are placed in the remote Identity Vault and the local Identity Vault. The options are:

- ◆ **Mirrored:** Mirrors the structure between the remote Identity Vault and the local Identity Vault.

If you choose this option, use the same option for configuring both eDirectory trees you are synchronizing.

This option in the driver configuration synchronizes User, Group, Organization, Country, and Organizational Unit objects. It also mirrors the structure of a subtree in the other tree.

- ◆ **Flat:** All of the objects are placed into a single container.

This option synchronizes User and Group objects and places all users in one container and all groups in another container.

This option is typically used in conjunction with the Department option (or a similar configuration) in the other tree.

This option doesn't create the containers that hold the users and groups. You must create those manually.

- ◆ **Department:** Users are placed in containers named after the department.

This option synchronizes User and Group objects and places all users and groups in a container based on the **Department** field in your management console.

This configuration is typically used in conjunction with the Flat option (or a similar configuration) in the other tree.

This option doesn't create the containers for each department. You must create those manually. They must be the same as the container specified during import.

**Remote Tree Base User Container:** Specify the source container of the user objects in the remote Identity Vault.

**Remote Tree Base Groups Container:** Specify the source container of the group objects in the remote Identity Vault.

## Entitlements

There are multiple sections in the **Entitlements** tab. Depending on which packages you installed, different options are enabled or displayed.

- ◆ [“Entitlements” on page 51](#)
- ◆ [“Data Collection” on page 52](#)
- ◆ [“Role Mapping” on page 52](#)
- ◆ [“Resource Mapping” on page 52](#)
- ◆ [“Parameter Format” on page 53](#)
- ◆ [“Entitlement Extensions” on page 53](#)

## Entitlements

For more information about entitlements, see [“Entitlements” on page 11](#).

**Use Entitlements to control eDirectory Accounts:** Select **True** to enable the driver to manage user accounts based on the driver's defined entitlements. Select **False** to disable management of user accounts based on the entitlements.

**Enable Login Disabled attribute sync:** Select **True** if the changes made to the loginDisabled attribute in the Identity Vault should be synced even if the User Account entitlement (Account) is enabled.

**Account action on Entitlement Revoke:** Select the action to take when a user account entitlement is revoked. The options are **Disable User, Do Nothing, or Delete User**. By default, **Disable User** is selected.

**Use Group Entitlement:** Select **True** to enable the driver to manage user groups based on the driver's defined entitlements.

Select **False** to disable management of group membership based on the entitlements.

**Advanced Settings:** Select show to display the entitlement options that allow or deny additional functionality like data collection and others. These settings should rarely be changed.

---

**NOTE:** The eDirectory driver is installed and configured in two trees. You should only install the entitlement package in one of the trees.

---

## Data Collection

Data collection enables the Identity Report Module to gather information to generate reports. For more information, see the [Administrator Guide to NetIQ Identity Reporting](#).

**Enable data collection:** Select **Yes** to enable data collection for the driver through the Data Collection Service by the Managed System Gateway driver. If you are not going to run reports on data collected by this driver, select **No**.

**Allow data collection from user accounts:** Select **Yes** to allow data collection by the Data Collection Service through the Managed System Gateway driver for the user accounts.

**Allow data collection from groups:** Select **Yes** to allow data collection by the Data Collection Service through the Managed System Gateway driver for groups.

## Role Mapping

The Role Mapping Administrator allows you to map business roles with IT roles.

**Enable role mapping:** Select **Yes** to make this driver visible to the Role Mapping Administrator.

**Allow mapping of user accounts:** Select **Yes** if you want to allow mapping of user accounts in the Role Mapping Administrator. An account is required before a role, profile, or license can be granted through the Role Mapping Administrator.

**Allow mapping of groups:** Select **Yes** if you want to allow mapping of groups in the Role Mapping Administrator.

## Resource Mapping

The Roles Based Provisioning Module allows you to map resources to users. For more information, see the [NetIQ Identity Manager - User's Guide to the Identity Applications](#).

**Enables resource mapping:** Select **Yes** to make this driver visible to the Roles Based Provisioning Module.

**Allow mapping of user accounts:** Select Yes if you want to allow mapping of user accounts in the Roles Based Provisioning Module. An account is required before a role, profile, or license can be granted.

**Allow mapping of groups:** Select **Yes** if you want to allow mapping of groups in the Roles Based Provisioning Module.

## Parameter Format

**Format for Account entitlement:** Select the parameter format the entitlement agent must use. The options are **Identity Manager 4** or **Legacy**.

**Format for Group entitlement:** Select the parameter format the entitlement agent must use. The options are **Identity Manager 4** or **Legacy**.

## Entitlement Extensions

**User account extensions:** The content of this field is added below the entitlement elements in the EntitlementConfiguration resource object.

**Group extensions:** The content of this field is added below the entitlement element in the EntitlementConfiguration resource object.

**Exchange mailbox extensions:** The content of this field is added below the entitlement element in the EntitlementConfiguration resource object.

## Password Synchronization

The following GCVs control password synchronization for the eDirectory driver. For more information, see the [NetIQ Identity Manager Password Management Guide](#).

In Designer, you must click the  icon next to a GCV to edit it. This displays the Password Synchronization Options dialog box for a better view of the relationship between the different GCVs.

In iManager, to edit the Password management options go to **Driver Properties > Global Configuration Values**, and then edit it in your Password synchronization policy tab.

**Connected System Name or Driver Name:** Specify the name of the driver. The e-mail notification template uses this value to identify the source of the notification message.

**Application accepts passwords from Identity Manager:** If **True**, allows passwords to flow from the Identity Manager data store to the connected system.

**Identity Manager accepts passwords from application:** If **True**, allows passwords to flow from the connected system to Identity Manager.

**Publish passwords to NDS password:** Use the password from the connected system to set the non-reversible NDS password in eDirectory.

**Publish passwords to Distribution Password:** Use the password from the connected system to set the NMAS Distribution Password used for Identity Manager password synchronization.

**Require password policy validation before publishing passwords:** If **True**, applies NMAS password policies during publish password operations. The password is not written to the data store if it does not comply.

**Reset user's external system password to the Identity Manager password on failure:** If True, on a publish Distribution Password failure, attempt to reset the password in the connected system by using the Distribution Password from the Identity Manager data store.

**Notify the user of password synchronization failure via e-mail:** If True, notify the user by e-mail of any password synchronization failures.

## Account Tracking

Account tracking is part of the Identity Reporting Module. For more information, see the [Administrator Guide to NetIQ Identity Reporting](#).

**Enable account tracking:** If this option is set to **True**, it enables account tracking policies. Set it to **False** if you do not want to execute account tracking policies.

**Realm:** Specifies the name of the realm, security domain, or namespace in which the account name is unique.

**Object Class:** Specifies the object class to track. Class names must be in the application namespace.

**Identifiers:** Specifies the account identifier attributes. Attribute names must be in the application namespace.

**Status attribute:** Specifies the name of the attribute in the application namespace to represent the account status.

**Status active value:** Specifies the value of the status attribute that represents an active state.

**Status inactive value:** Specifies the value of the status attribute that represents an inactive state.

**Subscription default status:** Specifies the default status the policies assume when an object is subscribed to the application and the status attribute is not set in the Identity Vault.

**Publication default status:** Specifies the default status the policies assume when an object is published to the Identity Vault and the status attribute is not set in the application.

## Managed System Information

These settings help the Identity Reporting Module function to generate reports. There are different sections in the **Managed System Information** tab.

- ◆ [“General Information” on page 54](#)
- ◆ [“System Ownership” on page 55](#)
- ◆ [“System Classification” on page 55](#)
- ◆ [“Connection and Miscellaneous Information” on page 55](#)

## General Information

**Name:** Specifies a descriptive name for this Identity Vault. This name is displayed in the reports.

**Description:** Specifies a brief description of this Identity Vault. This description is displayed in the reports.

**Location:** Specifies the physical location of this Identity Vault. This location is displayed in the reports.

**Vendor:** Specifies NetIQ as the vendor of the Identity Vault. This information is displayed in the reports.

**Version:** Specifies the version of this Identity Vault. This version information is displayed in the reports.

## System Ownership

**Business Owner:** Browse to and select the business owner in the Identity Vault for this Identity Vault. You must select a user object, not a role, group, or container.

**Application Owner:** Browse to and select the application owner in the Identity Vault for this Identity Vault. You must select a user object, not a role, group, or container.

## System Classification

**Classification:** Specifies the classification of the Identity Vault. This information is displayed in the reports. The options are:

- ◆ Mission-Critical
- ◆ Vital
- ◆ Not-Critical
- ◆ Other

If you select **Other**, you must specify a custom classification for the Identity Vault.

**Environment:** Specifies the type of environment the Identity Vault provides. The options are:

- ◆ Development
- ◆ Test
- ◆ Staging
- ◆ Production
- ◆ Other

If you select **Other**, you must specify a custom classification for the Identity Vault.

## Connection and Miscellaneous Information

**Connection and miscellaneous information:** This options is always set to **hide**, so that you don't make changes to these options. These options are system options that are necessary for reporting to work. If you make any changes, reporting stops working.





# B Synchronized Attributes

The filter for the basic driver configuration synchronizes the following attributes:

*Table B-1 eDirectory Driver Attributes That Are Synchronized*

accessCardNumber	Initials	preferredDeliveryMethod
ACL	instantMessagingID	preferredName
assistant	internationaliSDNNumber	Private Key
assistantPhone	Internet EMail Address	Public Key
businessCategory	jackNumber	registeredAddress
city	jobCode	roomNumber
CN	L	S
co	Language	SA
company	Mailbox ID	Security Equals
costCenter	Mailbox Location	Security Flags
costCenterDescription	mailstop	See Also
departmentNumber	manager	siteLocation
Description	managerWorkforceID	Surname
destinationIndicator	mobile	Telephone Number
directReports	NSCP:employeeNumber	teletexTerminalIdentifier
EMail Address	otherPhoneNumber	telexNumber
employeeStatus	O	Timezone
employeeType	OU	Title
Equivalent To Me	pager	tollFreePhoneNumber
Facsimile Telephone Number	personalTitle	UID
Full Name	photo	uniqueID
Generational Qualifier	Physical Delivery Office Name	vehicleInformation
Given Name	Postal Address	workforceID
Group Membership	Postal Code	x121Address
GUID	Postal Office Box	x500UniqueIdentifier
Higher Privileges		



# C Trace Levels

The driver supports the following trace levels:

*Table C-1 Supported Trace Levels*

Level	Description
0	No trace messages are displayed or logged
1-2	Basic trace messages like driver start/stop and documents sent/received are displayed and logged
3-12	Trace Level 1-2 messages are displayed and logged, as well as some additional informative messages
13	Trace Level 1-12 messages are displayed and logged, as well as in-depth details of messages received through the Publisher channel

