
NetIQ® Identity Manager

Driver for JDBC Fanout Implementation

Guide

October 2019

Legal Notice

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

Copyright (C) 2019 NetIQ Corporation. All rights reserved.

Contents

About NetIQ Corporation	7
About This Guide	9
About this Book and the Library	11
1 Overview	13
Components for Fanout Configuration	13
Data Transfer Between Systems	15
Supported Operations	15
Password Synchronization	15
Data Synchronization	15
Installing and Configuring the Driver	16
Supported Databases	16
Supported Third-Party JDBC Drivers	16
Driver Concepts	16
Database Concepts	18
2 Installing the Fanout Driver Components	19
Prerequisites for Installing the Driver	19
Installing the Fanout Agent with Identity Manager 4.8	19
Linux	20
Windows	20
Migrating the Fanout Agent Configuration	21
Installing ActiveMQ	21
Starting ActiveMQ	22
Working with ActiveMQ	22
Setting Up ActiveMQ Startup Service	22
Enabling the ActiveMQ Service on Linux	23
Enabling the ActiveMQ Service on Windows	23
Upgrading ActiveMQ	23
3 Configuring the Fanout Agent	25
Generating the Default Configuration File	25
Starting the Fanout Agent	28
Managing the Fanout Agent	29
Enabling Auditing for the Fanout Agent	32
Stopping the Fanout Agent	32
4 Creating New Fanout Driver	33
Creating a Fanout Driver Object in Designer	33
Importing the Driver Packages	33
Installing the Driver Packages	34
Configuring the Driver Object	36
Configuring the Database Connections for the Driver	37
Deploying the Driver Object	39
Starting the Driver	39

Managing the Connection Objects	40
Activating the Driver	40
Adding Packages to an Existing Driver	40
5 Upgrading an Existing Driver	43
What's New	43
What's New in Version 1.1.0.0	43
Upgrading to Version 1.1.0.0	43
.....Upgrading the	
Installed	
Packages	43
Updating the Fanout Agent and Driver	44
6 Securing Fanout Driver Communication	47
Creating a Keystore and a Truststore	47
Enabling SSL for ActiveMQ	48
Enabling SSL for the Fanout Driver Shim	48
Enabling SSL for the Fanout Agent	49
7 Troubleshooting the Driver	51
Mismatch in Driver Version	51
Fanout Driver and the Fanout Agent Time Out When ActiveMQ and Fanout Agent are Running	51
Issue with Broadcast Queries	51
ActiveMQ May Display Exception Error	51
Changing ActiveMQ Log Levels	52
Cannot Run Two Instances of the Fanout Agent Using the Same Queue Names	52
Unable to Update the Parameters Using the setConfig Command	52
Stopping the Fanout Agent Fails When Any Instance Does Not Respond	52
Cleaning a State File for a Connected System	52
Adding a Group To a Specific Instance	52
Driver Does Not Start	54
Driver Instance Does Not Start	54
Fanout Agent Stop Processing Events	54
No Connection from the Fanout Agent to ActiveMQ	54
Fanout Agent Does Not Send Events	55
Troubleshooting Driver Processes	55
Manually Deleting the Unused ActiveMQ Queues	55
Collecting Trace Files For Troubleshooting	55
8 Managing the Driver	57
A Known Issues and Limitations	59
Known Issues	59
Limitations	60
B Driver Properties	63
Driver Configuration	63
Driver Module	63
Authentication	64

Startup Option	64
Driver Parameters	64
ECMA Script	66
Global Configuration	66
Global Configuration Values	67
Global Configuration Values	67
Managed System Information	68
Entitlements	69
Account Tracking	71
Password Synchronization	71
JDBC Fanout Common	72
C Setting Up Trace Levels	73
D REST Endpoints	75

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.

Worldwide:	www.netiq.com/about_netiq/officelocations.asp
United States and Canada:	1-888-323-6768
Email:	info@netiq.com
Web Site:	www.netiq.com

Contacting Technical Support

For specific product issues, contact our Technical Support team.

Worldwide:	www.netiq.com/support/contactinfo.asp
North and South America:	1-713-418-5555
Europe, Middle East, and Africa:	+353 (0) 91-782 677
Email:	support@netiq.com
Web Site:	www.netiq.com/support

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 **comment on this topic** at the bottom of any page in the HTML version of the documentation posted at www.netiq.com/documentation. You can also email 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 <http://community.netiq.com>.

About This Guide

This guide explains how to install and configure the Identity Manager JDBC Fanout Driver.

Audience

This guide is intended for administrators implementing Identity Manager, application server developers, Web services administrators, and consultants.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation, or go to www.novell.com/documentation/feedback.html and enter your comments there.

Documentation Updates

For more information about the library for Identity Manager, see the following resources:

- ♦ [Identity Manager documentation website \(https://www.netiq.com/documentation/identity-manager-48/\)](https://www.netiq.com/documentation/identity-manager-48/)
- ♦ [Identity Manager drivers documentation website \(https://www.netiq.com/documentation/identity-manager-48-drivers/\)](https://www.netiq.com/documentation/identity-manager-48-drivers/)

About this Book and the Library

The *Identity Manager Java Database Connectivity (JDBC) Fanout Driver Implementation Guide* provides a generic solution for synchronizing data between an Identity Vault and multiple databases. This guide provides an overview of the driver's technology as well as configuration instructions.

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

For more information about the library for Identity Manager, see the following resources:

- ♦ [Identity Manager documentation website \(https://www.netiq.com/documentation/identity-manager-48/\)](https://www.netiq.com/documentation/identity-manager-48/)
- ♦ [Identity Manager drivers documentation website \(https://www.netiq.com/documentation/identity-manager-48-drivers/\)](https://www.netiq.com/documentation/identity-manager-48-drivers/)

1 Overview

The Identity Manager Java DataBase Connectivity (JDBC) Fanout driver supports the fanout capability at the driver level. The Fanout driver provisions users, groups, and password to multiple databases with minimal effort. This eliminates the need for the Identity Manager administrator to configure multiple JDBC drivers using the same policies to provision multiple databases of the same type. You can centrally manage user accounts and have them automatically created, configured, maintained, and removed when appropriate. This saves cost and time associated with managing the Identity Manager environment. In this configuration, the synchronization is unidirectional, from the Identity Vault to the connected database.

The Fanout driver supports the following features:

- ◆ Synchronizes users and groups from the Identity Vault to the target databases
- ◆ Synchronizes passwords from the Identity Vault to the target databases
- ◆ Provisions or deprovisions user accounts in the target databases based on entitlements
- ◆ Assigns or revokes user permissions in the target databases based on entitlements

IMPORTANT

- ◆ The Fanout driver is a Subscriber channel only driver.
 - ◆ The Remote Loader options do not apply to the Fanout driver. This driver uses the Fanout agent component to create multiple JDBC Fanout driver instances.
-

Components for Fanout Configuration

The Fanout driver relies on the following independent components. [Figure 1-1](#) shows how these components work together.

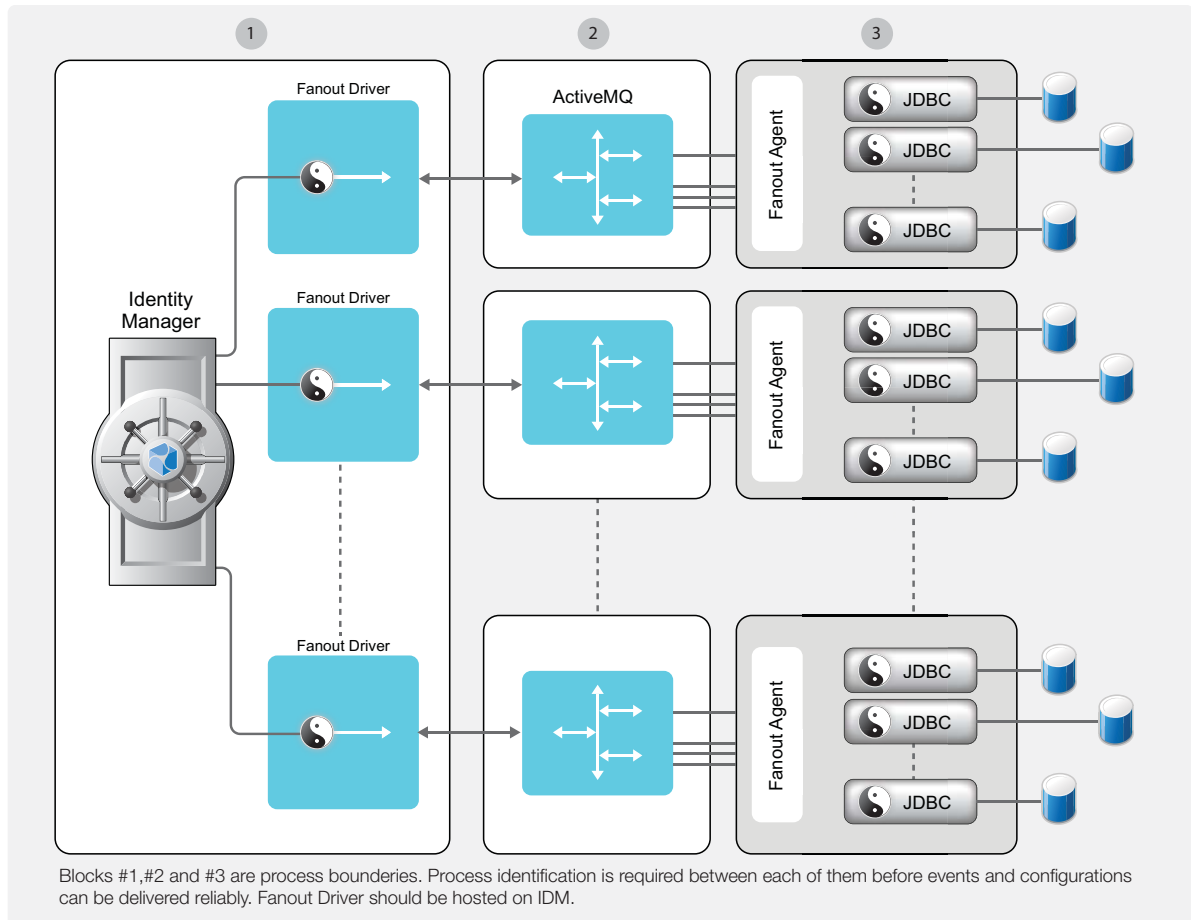
- ◆ **Fanout Driver Shim:** The Fanout driver shim is a Java-based interface driver. The driver shim virtually connects to the Fanout agent through ActiveMQ.
- ◆ **ActiveMQ:** The Fanout driver and the Fanout agent use ActiveMQ for transferring the Subscriber events, configuration data, and the queries. The Fanout agent creates a separate queue for each JDBC driver instance. The JDBC driver instances wait for the events in their respective queues. For information about installing ActiveMQ, see [“Working with ActiveMQ” on page 22](#). To manage the ActiveMQ queues, launch the URL: `http://<ActiveMQ IP Address>:8161/admin/queues.jsp`.
- ◆ **Fanout Agent:** The Fanout agent is a standalone Java process that works independently of the Identity Vault. The Fanout agent loads the JDBC driver instances based on the configuration of the connection objects in the Fanout driver. For more information about the connection objects, see [“Driver Concepts” on page 16](#).

The Fanout agent provides an interface through REST endpoints for performing basic monitoring and management tasks. To ease the initial deployment, the Fanout agent auto-initializes with the default configuration values when it is started for the first time. To change the default settings, use the REST endpoints. For more information about REST endpoints supported for the driver,

see [Appendix D, “REST Endpoints,” on page 75](#). The driver installation folder contains sample scripts that you can run to manage the Fanout agent. The sample scripts internally use the REST endpoints. For more information, see [“Managing the Fanout Agent” on page 29](#).

Alternatively, you can manually edit the configuration file. This method requires you to restart the Fanout agent for the changes to take effect. However, this is not required for all the changes made through the REST endpoints. To understand which changes need a restart of the Fanout agent, see [“Generating the Default Configuration File” on page 25](#).

Figure 1-1 Identity Manager JDBC Fanout Driver Architecture



The Fanout process works as follows:

- 1 The Identity Vault stores the connection object configuration. The configuration includes connection, authentication, and trace information.
- 2 The Fanout driver receives the initialization document from the engine.
- 3 The Fanout driver and the Fanout agent perform a handshake to establish a connection. This signals the start of the communication between the agent and the driver. The handshake is done through challenge-response sets.
- 4 The Fanout driver queries the Identity Vault for the connection objects associated with this driver and creates multiple initialization documents based on the content in the connection objects.
- 5 The Fanout driver sends the initialization documents to the Fanout agent through ActiveMQ.
- 6 The Fanout agent loads the JDBC driver instances.
- 7 The Fanout driver sends the events to the Fanout agent through ActiveMQ.

- 8 The Fanout agent determines which JDBC driver instances this event should be sent to.
- 9 The JDBC driver processes the event and sends the status of the event to the Fanout agent, which in turn sends this information to the Fanout driver through ActiveMQ.
- 10 The Fanout driver sends this status to the Identity Manager engine.

NOTE: Only one flavour of database can be managed per agent. For example: only Oracle databases or only MS SQL databases on a single agent.

Data Transfer Between Systems

The data is transferred between the Identity Vault and the Fanout driver only on the Subscriber channel.

The Subscriber channel performs the following activities:

- ♦ Watches for changes to the Identity Vault objects.
- ♦ Makes changes to the target databases to reflect those changes.

Supported Operations

The Fanout driver supports the following operations on the Subscriber channel:

- ♦ [“Password Synchronization” on page 15](#)
- ♦ [“Data Synchronization” on page 15](#)

Password Synchronization

The Fanout driver supports password set and check operations on the Subscriber channel. The driver does not support bidirectional password synchronization.

Data Synchronization

The Fanout driver supports direct and indirect data synchronization models.

Model	Association	Description
Direct	Usually associated with views	Views provide the abstraction mechanism that facilitates integration with existing customer tables.
Indirect	Usually associated with tables	Customer tables may not match the structure required by the driver. Therefore, you should create intermediate staging tables that match the structure that the driver requires.

For more information about data synchronization models see [Supported Data Synchronization Models](#) in the *NetIQ Identity Manager Driver for JDBC Implementation Guide*.

Installing and Configuring the Driver

For information about installing and configuring the Fanout driver, see [Chapter 2, “Installing the Fanout Driver Components,”](#) on page 19 and [Chapter 3, “Configuring the Fanout Agent,”](#) on page 25.

Supported Databases

The driver supports the following databases:

Table 1-1 Supported Databases

Database	Version
Microsoft SQL Server	2008, 2008 R2, 2012, 2014 and 2016
MySQL	5.5.x or later 5.6.x or later
Oracle	11g Release 1 (11.1) or later, 12c
Sybase Adaptive Server Enterprise (ASE)	15.0 or later

Identity Manager supports JDBC driver with other types and versions of databases as long as they meet the minimum requirements.

- ◆ Support the SQL-92 entry level grammar.
- ◆ Be JDBC//ODBC accessible.

The functionality may be limited and require contacting consultants for implementing the proper content.

Supported Third-Party JDBC Drivers

The Fanout driver supports the following third-party JDBC drivers:

- ◆ MySQL Connector/J
- ◆ Oracle Thin Client
- ◆ Oracle OCI
- ◆ jTDS

For more information, see [Third-Party JDBC Drivers](#) in the *NetIQ Identity Manager Driver for JDBC Implementation Guide*.

Driver Concepts

- ◆ **Connection Objects:** Connection objects are instances of the DirXML-Resource class. Each connection object holds authentication information of the server or database you are connecting to and the trace information for the instances that the Fanout agent configures and loads. For more information, see [DirXML-Data](#).

The Fanout driver uses the following attributes of the connection object to determine the status of the connection object:

- ♦ **DirXML-ContentType:** The driver uses this attribute to determine if it is a connection object or any other driver resource. To qualify for a connection object, this attribute should have one of the following values:
 - ♦ **application/vnd.novell.dirxml.fanout+xml:** This MIME value signals that the current connection object is disabled.
 - ♦ **application/vnd.novell.dirxml.fanout-enabled+xml:** This MIME value signals that the current connection object is enabled.

Enabled connection objects are loaded by the Fanout agent.

- ♦ **DirXML-Data:** This attribute contains the connection and trace information for the JDBC driver instance. The following is a sample of the XML document that DirXML-Data contains:

```
<?xml version="1.0" encoding="UTF-8"?>
<connection>
  <authentication-info>
    <server>jdbc:jtds:sqlserver://111.111.1.1:1433/idm</server>
    <user>idm</user>
  </authentication-info>
  <trace-info>
    <driverTraceFile>/home/sqlserver2.log</driverTraceFile>
    <driverTraceLevel>5</driverTraceLevel>
    <driverTraceFileSize />
  </trace-info>
  <connection-password-ref display-name="Connection Password"
name="com.fanout.conn.passwd">
    <value>fanout.connection_2.passwd</value>
  </connection-password-ref>
</connection>
```

The JDBC Fanout driver supports three trace levels. For information about what each trace level contains, see [“Setting Up Trace Levels” on page 73](#). For more information about the trace levels supported by generic JDBC driver, see [Trace Levels](#) in the *NetIQ Identity Manager Driver for JDBC Implementation Guide*.

To change the connection or trace information for a JDBC Fanout driver instance in iManager.

1. In iManager, click **View Objects**.
2. From the tree view, browse to and locate the driver set containing the driver.
3. Click the driver, then click the Fanout instance.
4. In the window that opens, click the **Edit Resource** tab.
5. Change the connection or trace information.
6. To save the changes, click **OK** and then click **Apply**.

To change the connection or trace information for a JDBC Fanout driver instance in Designer:

1. In the Outline view or Modeler, right-click the driver icon, then select **Fanout Configuration**.
2. In the Fanout configuration page, select the driver instance in the left navigation and change the driver’s connection or trace settings.

3. To save the changes, click **Save** in Designer's main menu.
4. Deploy the driver.
 - ♦ **Filter:** The driver filter includes two additional classes: DirXML-Resource and DirXML-Driver to allow you to handle dynamic changes to the connection objects. The changes made to the connection objects do not require you to restart the Fanout driver. Depending on the changes made to the connection objects, the Fanout agent restarts or stops the JDBC driver instances.

For more information about the general concepts of the JDBC driver, see [Understanding the Identity Manager Driver for JDBC](#) in the *NetIQ Identity Manager Driver for JDBC Implementation Guide*.

Database Concepts

For more information about the database concepts, see [Database Concepts](#) in the *NetIQ Identity Manager Driver for JDBC Implementation Guide*.

2 Installing the Fanout Driver Components

The Fanout driver and the Fanout agent files are included in the Identity Manager 4.6 installation media. The driver files are automatically installed on the Identity Manager server at the same time as the Identity Manager engine. The installation program extends the Identity Vault's schema and installs the driver shim. The installation program provides an option to install the Fanout agent. For more information, see [“Installing the Fanout Agent with Identity Manager 4.8” on page 19](#).

After the agent is installed, configure the agent to suit your requirement and then create the driver. For more information, see [Chapter 3, “Configuring the Fanout Agent,” on page 25](#).

If you are upgrading your existing driver, perform the following tasks:

- ◆ Update the current driver packages in Designer. For more information, see [“Creating a Fanout Driver Object in Designer” on page 33](#).
- ◆ Migrate your existing Fanout agent configuration to the new Fanout agent installed with Identity Manager 4.6. For more information, see [“Migrating the Fanout Agent Configuration” on page 21](#).

Prerequisites for Installing the Driver

Before installing the driver, ensure that the following software is installed on your computer:

- ◆ Identity Manager 4.6 or later

For more information about Identity Manager and its prerequisites, see the Setup Guide for your platform at the [Identity Manager documentation \(https://www.netiq.com/documentation/identity-manager-47\)](https://www.netiq.com/documentation/identity-manager-47) website.

- ◆ ActiveMQ 5.15.2 at a minimum
- ◆ Identity Manager Driver for Managed System Gateway version 4.0.2 or later
This driver is required for data collection to work with the Fanout driver.
- ◆ Identity Manager Driver for Data Collection Services version 4.0.1 or later with base package 2.3.0.20151207151321
This driver is required for data collection to work with the Fanout driver.
- ◆ Identity Manager Collector for auditing version 2011.1r3 or later
This collector is required for data collection to work with the Fanout driver.
- ◆ Third-party JDBC driver
This driver is required for connecting the Fanout driver to the target database.

Installing the Fanout Agent with Identity Manager 4.8

You can install the Fanout agent with the Identity Manager engine and select the **Fanout Agent** option during the installation.

The driver files are automatically installed on the Identity Manager server at the same time as the Identity Manager engine. For installing the Fanout agent, you must select the **Fanout Agent** option during the installation.

Linux

- 1 Log in as root or administrator on the computer where you want to install the Identity Manager engine.
- 2 Download the `Identity_Manager_4.8_Linux.iso` file from the NetIQ Downloads website.
- 3 Mount the downloaded `.iso`.
- 4 From the root directory of the `.iso` file, run the following command:
- 5 Read through the license agreement.
- 6 Enter `y` to accept the license agreement.
- 7 Decide the Identity Manager server edition you want to install. Enter `y` for Advanced Edition and `n` for Standard Edition.
- 8 Select Identity Manager Engine and complete the installation.

Windows

- 1 Log in as root or administrator on the computer where you want to install the Identity Manager engine.
- 2 From the directory that contains the installation files, run `idm_install.exe`.
- 3 Accept the license agreement, and then click **Next**.
- 4 In the Select Components window on Windows, select **Fanout Agent**.

When you select this option, the installer creates a `FanoutAgent` directory under `/opt/novell/dirxml/` on Linux and `C:\NetIQ\IdentityManager\` on Windows. The `FanoutAgent` directory includes the `bin` and `lib` directories.

For more information about the options, see “[Planning to Install Identity Manager](#)” in the *NetIQ Identity Manager Setup Guide for Windows*.
- 5 (Optional) To select specific drivers for the individual components, complete the following steps:
 - 5a Click **Customize the selected components**, and then click **Next**.
 - 5b Expand **Drivers** under the component that you want to install.
 - 5c Select the drivers that you want to install.
- 6 Click **Next**.
- 7 In the Activation Notice window, click **OK**.
- 8 For Authentication, specify a user account and its password with sufficient rights in eDirectory to extend the schema. Specify the user name in the LDAP format. For example, `cn=admin,o=company`.
- 9 For Pre-Installation Summary, verify the settings.
- 10 Click **Install**.
- 11 Activate Identity Manager.
- 12 Copy your existing Fanout configuration to the new configuration file. For more information, see [Migrating the Fanout Agent Configuration](#).
- 13 To create and configure your driver objects, consult the specific guide for that driver. For more information, see [Identity Manager Drivers documentation website](#).
- 14 (Optional) For the default installation locations, see `/tmp/idmInstall.log`.

Migrating the Fanout Agent Configuration

As part of upgrading your existing driver, you must copy your current Fanout configuration to the new configuration file after the Fanout agent is installed in the following default locations:

- ♦ **Linux:** `/opt/novell/dirxml/fanoutagent`
- ♦ **Windows:** `C:\NetIQ\IdentityManager\FanoutAgent`

Identity Manager provides a migration utility to help you copy the existing configuration to the newly installed Fanout agent directory. The utility is located in the `bin` directory of the new Fanout agent installation directory on both Linux and Windows platforms.

To copy the existing configuration, perform the following steps:

- 1 Stop the Fanout agent.
- 2 Copy the required third party jar files to the `lib` directory in the new Fanout agent directory.
- 3 Copy the existing configuration.

3a Navigate to `bin` in the installation directory of the new Fanout agent and locate the migration script for your platform.

- ♦ **Linux:** `fanoutMigration`
- ♦ **Windows:** `fanoutMigration.bat`

3b Run the migration script using the following syntax:

```
fanoutMigration <existing location of the Fanout agent>
```

For example, `./fanoutmigration /root/NIIdM_Driver_4.5_JDBCFanout`

where `/root/NIIdM_Driver_4.5_JDBCFanout` is the location of the existing Fanout agent.

NOTE: In addition to copying the configuration information, the command appends two new attributes to the configuration file. These attributes are required for setting the secure protocol version for SSL communication and for enabling the SuiteB communication. For more information, see [“Generating the Default Configuration File” on page 25](#).

- 3c** (Optional) If the configuration is successfully copied to the new location, delete the existing directory.
- 4 Start the Fanout agent by specifying the absolute path of the new Fanout agent configuration file in the command.

The Fanout agent configuration is located in `/opt/novell/dirxml/fanout/config` on Linux and `C:\NetIQ\IdentityManager\FanoutAgent\config` on Windows.

Installing ActiveMQ

You must install ActiveMQ on a server other than the Identity Manager server to avoid port conflict. If you install ActiveMQ on the Identity Manager server, ActiveMQ installed by the User Application and ActiveMQ used by the Fanout agent attempt to use the same default port (61616). To avoid this situation, change the port number to any available port number. This consideration also applies to the Sentinel server.

To install ActiveMQ:

- 1 Download ActiveMQ 5.15.2 from the [Apache ActiveMQ 5.15.2 Download page](#).
- 2 Unzip the file to a preferred location on your computer.

- 3 Copy the `activemq-all-<version>.jar` file from the extracted file to your Identity Manager and Fanout Agent installation directories.

For example, `/opt/novell/eDirectory/lib/dirxml/classes/` or `C:\NetIQ\IdentityManager\NDS\lib` directory for Identity Manager installation and `/opt/novell/dirxml/fanoutagent/lib` or `C:\NetIQ\IdentityManager\FanoutAgent\lib` directory for Fanout agent installation.

- 4 Restart eDirectory.

Starting ActiveMQ

To start an ActiveMQ instance, navigate to `<activemq unzipped path>/bin` and perform one of the following actions:

- ♦ **Linux:** Run the `./activemq start` command.
- ♦ **Windows:** Run the `.activemq.bat start` command.

Working with ActiveMQ

You can perform several management tasks by using ActiveMQ. For example:

Change default port : To change the default ports, perform the following steps:

- 1 Navigate to `<AMQ Installation Directory>/conf/activemq.xml`.
- 2 In a text editor, open `activemq.xml`.
- 3 Under `<transportConnectors>`, change the ports as required.

For example, change the port from 61616 to a different port for TCP protocol:

```
<transportConnector name="openwire" uri="tcp://
0.0.0.0:61616?maximumConnections=1000&wireFormat.maxFrameSize=104857600"/>
```

Change Web console port: To change the Web console port, complete the following steps:

- 1 Navigate to `<AMQ Installation Directory>/conf/jetty.xml`.
- 2 Open the `jetty.xml` file in a text editor.
- 3 Change the Web console port from 8161 to a different port.
- 4 Save the `jetty.xml` file.

Monitor the queues managed by the Fanout Agent: To view the queues managed by the Fanout agent, specify `https://<ActiveMQ IP Address/servername>:<Port>/admin/queues.jsp` in the address bar of your browser. For example: `https://192.168.0.1:8161/admin/queues.jsp`.

Setting Up ActiveMQ Startup Service

This section provides details about setting up ActiveMQ as a startup service.

- ♦ [“Enabling the ActiveMQ Service on Linux” on page 23](#)
- ♦ [“Enabling the ActiveMQ Service on Windows” on page 23](#)

Enabling the ActiveMQ Service on Linux

- 1 Set the path for the `JAVA_HOME` environment variable.
- 2 Set the path for the `ACTIVEMQHOME` environment variable in the `/root/.profile` file to point to the ActiveMQ installation location. The `.profile` file needs to be created if it does not exist.
- 3 Navigate to the ActiveMQ installation directory and run `activemqAddSvc` by executing the following command:

```
.activemqAddSvc
```

- 4 To start the ActiveMQ service, execute the following command:

```
/etc/init.d/activemqdxml start
```

To stop the ActiveMQ service, execute the following command:

```
/etc/init.d/activemqdxml stop
```

Enabling the ActiveMQ Service on Windows

To add the ActiveMQ service to the Windows services list, navigate to `ACTIVEMQ_HOME/bin/win64` and run `InstallService.bat`.

This creates the ActiveMQ service in the Windows service list. This service automatically starts when the system reboots.

To remove the ActiveMQ service from the Windows services list, run `UninstallService.bat`.

Upgrading ActiveMQ

- 1 Download the ActiveMQ *<version>* from the ActiveMQ Download page.
- 2 Extract the file to a preferred location on your computer.
- 3 Stop the Fanout Agent. For more information, see [“Stopping the Fanout Agent” on page 32](#).
- 4 Copy the extracted file to your Identity Manager and Fanout Agent installation directories.
For example, `/opt/novell/eDirectory/lib/dirxml/classes/` or `C:\NetIQ\IdentityManager\NDS\lib` directory for Identity Manager installation and `/opt/novell/dirxml/fanoutagent/lib` or `C:\NetIQ\IdentityManager\FanoutAgent\lib` directory for Fanout agent installation.
- 5 Start the Fanout Agent. For more information, see [“Starting the Fanout Agent” on page 28](#).

3 Configuring the Fanout Agent

After the Fanout agent is installed, configure the agent to suit your environment by using the scripts provided in the Fanout agent installation folder:

- ♦ **Linux:** `/opt/novell/dirxml/fanoutagent`
- ♦ **Windows:** `C:\NetIQ\IdentityManager\FanoutAgent`

NOTE: Set the `JAVA_HOME` environment variable on your platform.

The following sections provide instructions to configure the Fanout agent:

- ♦ [“Generating the Default Configuration File” on page 25](#)
- ♦ [“Starting the Fanout Agent” on page 28](#)
- ♦ [“Managing the Fanout Agent” on page 29](#)
- ♦ [“Enabling Auditing for the Fanout Agent” on page 32](#)
- ♦ [“Stopping the Fanout Agent” on page 32](#)

Generating the Default Configuration File

When you run the Fanout agent without any options, the agent creates the default configuration file and then stops running. To run the Fanout agent, execute one of the following commands based on your platform:

- ♦ **Linux:** `./startAgent`
- ♦ **Windows:** `startAgent.bat`

The Fanout agent creates the following directories in default installation path of the Fanout agent:

- ♦ **config:** This directory contains the default configuration file for the Fanout agent. The following table lists the parameters included in the default configuration file. Except `netiq.fanoutagent.connection.url`, you can leave other parameters unchanged. Specify the URL of the ActiveMQ instance in this parameter.

Parameter	Description
<code>netiq.fanoutagent.trace.level</code>	Fanout Agent trace level. Range 1-5. Setting the value of this parameter to 3 provides most of the XML and operation traces.
<code>netiq.fanoutagent.trace.file</code>	Path of the Fanout agent trace file.
<code>netiq.fanoutagent.instance.name</code>	Name of the Fanout agent instance.
<code>netiq.fanoutagent.connection.url</code>	Connection URL of the ActiveMQ instance.
<code>netiq.fanoutagent-config.recv.queue</code>	Configuration queue to receive the initialization parameters.
<code>netiq.fanoutagent-config.snd.queue</code>	Configuration queue to query specific configurations.
<code>netiq.fanoutagent-query.in.recv.queue</code>	Query-in queue to receive query responses from the Identity Vault.

Parameter	Description
netiq.fanoutagent-query.in.send.queue	Query-in queue to send queries to the Identity Vault.
netiq.fanoutagent-sub.event.recv.queue	Subscriber event queue to receive the Subscriber events.
netiq.fanoutagent-sub.event.send.queue	Subscriber event queue to send the Subscriber event status.
netiq.fanoutagent-sub.delayed.event.send.queue	Subscriber event queue to send the status of the delayed event.
netiq.fanoutagent-query.out.recv.queue	Query-out queue to receive the query for the agent.
netiq.fanoutagent-query.out.send.queue	Query-out queue to send the query response.
netiq.fanoutagent.cmd.trace.level	Trace level for the Fanout command server.
netiq.fanoutagent.cmd.srv.ip	IP address to which the command server establishes connection with. This helps you to restrict the command server to listen to a specific interface.
netiq.fanoutagent.cmd.srv.port	Port number on which the command server listens.
netiq.fanoutagent.cmd.trace.file.count	Number of trace files available for the command server. After the limit is reached, the older file are automatically deleted.
netiq.fanoutagent.cmd.allow.http	Parameter to disable <code>https</code> on the command server.
netiq.fanoutagent.cmd.trace.file.size	Size of the Fanout agent command server trace files in MB.
netiq.fanoutagent.cmd.keystore.file	Path to the keystore used by the command server.
netiq.fanoutagent.cmd.trace.dir	The directory where the Fanout agent command server traces are created.
netiq.fanoutagent.connection.truststore.file	Path to the truststore file used for mutual authentication for a secure connection.
netiq.fanoutagent.connection.keystore.file	Path to the keystore file used for mutual authentication for a secure connection.
netiq.fanoutagent.connection.enforceSuiteB	The default setting is false. This means that the Fanout agent does not use Suite B cryptographic algorithms to communicate with the Fanout driver. NOTE: This parameter is included in Fanout agent 1.1.
netiq.fanoutagent.connection.secureprotocol	Version of the TLS protocol that the Fanout agent uses to connect to the Fanout driver. The Fanout agent supports TLSv1, TLSv1_1, and TLSv1_2. NOTE: This parameter is included in Fanout agent 1.1.

Parameter	Description
netiq.fanoutagent-sub.event.max.retry	<p>The maximum limit for retrying an event. The default value is -1. This allows the JDBC instance running in the Fanout agent to retry an event for every 30 seconds until a success or an error is received from the JDBC driver shim.</p> <p>If you want a JDBC instance to retry an event for a finite number of times until a success or an error is received from the driver shim, set the parameter to a value greater than or equal to zero.</p> <p>For example, when you set the value to 3, the instance retries an event three times. If the instance receives a retry status after the retry limit has been exhausted, the Fanout agent discards that event and returns an error status to the Identity Manager engine.</p> <p>NOTE: The retrying of events in one instance does not affect the event processing in other instances running in the Fanout agent.</p>

You can change the default configuration of the Fanout agent to suit your requirement. Changes are dynamically reflected in some parameters. For changes to take effect in other parameters, restart the Fanout agent. The following parameters are dynamically reflected:

- ◆ netiq.fanoutagent.trace.level
- ◆ netiq.fanoutagent.trace.file
- ◆ netiq.fanoutagent.cmd.trace.level
- ◆ netiq.fanoutagent.cmd.trace.file.count

Password changes are dynamically reflected. For example, agent password values is changed dynamically. For future commands, you must use the new agent password.

NOTE: If you run the Fanout agent without any options after customizing the default configuration file, the agent will overwrite the changes made to the parameters. NetIQ recommends that you rename the configuration file to a different name to avoid overwriting the file and use the renamed file for subsequent operations.

- ◆ **logs:** This directory contains the trace files.
- ◆ **tmp:** This directory contains the temporary files created by the Fanout agent.

The Fanout agent also creates `.profile` file under the `root` folder. This file contains information about the Fanout installation directory and the current Java path that is used by the Fanout agent. The following example is a `.profile` file:

```
JAVA_HOME=/opt/novell/jdk1.7.0_25/jre
PATH=$PATH:/opt/novell/jdk1.7.0_25/jre/bin
FANOUTHOME=/opt/novell/mysql-fanout/agent
```

NOTE: For multiple Fanout agents, you require equal number of ActiveMQs. If you are using the same ActiveMQ with multiple Fanout agents, you need to manually clean the ActiveMQ queues before using that ActiveMQ with a different Fanout agent. To clean an ActiveMQ queue, use ActiveMQ Web console. ActiveMQ also provides other options for cleaning the queues. For more information, see [ActiveMQ documentation](#).

Starting the Fanout Agent

You can start the Fanout agent by passing the `startAgent` command in the command prompt or by using the startup script, which enables the agent as a startup service.

Before starting the Fanout agent, ensure that your server meets the following requirements:

- ♦ ActiveMQ is running.
This allows the Fanout agent to initialize the required queues in ActiveMQ when the agent starts. If ActiveMQ is not running, the agent returns an error and does not start properly.
- ♦ The Java and Curl tool are included in the system path of your operating system.
You use Java and Curl tool to manage the Fanout agent. If Curl is not present in your environment, download the SSL version of the Curl binary from [Curl Releases and Downloads](#).
- ♦ Copy the appropriate third-party JDBC connector to the `lib` directory of the Fanout agent default installation folder. For example, `ojdbc6.jar`. For more information, see the [Supported Third Party JDBC Drivers](#) in the *NetIQ Identity Manager Driver for JDBC Implementation Guide*.

To start the Fanout Agent in the command prompt:

- 1 Open a command prompt.
- 2 Run the `startAgent` command and pass the default configuration file name as a parameter in the command.

```
startAgent -config <FanoutAgent Installation Location>/config/fanoutagent-  
config.properties
```

The installation path will vary depending on the Fanout agent version you are running.

- 3 (Optional) To change the agent and the shim passwords, use the `setPassword` command.
The default passwords are **netiq**. For more information, see [“Managing the Fanout Agent” on page 29](#).
- 4 (Optional) To change the keystore password, use the `setKSPassword` command.
The default password for the keystore is **netiq123**.
- 5 (Optional) To change the encryption key, use the `setEncryptionKey` command.
The default value for the key is **netiq**.
- 6 (Optional) Establish a secure connection between the Fanout agent and ActiveMQ. For more information, see [Chapter 6, “Securing Fanout Driver Communication,” on page 47](#).

IMPORTANT: Ensure that you specify the same values for these parameters during Fanout driver configuration.

To enable the Fanout agent service to automatically start when the system starts, perform the following actions for your platform:

Linux:

- 1 Set the `JAVA_HOME` environment variable.
- 2 Copy the `fanoutd.xml` file from `/opt/novell/dirxml/fanoutagent/bin` to `/etc/init.d`.
- 3 Run the `fanoutAgentAddSvc` file from `/opt/novell/dirxml/fanoutagent/bin` by executing the following command:

```
./fanoutAgentAddSvc
```

- 4 Start the Fanout agent service by executing the following command:

```
/etc/init.d/fanoutdxml start
```

To stop the Fanout agent service, execute the following command:

```
/etc/init.d/fanoutdxml stop
```

Windows:

- 1 Set the `JAVA_HOME` environment variable.
- 2 Run `vcredis_x64.exe` file from `C:\NetIQ\IdentityManager\FanoutAgent\bin`.
- 3 Install the Fanout agent service by executing the following command:

```
C:\NetIQ\IdentityManager\FanoutAgent\bin\FanoutAgentSvc.exe -i
```

The display name of the service is **Fanout Agent Service**.

NOTE: If both ActiveMQ and Fanout agent services are running on the same server, ensure that ActiveMQ service starts before the Fanout agent service. To set the Fanout agent service to a delayed start, run the following command on Windows:

```
sc config FanoutAgent start= delayed-auto
```

- 4 Start the Fanout agent service.

To remove the Fanout agent service on Windows, execute the following command:

```
C:\NetIQ\IdentityManager\FanoutAgent\bin\FanoutAgentSvc.exe -u
```

IMPORTANT: You can run only one Fanout agent instance on a specific server as a service. However, if you run a Fanout agent as an application, Identity Manager allows you to run multiple instances of the Fanout agent on the same server by using separate configuration files.

The Fanout agent service loads the configuration properties only from `C:\NetIQ\IdentityManager\FanoutAgent\config` path. Ensure that the configuration properties file name is `fanoutagent-config.properties` (default file name).

A Fanout agent instance that is started as a service must be stopped as a service only. This means that you should not stop the agent using the `stopAgent` command or REST endpoint if it is started as a service.

Managing the Fanout Agent

You can manage the Fanout agent by using the scripts from the Fanout agent installation directory. [Table 3-1](#) lists the commands to manage the Fanout agent. To view the help, invoke the command with `--help` option.

Table 3-1 *Commands to Manage the Fanout Agent*

Command	Description	Usage
startAgent	Starts the Fanout agent instance	startAgent -config startAgent -config <FanoutAgent Installation Location>/config/fanoutagent-config.properties
stopAgent	Stops the Fanout Agent instance	stopAgent -config stopAgent -config <FanoutAgent Installation Location>/config/fanoutagent-config.properties netiq
getservices	Fetches the status of the JDBC driver instances loaded by the Fanout agent and writes the status to a file.	getServices -config <FanoutAgent Installation Location>/config/fanoutagent-config.properties 0 ./services.txt netiq
startService	Starts the specified JDBC driver instance. You can get the JDBC driver instance name from the output file generated by the getServices command.	startService -config <FanoutAgent Installation Location>/config/fanoutagent-config.properties serviceName netiq
stopService	Stops the specified JDBC driver instance. You can get the JDBC driver instance name from the output file generated by the getServices command.	stopService -config <FanoutAgent Installation Location>/config/fanoutagent-config.properties serviceName netiq
setPassword	Sets the agent and shim passwords for the Fanout agent instance.	setPassword -config <FanoutAgent Installation Location>/config/fanoutagent-config.properties netiq microfocus netiq microfocus
setAMQKSPassword	Sets the Keystore password used for mutual authentication with ActiveMQ when SSL is enabled	setAMQKSPassword -config <FanoutAgent Installation Location>/config/fanoutagent-config.properties netiq microfocus
setAMQTSPassword	Sets the Truststore password used for mutual authentication with ActiveMQ when SSL is enabled.	setAMQTSPassword -config <FanoutAgent Installation Location>/config/fanoutagent-config.properties netiq microfocus

Command	Description	Usage
setKSPassword	Sets the keystore password for the Fanout command server.	<p>setKSPassword -config <fanoutagent-config.properties in the FanoutAgent Installation Location> <type 1 or type 2> <keystore or key in the keystore password> <agent password></p> <p>where <code>type 1</code> specifies the keystore password and <code>type 2</code> specifies the key password in the keystore.</p> <p>Examples:</p> <ul style="list-style-type: none"> ◆ <code>setKSPassword -config /opt/novell/dirxml/config/fanoutagent-config.properties 1 netiq123 novell</code> ◆ <code>setKSPassword -config C:\NetIQ\IdentityManager\NDS\Agent\config\fanoutagent-config\properties 1 netiq123 novell</code> ◆ <code>setKSPassword -config /opt/novell/dirxml/config/fanoutagent-config.properties 2 netiq123 novell</code> ◆ <code>setKSPassword -config C:\NetIQ\IdentityManager\NDS\Agent\config\fanoutagent-config\properties 2 netiq123 novell</code>
setConfig	<p>Changes the configuration options as specified in the input file.</p> <p>NOTE: Stop the agent before changing <code>netiq.fanoutagent.trace.level</code>, <code>netiq.fanoutagent.cmd.srv.ip</code>, <code>netiq.fanoutagent.cmd.srv.port</code>, and <code>netiq.fanoutagent.cmd.allow.http</code> parameters and then restart the agent for the changes to take effect.</p>	<p>setConfig -config <FanoutAgent Installation Location>/config/fanoutagent-config.properties ./sample_cfg.json netiq</p>
setEncryptionKey	Sets the encryption key. You can use this key to encrypt or decrypt ActiveMQ messages.	<p>setEncryptionKey -config config/fanoutagent-config.properties netiq netiq</p>
fanoutMigration	<p>Migrates the Fanout agent files from the existing location to the new installation location of the agent.</p> <p>Use this command to migrate the agent version 1.0 to 1.1 shipped with Identity Manager 4.6.</p>	<p>fanoutMigration <existing location of the Fanout agent></p> <p>For information about how this command is used, see “Migrating the Fanout Agent Configuration” on page 21.</p>

Enabling Auditing for the Fanout Agent

The Fanout agent supports both XDAS and legacy auditing solutions. The files required for enabling auditing are included in the Identity Manager installation package.

For more information about configuring the auditing, see [NetIQ Identity Manager - Configuring Auditing in Identity Manager](#).

To enable XDAS auditing, navigate to the `<FanoutAgent Default Installation Location>/config` folder and rename the `xdasconfiguration.properties.template` file to `xdasconfiguration.properties`. For more information about configuring XDAS, see [NetIQ Identity Manager - Configuring Auditing in Identity Manager](#).

For more information about XDAS auditing, see [XDASv2 Administration Guide](#).

NOTE: The Fanout agent does not support XDAS event caching.

Stopping the Fanout Agent

You can stop the Fanout agent by passing the `stopAgent` command in the command prompt.

To stop the Fanout Agent in the command prompt:

- 1 Open a command prompt.
- 2 Run the `stopAgent` command and pass the default configuration file name as a parameter in the command.

```
stopAgent -config <FanoutAgent Installation Location>/config/fanoutagent-  
config.properties <agent password>.
```

The installation path will vary depending on the Fanout agent version you are running.

4 Creating New Fanout Driver

After the Fanout driver files are installed on the server where you want to run the driver object, you can create a driver object in the Identity Vault. Creating a driver object consists of installing the driver packages and then modifying the driver configuration to suit the environment.

- ◆ [“Creating a Fanout Driver Object in Designer” on page 33](#)
- ◆ [“Activating the Driver” on page 40](#)
- ◆ [“Adding Packages to an Existing Driver” on page 40](#)

You must install the Fanout components before configuring a new Fanout driver object. For more information, see [Chapter 2, “Installing the Fanout Driver Components,” on page 19](#).

Creating a Fanout Driver Object in Designer

Creating the driver includes installing the Fanout driver packages and then modifying the configuration to suit your environment.

- ◆ [“Importing the Driver Packages” on page 33](#)
- ◆ [“Installing the Driver Packages” on page 34](#)
- ◆ [“Configuring the Driver Object” on page 36](#)
- ◆ [“Configuring the Database Connections for the Driver” on page 37](#)
- ◆ [“Deploying the Driver Object” on page 39](#)
- ◆ [“Starting the Driver” on page 39](#)
- ◆ [“Managing the Connection Objects” on page 40](#)

Importing the Driver Packages

You can update the driver packages at any time. The driver packages are stored in the Package Catalog. Packages are initially imported into the Package Catalog when you create, import, or convert a project. Ensure that you import the latest packages into the Package Catalog before installing the driver.

To verify the latest packages in the Package Catalog:

- 1 Open Designer.
- 2 In the toolbar, click **Help** > **Check for Package Updates**.
- 3 Click **OK** to import the package updates.
If prompted to restart Designer for the changes to take effect, click **Yes**, save your project, and then wait for the Designer to restart.
or
Click **OK** if there are no package updates.
- 4 Continue with [“Installing the Driver Packages” on page 34](#).

Installing the Driver Packages

- 1 In Designer, open your project.
- 2 In the Modeler, drag and drop a supported JDBC database from the Designer palette.
For example, Oracle. For information about supported databases, see [“Supported Databases” on page 16](#).
- 3 Select the Fanout Base from the list of available base packages, then click **Next**.
- 4 Select the **Synchronization Mode**. For more information, see [“Data Synchronization” on page 15](#).
- 5 Select the optional features to install for the Fanout driver, then click **Next**.
All options are selected by default. The options are:
 - ♦ **Entitlements Support:** These packages contain the policies that provision the user accounts on the connected database. For more information, see the [NetIQ Identity Manager Entitlements Guide](#).
 - ♦ **Data Collection:** These packages contain the policies that enable the driver to collect data for reports. If you are using Identity Reporting, ensure that this option is selected. For more information, see [NetIQ Identity Reporting: User’s Guide to Running Reports](#).
 - ♦ **Account Tracking:** These packages contain the policies that enable account tracking information for reports. If you are using Identity Reporting, ensure that this option is selected. For more information, see [NetIQ Identity Reporting: User’s Guide to Running Reports](#).
- 6 (Conditional) If there are package dependencies for the packages you selected to install for this driver, you must install them to install the selected package. Click **OK** to install the package dependency listed.
- 7 (Conditional) If more than one type of package dependency is installed, Designer displays separate configuration pages for each package. Click **OK** to install any additional package dependencies.
- 8 (Conditional) The Common Settings page is displayed only when the Common Settings package is installed as a dependency. On the Install Common Settings page, specify the common settings for User and Group containers:
 - ♦ **User Container:** Select the Identity Vault container where the user accounts will be added in the Identity Vault. This value becomes the default for all drivers in the driver set.
 - ♦ **Group Container:** Select the Identity Vault container where the groups will be added in the Identity Vault. This value becomes the default for all drivers in the driver set.
- 9 Click **Next**.
When all dependencies are installed, you must configure the components.
- 10 On the Driver Information page, specify a name for the driver that is unique within the driver set, and then click **Next**.
- 11 On the Application Authentication page, fill in the following information:
 - ♦ **Connection Information:** Specify the URL of the ActiveMQ instance to which this driver connects to. For example, `tcp://111.1.1.1:61616`.
 - ♦ **Synchronization Filter:** Select the synchronization filter. For more information, see [“Database Scoping Parameters”](#) from the [NetIQ Identity Manager Driver for JDBC Implementation Guide](#).
 - ♦ **Synchronization Model:** Select the synchronization model based on the synchronization mode specified in Step 4.
- 12 Do not change the default values of the remaining parameters on this page, then click **Next**.

- 13 On the Entitlements Information page, specify a name for the **Account Entitlement Value** field, then click **Next**.
- 14 (Conditional) This page is displayed only if you selected to install the Data Collection and Account Tracking groups of packages. On the Managed System Information page, fill in the following fields to define your connected database application:
- ◆ **Name:** Specify a descriptive name for the connected database application. The name is displayed in reports.
 - ◆ **Description:** Specify a brief description for the connected database application. The description is displayed in reports.
 - ◆ **Location:** Specify the physical location of the connected database application. The location is displayed in reports.
 - ◆ **Vendor:** Specify the vendor of the connected database application. This information is displayed in reports.
 - ◆ **Version:** Specify the version of the connected database application. The version is displayed in reports.
- 15 Click **Next**.
- 16 (Conditional) This page is displayed only if you selected to install the Managed System packages and the Account Tracking packages. On the Install Managed System Information page, fill in the following fields to define the classification of the connected database application. This information is displayed in the reports. The options are:
- ◆ **Classification:** Select the classification of the connected database application. This information is displayed in the reports. Your options are:
 - ◆ Mission-Critical
 - ◆ Vital
 - ◆ Not-Critical
 - ◆ OtherIf you select **Other**, you must specify a custom classification for the JDBC system.
 - ◆ **Environment:** Select the type of the connected database application environment. The options are:
 - ◆ Development
 - ◆ Test
 - ◆ Staging
 - ◆ Production
 - ◆ OtherIf you select **Other**, you must specify a custom classification for the database application.
- Click **Next**.
- 17 (Conditional) This page is displayed only if you selected to install the Data Collection and Account Tracking groups of packages. Fill in the following fields to define the ownership of the connected database application:
- ◆ **Business Owner:** Select a user object in the Identity Vault that is the business owner of the database application. 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 of the database application. This can only be a user object, not a role, group, or container.

Click **Next**.

- 18 (Conditional) This page is displayed only if you selected to install the Account Tracking groups of packages. On the Account Tracking Initial Configuration page, fill in the following fields:
- ♦ **Fanout Database Type:** Select the required database. The Fanout supports databases such as, MySQL, SQL Server, Sybase, and Oracle.
 - ♦ **Synchronization Model:** Specify the mode of data synchronization.
 - ♦ **User Table:** This field is populated based on your selection in the **Synchronization Model**. Specify the table or view in the connected database for which account tracking is enabled. By default, the value is `usr`.
 - ♦ **Realm:** Specify the name of the realm that uniquely identifies the location of user accounts in the connected database. For example, `mysql.indirect.usr`, where `mysql` is the database name with the indirect data synchronization model, and `user` is the table or view in the connected database for which account tracking is enabled.

Click **Next**.

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

The driver is now created. To modify the configuration settings, proceed to the “[Configuring the Driver Object](#)” on page 36.

Configuring the Driver Object

After importing the packages and creating the driver object, configure the driver to make it operational. Many settings are available to help you customize and optimize the driver. However, you should first configure the driver parameters located on the Driver Configuration page.

- 1 In Designer, open your project.
- 2 In the Modeler, right-click the driver icon or the driver line, then select **Properties**.
- 3 Click **Driver Configuration** and select the **Driver Parameters** tab.
- 4 In the **Driver Options** tab, specify the queue names in the **Transport** parameters. The different queues are:
 - ♦ **Subscriber Event Queues (SEND/RECV/DELAYED):** The driver uses these queues to send the Subscriber events to the Fanout agent and receive the status of these events.
 - ♦ **Configuration Queues (SEND/RECV):** The driver uses these queues for sending the driver initialization documents and exchanging the handshake documents with the Fanout agent.
 - ♦ **Query-In Queues (SEND/RECV):** The driver uses these queues to receive the queries from the Fanout agent and sends the response of the queries to the Fanout agent.
 - ♦ **Query-Out Queues (SEND/RECV):** The driver uses these queues to send the queries to the Fanout agent and receive the response of the queries from the Fanout agent.

NOTE: If you specified a different name for the queues in these parameters, ensure that the same queue name is used during Fanout agent configuration.

- 5 In the **Fanout Agent Configuration Parameters**, fill in the following information:
 - ♦ **Fanout Shim Password:** Click **Set Password** to specify the Fanout shim password.
 - ♦ **Fanout Agent Password:** Click **Set Password** to specify the Fanout agent password. The Fanout driver uses these passwords for performing handshake with the Fanout agent.
 - ♦ **Encryption Key:** Click **Set Password** to specify the key to encrypt or decrypt the sensitive data before sending the data to the message queues.

NOTE: Ensure that you provide the same value for the Fanout agent and shim passwords and the encryption key. The default passwords are **netiq**.

- 6 To enable the SSL communication between the Fanout driver and ActiveMQ, specify the following information:
- ♦ **AMQ Keystore Path:** The full path to the keystore file. For example, `/root/amq-clients.ks`.
 - ♦ **AMQ Keystore Password:** The password used by the keystore.
 - ♦ **AMQ Truststore Path for SSL Certs:** The full path to the truststore file. For example, `/root/amq-clients.ts`.
 - ♦ **AMQ Truststore Password:** The password used by the truststore.
 - ♦ **Secure Protocol Version:** Specifies the version of the TLS protocol that the driver uses to connect to the Identity Manager engine. Identity Manager supports TLSv1, TLSv1_1, and TLSv1_2.

NOTE: This parameter is included in Fanout driver 1.1.

- ♦ **Enforce Suite B:** Specifies whether the driver should use Suite B for establishing communication with ActiveMQ. When Suite B is enabled, the communication is authenticated with Suite B cryptographic algorithms. This communication is supported only on TLS 1.2 protocol.

NOTE: This parameter is included in Fanout driver 1.1.

For more information about securing communication, see [Chapter 6, “Securing Fanout Driver Communication,” on page 47](#).

- 7 Do not change the default value of **Fanout Shim classname**.

- 8 Do not change the default value of **Matching Attributes**.

The Fanout agent uses **Matching Attributes** to match the objects in the delayed add events. This parameter must be schema-mapped equivalent of the attributes that are used in the object matching policy. If you are using different attributes, specify the attribute names according to the connected system schema.

- 9 The **Normal JDBC Driver settings** section for the Fanout driver is similar to the JDBC driver. For more information about these parameters, see [“Driver Parameters” in the NetIQ JDBC Driver Guide](#).

After completing the configuration tasks, continue with [“Configuring the Database Connections for the Driver” on page 37](#).

Configuring the Database Connections for the Driver


Designer lets you configure multiple database connections for the Fanout driver. Each JDBC driver instance loaded by the Fanout agent uses this information to connect to the database and for tracing purposes.

Alternatively, you can run the `createConnLDIF` script to create an LDIF file that includes the connection objects for the Fanout driver. However, you can run the script only after deploying the driver. For more information, see [“Configuring the Database Connections by Using the createConnLDIF Script” on page 38](#).

The advantage of using Designer is that it allows you to manage the connections after they are configured. If you use the `createConnLDIF` script for creating the connections, the script does not provide this flexibility.

- ♦ “Configuring the Database Connections in Designer” on page 38
- ♦ “Configuring the Database Connections by Using the `createConnLDIF` Script” on page 38

Configuring the Database Connections in Designer

- 1 Open your project in Designer.
- 2 In the Modeler, right-click the driver icon and select **Fanout Configuration**.
- 3 Click  icon to create a Fanout connection.
- 4 Specify the Fanout connection details:


- ♦ **Name:** Specify the name for the new connection.

NOTE: NetIQ restricts the connection object name to 15 characters.

- ♦ **User:** Specify the user name with which the JDBC driver instance will authenticate to the database.
- ♦ **Connection Password:** Specify the password with which the JDBC driver instance will authenticate to the database
- ♦ **Server:** Specify the server with which the JDBC driver instance will connect to. For more information, see [JDBC URL Syntaxes](#) in the *NetIQ Identity Manager Driver for JDBC Implementation Guide*.
- ♦ **Trace Level:** Specify the trace level for the JDBC driver instance. This defines the level for logging the trace messages.
- ♦ **Trace File:** Specify the name of the trace file. This file includes the trace and debugging messages for the JDBC driver instance.

NOTE: In order to have a single trace file for both database connection and the Fanout agent, configure them with the same trace file name.

- ♦ **Trace File Size:** Specify the trace file size. This defines the limit for the trace file. This parameter is not currently supported with the driver.

- 5 Click  icon to enable or disable the selected connection. By default, the connection is disabled.
- 6 Click **Save** on the Designer toolbar.
- 7 (Conditional) To create multiple database connections for the Fanout configuration, repeat Step 3 through Step 6.

Configuring the Database Connections by Using the `createConnLDIF` Script

After deploying the driver, you can run the `createConnLDIF` script to create an LDIF file that allows you to create multiple database connection objects. The `createConnLDIF` file is located in default installation path of the Fanout agent.

To create a database connection:

- 1 Create a CSV file with the required database connection information. A sample CSV file is located in the default installation directory of the Fanout agent.
- 2 Run the `createConnLDIF` script as follows:

```
createConnLDIF [DriverDN in LDAP format] [input csv absolute file path] [output ldif absolute file path]
```

This script creates the LDIF file in the specified location.

- 3 Import the LDIF file into eDirectory using any LDAP tool.

Deploying the Driver Object

After you create the driver in Designer, you can deploy the driver into the Identity Vault.

To deploy the driver:


- 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, and then click **Deploy**.
- 6 Read the success message, and then click **OK**.
- 7 Click **Define Security Equivalence** to assign rights to the driver.
- 8 Click **Exclude Administrative Roles** to exclude users that should not be synchronized.
- 9 Click **OK**.

Starting the Driver

After creating a driver, you must start it. Identity Manager is an event-driven system and starts caching the events once the driver is deployed. These events are processed when you start the driver.

NOTE: NetIQ recommends that you complete the Fanout agent configuration before starting the driver. For more information, see [Chapter 3, "Configuring the Fanout Agent," on page 25](#).

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**.

Managing the Connection Objects

When you make changes to the connection objects and deploy the Fanout driver, you need not restart the driver. Depending on the connection object changes, the Fanout driver starts, stops, or restarts the JDBC driver instances.

When you add, modify, or delete a connection object, the Fanout driver sends the configuration update to the Fanout agent. Based on the changes made, the Fanout agent starts, stops, or restarts the corresponding JDBC driver instance.

NOTE: To detect the changes to the connection objects, Identity Manager provides two additional filter classes namely **DirXML-Resource** and **DirXML-Driver** in the default filter of the Fanout driver.

Ensure that you review the following notes before redeploying the Fanout driver:

- ◆ When you deploy the Fanout driver, all the connection objects are also deployed. If you perform any dynamic updates to the connection objects or if you create the connection object dynamically using Designer, NetIQ recommends that you deploy those connection objects separately to avoid the entire driver restart.
- ◆ If you change the password for a connection object, the connection object must be deployed again. In this case, NetIQ recommends that you do not separately deploy or reconcile the named-passwords for the connection objects.
- ◆ When you use Designer to delete an existing Fanout connection, the connection object is deleted only from Designer and it is not updated in the Identity Vault. To delete an existing Fanout connection object from the Identity Vault, use iManager to manually remove the connection object.

Activating the Driver

The Identity Manager driver for JDBC Fanout requires a separate activation. After purchasing the Integration Module for Database, you will receive activation details in your NetIQ Customer Center.

If you create a new JDBC Fanout driver in a driver set that already includes an activated driver from this integration module, the new driver inherits the activation from the driver set.

If you create the driver in a driver set that has not been previously activated with this integration module, the driver will run in the evaluation mode for 90 days. You must activate the driver with this integration module during the evaluation period; otherwise, the driver will be disabled.

If driver activation has expired, the trace 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*.


Adding Packages to an Existing Driver

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

- 1 Right-click the driver, then click **Properties**.
- 2 Click **Packages**, then upgrade the already installed driver Base package.
 - 2a Select the package from the list of packages, then click the **Select Operation** cell.
 - 2b Click **Upgrade** from the drop-down list, then click **Apply**.

2c Click **OK** to close the Package Management page.

You can upgrade the Password Synchronization package in a similar way.

- 3** Click the **Add Packages** icon .
- 4** Select the packages to install.
- 5** (Optional) If you want to see all available packages for the driver, clear 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.
- 6** Click **Apply** to install all of the packages listed with the **Install** operation.
- 7** (Conditional) Fill in the fields with appropriate information to install the package you selected for the driver, then click **Next**.
- 8** Read the summary of the installation, then click **Finish**.
- 9** Click **OK** to close the Package Management page after you have reviewed the installed packages.
- 10** Modify the driver configuration settings. See [“Configuring the Driver Object” on page 36](#).
- 11** Deploy the driver. See [“Deploying the Driver Object” on page 39](#).
- 12** Start the driver. See [“Starting the Driver” on page 39](#).
- 13** Repeat [Step 1](#) through [Step 9](#) for each driver where you want to add the new packages.

5 Upgrading an Existing Driver

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

What's New

- ♦ [“What's New in Version 1.1.0.0” on page 43](#)

What's New in Version 1.1.0.0

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

Upgrading to Version 1.1.0.0

In general, the Fanout driver upgrade process involves upgrading the installed driver packages and updating the existing Fanout agent and driver files.

This section provides general instructions for updating a driver. For information about updating the driver to a specific version, search for that driver patch in the [NetIQ Patch Finder Download Page](#) and follow the instructions from the Readme file accompanying the driver patch release.

- ♦ [“Upgrading the Installed Packages” on page 43](#)
- ♦ [“Updating the Fanout Agent and Driver” on page 44](#)

Upgrading the Installed Packages

- 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. For more information about managing packages, see 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*.

Updating the Fanout Agent and Driver

- ◆ [“Prerequisites” on page 44](#)
- ◆ [“Updating the Fanout Agent” on page 44](#)
- ◆ [“Updating the Driver Files” on page 45](#)

Prerequisites

- 1 Take a back-up of the current driver configuration.
- 2 Stop the driver instance.
- 3 Stop the Identity Vault.
- 4 In a browser, navigate to the [NetIQ Patch Finder Download Page](#).
- 5 Under **Patches**, click **Search Patches**.
- 6 Specify **Identity Manager *nn* Fanout driver *nn*** in the search box.
- 7 Download and unzip the contents of the patch file to a temporary location on your server.
For example, IDM4.5_FanoutAgent_1110.zip.

Updating the Fanout Agent

- 1 Stop the Fanout agent. For more information, see [“Stopping the Fanout Agent” on page 32](#).
- 2 Remove all the versions of the `activemq-all-<Version>.jar` file from your current `<Fanout agent installation>\lib` directory.
- 3 Update the Fanout agent files:
 - ◆ **Linux:** Depending on your Identity Manager installation type as `root` or `non-root`, log in to your server with appropriate rights and run the following command in a command prompt:

```
rpm -Uvh <Fanout Agent Patch File Temporary Location>/linux/netiq-DXMfanout.rpm
```
 - ◆ **Windows:** Navigate to the `<Extracted Fanout Agent Patch File Temporary Location>\windows\FanoutAgent\bin` folder and copy the following files to `<IdentityManager installation>\NDS\lib` folder.
 - ◆ `stopAgent.bat`
 - ◆ `FanoutAgent.jar`
 - ◆ `activemq-all-5.15.2.jar`
- 4 Start the Fanout agent. For more information, see [“Starting the Fanout Agent” on page 28](#).

Updating the Driver Files

1 (Conditional) Stop the driver instance, then stop the Identity Vault if not already stopped:

2 Update the driver files:

- ◆ **Linux:** To upgrade the existing RPM, log in as `root` and run the following command in a command prompt:

```
rpm -Uvh <Driver Patch File Temporary Location>/linux/novell-  
DXMLfanoutdriver.rpm
```

For example, `rpm -Uvh <IDM4.5_FanoutAgent_1110.zip>/linux/novell-
DXMLfanoutdriver.rpm`

- ◆ **Windows:** Perform the following actions:

1. Navigate to the `<Extracted Driver Patch File Temporary Location>\windows` folder and copy the `FanoutDriverShim.jar` file to `<IdentityManager installation>\NDS\lib` folder.
2. Copy the `activemq-all-5.14.3.jar` from `<Extracted Driver Patch File Temporary Location>\windows\FanoutAgent\lib` folder to `<C:\NetIQ\IdentityManager\NDS\lib` folder.

3 Start the Identity Vault and the Fanout driver instance.

6 Securing Fanout Driver Communication

NetIQ recommends using Secure Socket Layer (SSL) protocols for driver communication. By default, the SSL protocol is not configured among the Fanout components. You must configure the SSL connection among the following Fanout components:

- ◆ Between the Fanout driver shim and ActiveMQ

Refer to the following sections for instructions:

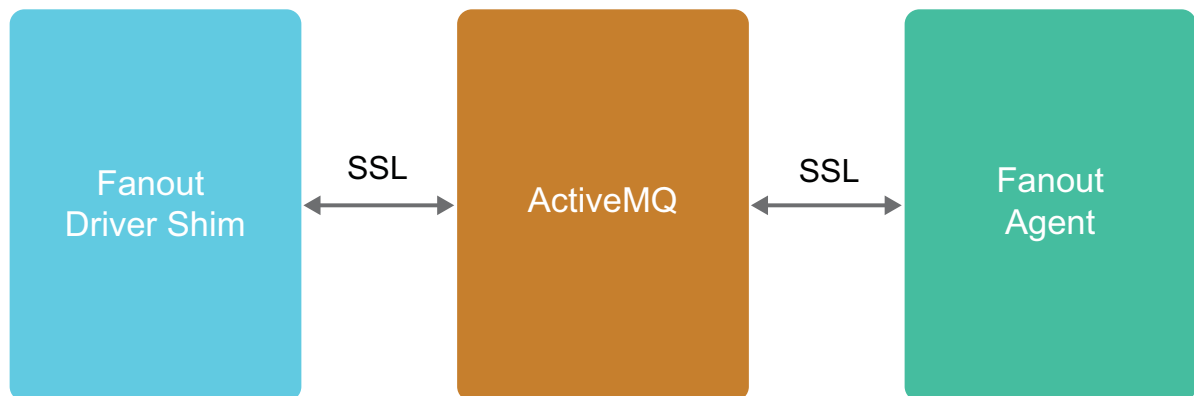
- ◆ “Creating a Keystore and a Truststore” on page 47
- ◆ “Enabling SSL for the Fanout Driver Shim” on page 48
- ◆ “Enabling SSL for ActiveMQ” on page 48

- ◆ Between ActiveMQ and the Fanout agent

Refer to the following sections for instructions:

- ◆ “Creating a Keystore and a Truststore” on page 47
- ◆ “Enabling SSL for ActiveMQ” on page 48
- ◆ “Enabling SSL for the Fanout Agent” on page 49

Figure 6-1 Secured Connection among Fanout Components of the JDBC Fanout Driver



To support the SSL connections, you need to create keystore and truststore files. This section explains how to create, export, and store this certificate on your server

After the secured connection is enabled, the Fanout components perform an SSL handshake to establish a secure channel.

Creating a Keystore and a Truststore

- 1 Create a certificate for the broker by using the keytool.

```
keytool -genkey -alias broker -keyalg RSA -keystore broker.ks
```

- 2 Export the broker's certificate to share with clients.

```
keytool -export -alias broker -keystore broker.ks -file broker_cert
```

- 3 Create a certificate/keystore for the client.

```
keytool -genkey -alias client -keyalg RSA -keystore client.ks
```

- 4 Create a truststore for the client and import the broker's certificate. This establishes that the client "trusts" the broker.

```
keytool -import -alias broker -keystore client.ts -file broker_cert
```

- 5 Export the client's certificate so it can be shared with broker:

```
keytool -export -alias client -keystore client.ks -file client_cert
```

- 6 Create a truststore for the broker, and import the client's certificate. This establishes that the broker "trusts" the client:

```
keytool -import -alias client -keystore broker.ts -file client_cert
```

NOTE: You must use the same passwords that were used for creating the keystores to configure the Fanout components for SSL. For more information about creating certificates, see [Setting up the Key and Trust Stores](#).

Enabling SSL for ActiveMQ

- 1 Copy the files `broker.ks` and `broker.ts` to the server where ActiveMQ is installed.
- 2 Edit the `activemq.xml` file.

- 2a Navigate to the `<AMQ install path>/conf/` folder and open the `activemq.xml` file.

- 2b Add the transport connector to enable SSL in the `transportConnectors` section.

```
<transportConnector name="ssl" uri="ssl://0.0.0.0:61617?trace=true"/>
```

- 2c Add the keystore or truststore paths and their passwords in the `sslContext` section.

```
<sslContext> <sslContext keyStore="file:///root/activemqkeystore/  
broker.ks" keyStorePassword="novell" trustStore="file:///root/  
activemqkeystore/broker.ts" trustStorePassword="novell"/> </sslContext>
```

- 3 Restart ActiveMQ.

NOTE: For more information about enabling secure transport on ActiveMQ, see [Using Spring to configure SSL for a Broker instance](#).

Enabling SSL for the Fanout Driver Shim

- 1 Copy the `client.ks` and `client.ts` files to the Identity Manager server.
- 2 Edit the following driver properties using Designer:
 - ♦ **AMQ Keystore Path:** Specify the `client.ks` path.
 - ♦ **AMQ Keystore Password:** Specify the password for `client.ks`.
 - ♦ **AMQ Truststore Path:** Specify the `client.ts` path.
 - ♦ **AMQ Truststore Password:** Specify the password for `client.ts`.
- 3 Save the configuration changes.
- 4 Deploy the driver.

Enabling SSL for the Fanout Agent

- 1 Copy the `client.ks` and `client.ts` files to the server where you install the Fanout agent.
- 2 Open the config file and specify the parameters as follows:
 - ♦ **netiq.fanoutagent.connection.truststore.file:** Specify the `client.ts` path. For example, `netiq.fanoutagent.connection.truststore.file=/home/keys/client.ts`.
 - ♦ **netiq.fanoutagent.connection.keystore.file:** Specify the `client.ks` path. For example, `netiq.fanoutagent.connection.keystore.file=/home/keys/client.ks`.
- 3 Start the Fanout Agent in **Server Only Mode** where it accepts only the configuration changes.
- 4 Set the passwords as follows:
 - ♦ `setAMQKSPassword -config <fanoutaget_installdir>/Fanoutagent/config/fanoutagent-config.properties <keystore password> <agent password>`
 - ♦ `setAMQTSPassword -config <fanoutaget_installdir>/Fanoutagent/config/fanoutagent-config.properties <truststore password> <agent password>`
- 5 Stop and start the Fanout Agent.

7 Troubleshooting the Driver

This section provides information about the issues that you might encounter while working with the driver.

Mismatch in Driver Version

iManager and the driver trace display different versions of the Fanout driver. This occurs because they fetch the version documents from different sources - the Fanout driver and the JDBC driver. For example, iManager's Version Discovery shows the version of the Fanout driver, but the trace shows both versions for both the drivers.

This is because some documents are directly returned from JDBC driver whereas some are constructed by the Fanout driver.

Fanout Driver and the Fanout Agent Time Out When ActiveMQ and Fanout Agent are Running

When the Fanout driver does not receive response within a specified time, it displays a time out warning message and processes the subsequent event. The response for the previous event will be processed when it is available.

It is safe to ignore the warning as this does not cause any functionality loss.

Issue with Broadcast Queries

When you issue a dxcmd query without a specific instance to the driver, the driver sends the query to all the instances. This is not the case if the driver is configured to run with the Subscriber Service channel, where the queries are routed through this channel.

ActiveMQ May Display Exception Error

ActiveMQ may display an exception error while it is viewing a queue with an unprocessed event. This issue is randomly observed.

To work around this issue, install the latest patch from the [ActiveMQ web site](#).

Changing ActiveMQ Log Levels

You can view information about an event in transit or waiting state in the ActiveMQ Web console. By default, ActiveMQ displays log messages with a verbosity of INFO and WARN. You can change the log levels in ActiveMQ in the `<Apache ActiveMQ Installation Location>/conf/log4j.properties` file.

For example, to see log messages for more verbose levels such as DEBUG, add the following line to the `log4j.properties` file:

```
log4j.logger.org.apache.activemq=DEBUG
```

Cannot Run Two Instances of the Fanout Agent Using the Same Queue Names

For multiple Fanout agents, you require equal number of ActiveMQs. In case the same ActiveMQ is used for multiple Fanout agents, manually clean the ActiveMQ queues before using a different Fanout agent.

Unable to Update the Parameters Using the setConfig Command

In order to change the `netiq.fanoutagent.trace.level`, `netiq.fanoutagent.cmd.srv.ip`, `netiq.fanoutagent.cmd.srv.port`, and `netiq.fanoutagent.cmd.allow.http` parameters, stop the Fanout agent, change the parameters in the configuration file, and restart the agent.

Stopping the Fanout Agent Fails When Any Instance Does Not Respond

When any instance fails to respond then, the Fanout agent must be forced to stop. This action will not result in any event loss.

Cleaning a State File for a Connected System

The State Directory specifies the location of the state data stored by the driver instance. The state data might be used to store additional state information in the future. The driver constructs the state files using the object GUID of the connected instance. It begins with `jdbc_` and ends with the GUID of the driver instance. Each driver instance has one state file with a unique file format: `jdbc_<driver instance guid>_1`. For example, `jdbc_bd2a3dd5-d571-4171-a195-28869577b87e_1`.

To clean a state file for a driver instance, remove the corresponding state file from the State Directory.

Adding a Group To a Specific Instance

To add a group to a specific instance perform the following actions:

1. You can either map an existing attribute or you can extend the schema to add new attribute for reading `connection-dns`.

NOTE: The policy is written assuming that the created attribute name as `conn-dn`. You need to modify the policy as per the created attribute name.

2. Click **Filter** and add the created or existing attribute from the list of attributes.
 - a. Click **OK**.
 - b. Select the created or existing attribute and the **Subscriber** option as **Notify**.
3. Add the policy under **Event Transformation Policies**:

```
<?xml version="1.0" encoding="UTF-8"?><policy>
  <rule>
    <description>Check for GroupAdd With ConnDn</description>
    <conditions>
      <and>
        <if-class-name mode="nocase" op="equal">Group</if-class-name>
        <if-operation mode="nocase" op="equal">add</if-operation>
        <if-op-attr name="conn-dn" op="available"/>
      </and>
    </conditions>
    <actions>
      <do-set-local-variable name="conn-dn" scope="policy">
        <arg-node-set>
          <token-xpath expression="add-attr[@attr-name='conn-dn']/
value"/>
          </arg-node-set>
        </do-set-local-variable>
        <do-strip-op-attr name="conn-dn"/>
        <do-for-each>
          <arg-node-set>
            <token-local-variable name="conn-dn"/>
          </arg-node-set>
          <arg-actions>
            <do-set-local-variable name="addDoc" scope="policy">
              <arg-node-set>
                <token-xml-parse>
                  <token-text xml:space="preserve">&lt;add>&lt;/
add></token-text>
                  </token-xml-parse>
                </arg-node-set>
                </do-set-local-variable>
                <do-clone-xpath dest-expression="$addDoc/add" src-
expression="@*" />
                <do-set-xml-attr expression="$addDoc/add"
name="connection-dn">
                  <arg-string>
                    <token-xpath expression="$current-node/
text()" />
                    </arg-string>
                  </do-set-xml-attr>
                  <do-clone-xpath dest-expression=".." src-
expression="$addDoc/node()" />
                </arg-actions>
              </do-for-each>
              <do-strip-xpath expression="."/>
            </actions>
          </rule>
        </policy>
```

4. Restart the driver.

5. Add a group using the `ldapadd` command.

The following is a sample `ldif` file for group add:

```
dn: cn=group_sync_inst,ou=groups,o=data
objectClass: groupOfNames
conn-dn: instance1
conn-dn: instance2
description: group is a collection of users
```

NOTE: By using this `ldif` file, the group (`group_sync_inst`) is synchronised with the instances (`instance1` and `instance2`).

Driver Does Not Start

The driver might not start if one of the following conditions exist in your environment:

- ♦ Handshake fails when the Fanout agent and the shim password are not same in the Fanout agent and the Fanout driver.
- ♦ ActiveMQ's TCP or SSL ports are blocked for the Fanout agent and the JDBC driver.

Driver Instance Does Not Start

The driver instance might not start if one of the following conditions exists in your environment:

- ♦ The JDBC driver instance object is available but the corresponding named password is not created. Make sure that you are using any of the following order to create a named password and instance object:
- ♦ The third-party JDBC drivers are not included in the Fanout agent.

You must create named password and instance object in one of the following ways:

- ♦ Create a named password and add the instance object.
- ♦ Add the instance object in disabled state, add the named password, and then enable the instance object.

Fanout Agent Stop Processing Events

When a JDBC driver instance is not responding because of an irresponsive database, the events for the database are not processed until the database returns to a normal state.

No Connection from the Fanout Agent to ActiveMQ

The Fanout agent is not able to establish a connection with ActiveMQ when one of the following conditions exist:

- ♦ ActiveMQ instance's IP address or port are not correctly configured in the Fanout agent or the JDBC driver configuration.
- ♦ The previous session of the Fanout agent or the JDBC driver is not properly stopped.

Fanout Agent Does Not Send Events

There are no events from the Fanout agent in one of the following conditions:

- ♦ The Fanout Agent and the JDBC driver are not pointing to the same ActiveMQ instance, the connection is not established.
- ♦ If the events are not available in the Fanout agent.

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 Designer for Identity Manager Administration Guide*.

Manually Deleting the Unused ActiveMQ Queues

When an instance is stopped in the Fanout Agent, the Fanout Agent does not automatically delete the corresponding instance queue in ActiveMQ. This is because if an instance queue contains an unprocessed event, deleting the queue results in loss of that event. However, you can manually delete an unused queue of a stopped instance.

- ♦ Log in as administrator into Apache ActiveMQ.
- ♦ Click **Manage ActiveMQ broker**, then click **Queues**.
- ♦ Click **Delete** in the operations column for the queue that you want to delete.

Collecting Trace Files For Troubleshooting

While troubleshooting any issues, you require to view the required trace files to analyze the issue. Ensure the following trace files are collected to help you analyze any issue:

1. **Fanout driver trace:** This is a common trace file that can be configured using Designer or iManager.
 - ♦ To configure using Designer, go to **Driver Properties > Trace** and set the trace level and trace file name.
 - ♦ To configure using iManager, click the driver icon, select **Edit Properties > Misc** and configure the trace level and trace file path.
2. **Fanout agent trace:** This trace file contains the processing logs performed by the Fanout agent. The trace file is located at `<fanout agent root directory>\logs`.

To configure the Fanout agent trace level and the trace path, modify the following parameter in the Fanout agent configuration file:

NOTE: By default, trace level is set to 5 and file name is `fanout_trace.txt`.

- ♦ `netiq.fanoutagent.trace.level=5`
- ♦ `netiq.fanoutagent.trace.file=fanout_trace.txt`

3. **JDBC driver instance trace:** This trace contains the JDBC instance level tracing, that can be configured using Designer or iManager. These trace files are located in the server where the Fanout agent is installed.

- ◆ To configure using Designer, provide the trace level and trace file path for the particular JDBC instance, while configuring the connection object.
- ◆ To configure using iManager, navigate to the driver object and open the connection instance object. Update the following xml attributes in the connection instance object xml:

```
<trace-info>  
    <driverTraceFile>trace-file-path</driverTraceFile>  
    <driverTraceLevel>trace-level</driverTraceLevel>  
    <driverTraceFileSize>trace-file-size</driverTraceFileSize>  
</trace-info>
```


8 Managing the Driver

As you work with the JDBC 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 Driver Administration Guide](#).

A

Known Issues and Limitations

NetIQ Corporation strives to ensure that our products provide quality solutions for your enterprise software needs. The following issues are currently being researched. If you need further assistance with any issue, please contact [Technical Support](#).

For the list of the known issues in Identity Manager, see the Release Notes on the [Identity Manager Documentation page \(https://www.netiq.com/documentation/identity-manager-47/\)](https://www.netiq.com/documentation/identity-manager-47/).

Known Issues

- ◆ In the MSSQL database the underline base table can be modified through a view only if the INSERT, UPDATE and DELETE statements refers to the column of the single base table.
- ◆ If you are connecting the driver to Microsoft SQL database, the driver does not support Windows NTLM and Windows Single Sign-On authentication.
- ◆ For modifications of the referential attributes (for example: Manager), the JDBC driver returns a success status even if the referenced object or user does not have the association (not present in the target database).
- ◆ Due to the above mentioned reason, if the Out of Band sync is enabled for the referential attribute, there is a possibility of event loss if the modify event of the referential attribute is processed before the referenced user's add event was processed. Hence, NetIQ recommends not to enable Out of Band events for referential attributes.
- ◆ The Fanout driver configuration page of iManager displays the Remote Loader option. Ensure that you do not use this option with the Fanout driver.
- ◆ The Fanout driver does not support direct synchronization mode for MySQL InnoDB. This is because the driver does not support the Subscriber Add operation in this mode.
- ◆ If you use the Tab key to navigate the Password field in the Fanout Configuration page of Designer, Designer prompts you to save the resource when no change is made to the resource.
- ◆ Identity Manager does not support the non-root installation of the Fanout agent.
- ◆ When you stop the Fanout agent, the command server log file displays a warning message stating that one thread cannot be stopped. Ignore the warning message.
- ◆ If you are connecting to a Sybase database, additional operations cannot be performed if the transaction log is full. The JDBC driver instance waits for the transaction log to be cleared before processing further events.
- ◆ JDBC drivers stop working if a wrong password is specified when a driver is authenticating with a connected system. This is an expected behavior. However, the JDBC driver for Sybase continues to retry the connection and does not stop.
- ◆ Changes in Fanout instance names do not reflect in the DirXML-Accounts and DirXML-Associations attributes. Therefore, they become invalid. NetIQ does not recommend you to change the connection object names and the instance names.

- ◆ Sometimes when a codemap refresh is done, you may not receive codemap data of all the connected databases. You may get errors such as “Unable to complete the CODE MAP refresh for entitlement” in catalina.out. This is due to the time taken by the query to return the codemap results. Hence, you need to increase the timeout for the following parameters:
 - ◆ **Default Query Timeout (IDMProv web UI > Roles and Resources > Configure Roles and Resource Settings > Entitlement Query Settings > Default Query Timeout).**
If the value of this parameter is 10 minutes, then increase the value of NCPCLIENT_REQ_TIMEOUT parameter.
 - ◆ **NCPCLIENT_REQ_TIMEOUT**
Refer to [Knowledgebase](#) for more information on how to increase the value of this parameter.
- ◆ If you upgraded to Identity Manager 4.7 and updated the base packages for your driver, the package update process does not overwrite the default setting (`False`) of **Enable Service Channel ECV**.
To workaround this issue, manually change the ECV for the driver.
This issue does not occur when you create a new driver.
To change the ECV in Designer:
 - 1 In Modeler, right-click the driver line.
 - 2 Select **Properties > Engine Control Values**.
 - 3 Click the tooltip icon to the right of **Engine Controls for Server**.
If a server is associated with the Identity Vault, and if you are authenticated, the engine control values display in the large pane.
 - 4 Change the value for **Enable Subscriber Service Channel**.
 - 5 Click **OK**.
 - 6 For the change to take effect, deploy the driver to the live Identity Vault.

To change the ECV in iManager:

- 1 Log in to the instance of iManager that manages your Identity Vault.
- 2 In the navigation frame, select **Identity Manager**.
- 3 Select **Identity Manager Overview**.
- 4 Use the search page to display the Identity Manager Overview for the driver set that contains your driver.
- 5 Click the round status indicator in the upper right corner of the driver icon.
- 6 Select **Edit Properties > Engine Control Values**.
- 7 Change the value for **Enable Subscriber Service Channel**.
- 8 Click **OK**, then click **Apply**.
- 9 For the change to take effect, restart the driver.

Limitations

- ◆ The Managed System Entitlement and Account Summary report displays inaccurate data for the following two components:
 - ◆ Number of Logical Systems
 - ◆ Account Entitlement Number of Assigned Accounts

For example, if there are 'n' number of Logical Systems, the report displays 'n+1' number of Logical Systems. If there are 'n' Account tracking identifiers and 'm' Accounts, the report displays 'm*n' Accounts.

- ♦ The driver displays `User is unassociated` exception while performing modify and delete operations with only `dest-dn` because these operations are not supported without providing user association.


B Driver Properties

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

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

Driver Configuration

In iManager:

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

By default, the Driver Configuration page displays.

In Designer:

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

The Driver Configuration options are divided into the following sections:

Driver Module

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

Java: Use this option 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. Select this option to run the driver locally.

The Java class name is: `com.netiq.idm.driver.fanoutshim.FanoutDriverShim`.

Native: This option is not used with the JDBC Fanout driver.

Connect to Remote Loader: This option is not valid for this driver.

Name: Displays the java class name.

IMPORTANT: Although **Driver Object Password** option is editable, this parameter is not applicable for the Fanout driver.

Authentication

The authentication section describes the parameters required for authentication to the connected Active MQ.

Connection Information (Designer only): Specify the IP address or name of the server the application shim should communicate with. Use the syntax: `protocol://host:port`. For example, `tcp://192.99.162.46:61616`

Driver Cache Limit (kilobytes): Specify the maximum event cache file size (in KB). If it is set to zero, the file size is unlimited. select **Unlimited** option to set the file size to unlimited in Designer.

IMPORTANT: Although **Application Authentication ID** and **Set Password** options are editable, these parameters are not applicable for the Fanout driver.

The Remote Loader options do not apply to the Fanout driver. This driver uses the Fanout agent component to create multiple JDBC Fanout driver instances.

Startup Option

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

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

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

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

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 presented by category:

Driver Settings

Fanout transport related parameters: Select **Show** to view the transport related parameters for Fanout driver.

Show Subscriber Event Queue parameters: Select **Show** to view the Subscriber event parameters. The options are:

- ◆ **SEND:** The driver uses this queue for sending the Subscriber events to the Fanout agent.

- ♦ **RECV:** The driver uses this queue for receiving the Subscriber events from the Fanout agent.
- ♦ **DELAYED RECV:** This driver uses this queue for receiving the delayed Subscriber events from the Fanout agent.

Show Configuration Queue Parameters: Select **Show** to view the configuration queue parameters. The options are SEND and RECV.

Show Query-in Queue Parameters: Select **Show** to view the query-in queue parameters. The options are SEND and RECV.

Show Query-out Queue Parameters: Select **Show** to view the query-out queue parameters. The options are SEND and RECV.

The Fanout agent creates queues for each connection instance. It uses these queues to send events to individual connection instances. These queues act as event caches (at Fanout agent level) for the connection instances.

Show Other Parameters: Select **Show** to view the additional parameters.

- ♦ **Configuration batch size:** Specifies the batch size for the driver configuration document. The value ranges from 1 - 99999.
- ♦ **Show Fanout Parameters:** Select **Show** to view the Fanout connection related information such as Fanout agent password, configuration information, Fanout agent shim password.
- ♦ **Fanout Shim Password:** Specifies the password for the Fanout driver shim. After successful authentication, the Fanout agent loads/creates the driver instances of the specified shim class name.
- ♦ **Fanout Agent Password:** Specifies the password for the Fanout agent you are connecting to. The Fanout agent establishes connection only after a valid authentication.
- ♦ **Encryption Key:** Specifies the key to encrypt/decrypt the sensitive data before sending to the message queue(s).
- ♦ **AMQ Keystore Key:** Specifies the full path to the keystore file.
- ♦ **AMQ Keystore Password:** Specifies the keystore password.
- ♦ **AMQ Truststore Path for SSL Certs:** Specifies the full path to the truststore file.
- ♦ **AMQ Truststore Password:** Specifies the truststore password.
- ♦ **Secure Protocol Version:** Specifies the version of the TLS protocol that the driver uses to connect to the Identity Manager engine. Identity Manager supports TLSv1, TLSv1_1, and TLSv1_2.

NOTE: This parameter is included in Fanout driver 1.1.

- ♦ **Enforce Suite B:** Specifies whether the driver uses Suite B for establishing communication with ActiveMQ. When Suite B is enabled, the communication is authenticated with Suite B cryptographic algorithms. This communication is supported only on TLS 1.2 protocol.

NOTE: This parameter is included in Fanout driver 1.1.

- ♦ **Fanout Shim classname:** Specifies the driver shim classname that the Fanout agent loads when you start the a Fanout driver.
- ♦ **Matching Attributes:** Used by the Fanout agent to match the objects in the delayed add events. This parameter must be schema-mapped equivalent of the attributes that are used in the object matching policy. If you are using different attributes, specify the attribute names according to the connected system schema.

Normal JDBC Driver Settings

For the normal JDBC driver setting, see [Driver Parameters](#) from the [JDBC Driver Guide](#).

Subscriber Settings

Disable Subscriber: Select no (default) to allow flow of events from Identity Manager engine to the connected database.

Show primary key parameters: Select **Show** if you want to configure the primary key parameters.

- ◆ **Generation/retrieval method (table-global):** Select the desired option to generate/retrieve the primary key values. This setting is global for all tables and views. The options are as follows:
 - ◆ subscription event (default)
 - ◆ subscriber-generated
 - ◆ auto-generated / identity column
- ◆ **Retrieval timing (table-global):** Select the desired option to retrieve the primary key value. This setting is global for all tables and views. The options are:
 - ◆ before row insertion (default)
 - ◆ after row insertion
- ◆ **Method and timing (table-global):** Specify how and when the primary key values are generated or retrieved on a per table or view basis. This parameter overrides global method and timing settings. Use semicolon, comma, or space as the delimiter for multiple values. For example: `usr("?=indirect.proc_idu()"); grp("indirect.proc_idg(idg)").`

Disable statement-level locking: Select the appropriate option to disable statement locking. This option determines if explicit locking or database resources are disabled on the Subscriber channel. The default value is set to no.

Check update counts: Select yes to enable the Subscriber channel to check for any updates after any of the insert, update, or delete statements are executed against the tables. This option ensures that the statements are resulting in updating the database. The default value is set to yes.

Query TimeOut (in minutes): Specify the time in minutes that the driver waits for a response from the Fanout agent when the driver issues a query to the agent. The default value is 1 minute.

ECMA Script

Displays an ordered list of ECMAScript resource files. The files contain extension functions for the driver that Identity Manager loads when the driver starts. You can add additional files, remove existing files, or change the order the files are executed.

Global Configuration


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

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 Fanout driver includes several predefined GCVs. You can also add your own if you discover you need additional ones as you implement policies in the driver.

To access the driver's GCVs in iManager:

- 1 Click  to display the Identity Manager Administration page.
- 2 Open the driver set that contains the driver whose properties you want to edit:
 - 2a In the **Administration** list, click **Identity Manager Overview**.
 - 2b If the driver set is not listed on the **Driver Sets** tab, use the **Search In** field to search for and display the driver set.
 - 2c Click the driver set to open the Driver Set Overview page.
- 3 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 organized as follows:

- ♦ [“Global Configuration Values” on page 67](#)
- ♦ [“Managed System Information” on page 68](#)
- ♦ [“Entitlements” on page 69](#)
- ♦ [“Account Tracking” on page 71](#)
- ♦ [“Password Synchronization” on page 71](#)
- ♦ [“JDBC Fanout Common” on page 72](#)

Global Configuration Values

The following global configuration values are used for database options and base configuration options.

JDBC connection URL format used: Specify the connection URL format used for the JDBC driver to connect to the databases. Use '<HOST>', '<PORT>' and '<DB>' tokens to specify the location of host's IP address, port and database/SID in the connection URL.

NOTE

- ♦ The tokens are case-sensitive and angle-brackets are mandatory since they are used as delimiters.

If you use the same Fanout driver to connect oracle pluggable database and oracle traditional database, the url template of the databases should be separated using a comma. For example:
`jdbc:oracle:thin:@<HOST>:<PORT>/<DB>, jdbc:oracle:thin:@<HOST>:<PORT>:<DB>`

Synchronization model: Select the synchronization model. The synchronization options are: Direct and Indirect. Direct synchronization uses views to synchronize directly to existing tables of arbitrary structure. Indirect synchronization synchronizes to intermediate staging tables with a particular structure.

UserName Column: Specify the exact column name of the `usr` table that store the usernames.

Managed System Information

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

General Information

Name: Specify a descriptive name for the managed system.

Description: Specify a brief description of the managed system.

Location: Specify the physical location of the managed system.

Vendor: Specify the vendor of the managed system.

Version: Specify the version of the managed system.

System Ownership

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

System Classification

Classification: Select the classification of the connected application. 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 connected application.

Environment: Select the type of environment the connected application provides. The options are:

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

If you select **Other**, you must specify a custom classification for the connected application.

Connection and Miscellaneous Information

Connection and miscellaneous information: This set of 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.

JDBC Fanout Instances Information

These settings help to configure the Managed System Service related details of each JDBC FanOut instance. To create a new instance, click the plus sign and fill in the following information:

- ◆ **JDBC FanOut Instance Name:** Specify the descriptive name of the new logical instance of the managed system.
- ◆ **Show other configuration values:** Select **Show** to display additional information related to the FanOut instance. For more information, see [“Managed System Information” on page 68](#).
- ◆ **Connection and miscellaneous information:** Select **Show** to display the system options. The options are:
 - ◆ Instance ID
 - ◆ Authentication IP Address
 - ◆ Authentication Port
 - ◆ Authentication ID
 - ◆ Database Schema
 - ◆ Type

NOTE: The connection information options are auto-generated and always set to **hide**.

Entitlements

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

Entitlements

Account Entitlement Value: Specify the entitlement value to assign for user account during the account creation. Role Based Provisioning Module displays this value to the user during account provisioning.

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

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

Allow Login Disabled in Subscriber Channel: Select **True** to enable the driver to control the flow of **Login Disabled** attribute in the Subscriber Channel and only on a regular attribute change.

Advanced Settings: Entitlement options that allow or deny additional functionality like data collection, role mapping, resource mapping, parameter format, and entitlement extensions. Leave these settings as default.

Data Collection

Data collection enables Identity Reporting to gather information to generate reports. For more information, see the [NetIQ Identity Reporting: User's Guide to Running Reports](#).

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 for the user accounts.

Allow data collection from groups: Select **Yes** to allow data collection by the Data Collection Service for groups.

Role Mapping

The Identity Applications role administration functionality allows you to map business roles with IT roles. For more information, see [Identity Applications Administration](#) in the [NetIQ Identity Manager - Administrator's Guide to the Identity Applications](#).

Enable role mapping: Select **Yes** to make this driver visible to Identity Applications.

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

Allow mapping of groups: Select **Yes** if you want to allow mapping of groups in Identity Applications.

Resource Mapping

Identity Applications allow 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 Identity Applications.

Allow mapping of user accounts: Select **Yes** if you want to allow mapping of user accounts in Identity Applications. 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 Identity Applications.

Parameter Format

Format for Account entitlement: Specify the parameter format the entitlement agent must use when granting the user account entitlement. The options are **Identity Manager 4** and **Legacy**.

Format for Group entitlement: Specify the parameter format the entitlement agent must use when granting the group entitlement. The options are **Identity Manager 4** and **Legacy**.

Entitlements Extensions

User account extension: Specify the user account extension. The content of this field is added below the entitlement elements in the EntitlementConfiguration resource object

Group extensions: Specify the group extensions. The content of this field is added below the entitlement elements in the EntitlementConfiguration resource object

Account Tracking

The following controls the Account tracking is part of Identity Reporting. For more information, see the [NetIQ Identity Reporting: User's Guide to Running Reports](#).

Enable Account Tracking: Set this to **True** to enable account tracking policies for the Fanout driver. Set it to **False** if you do not want to execute account tracking policies.

- ◆ Object class
- ◆ Realm
- ◆ Identifiers for Account
- ◆ Status Attribute
- ◆ Status active value
- ◆ Status inactive value
- ◆ Subscription default status
- ◆ Publication default status

Password Synchronization

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

Application Accepts Passwords from Identity Manager: If this option is set to **True**, the driver allows passwords to flow from the Identity Manager data store to the connected server.

Identity Manager Accepts Passwords from the Application: If this option is set to **True**, it 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 the Identity Vault.


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 passwords policy validation before publishing passwords: Select **True** to apply NMAS password policies when publishing passwords. 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 this option is set to **True**, and the Distribution Password fails to distribute, 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 this option is set to **True**, notify the user by e-mail of any password synchronization failures.

Connected System or Driver Name: Specifies the name of the connected system, application or Identity Manager driver. This value is used by the e-mail notification templates to identify the source of notification messages.

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.

JDBC Fanout Common

Allow 'Group add' in Fanout mode: This GCV controls the creation of groups in the Subscriber channel. By default, this is **Disabled**. The driver vetoes the group add operations. Enabling this option allows the driver to send the group add events to each of the JDBC instances configured by the driver.

Synchronize the first or the last replica value: Select the appropriate option to synchronize the first or last replica value of multi-valued attributes mapped to single-valued columns. The options are: First and Last.

C

Setting Up Trace Levels

The driver supports the following trace levels:

Level	Description
0	Status messages (success/failure/warning) from the engine. No trace messages are displayed or logged by the Fanout driver.
1	Basic trace messages are displayed and logged. Higher trace levels provide more detail.
3	Level 1 messages and information about the XML documents that are exchanged by the Fanout driver and the Fanout agent.
5	Trace Level 3 messages plus debugging messages.

D REST Endpoints

Table D-1 on page 75 lists the endpoints that the Fanout agent exposes for performing basic monitoring and management tasks.

Table D-1 REST endpoints

Path	Method	Parameter(s)	Description
/fanoutagent/config	PUT	{“KEY”:”VALUE”,...} Key-value pair of the parameters received through the GET method call to this endpoint (refer (3))	Used to set or update the parameter values for the Fanout agent. Clients to this endpoint should provide the changed/modified value pairs only. Password updates cannot be done through this endpoint. Due to the alternate handling of sensitive data, different endpoints are available for password updates.
/fanoutagent/config	GET	{None}	Provides the current configuration values of the Fanout agent.
/fanoutagent/config/ setpassword	PUT	{“OLD_SHIM_PASSWD”:”...”,“NEW_SHIM_PASSWD”:”...”,“OLD_AGENT_PASSWD”:””,“NEW_AGENT_PASSWD”:””}All the values must be provided	Used to set the agent and the shim password on the Fanout agent. NOTE: Both the password pairs must be provided. Otherwise, the agent might be locked down that can result in an error at startup. This is because one of the passwords is used for encrypting the other and hence the dependency. There are no endpoints for fetching the password.

Path	Method	Parameter(s)	Description
/fanoutagent/config/ setkspassword	PUT	{"PASSWORD":"..."}	<p>Sets the keystore password for the command server.</p> <p>The keystore contains a self-signed certificate created by the Fanout agent at the initial startup.</p> <p>A user provided keystore with a certificate can also be used.</p> <p>The certificate is used for securing the endpoint transport. An option exists to disable the SSL, which is not the recommended/default option.</p>
/fanoutagent/service	GET	{None}	<p>Retrieves the list of connected system services (shims) in the Fanout agent. Both running and stopped shims are reported in the response of this endpoint.</p>
/fanoutagent/ service/{state}	GET	{NonePossible values for state are RUNNING or STOPPED}	<p>Provides the list of connected system services (shims) in the Fanout agent based on the service state.</p> <p>NOTE: Legal values are RUNNING and STOPPED.</p>
/fanoutagent/ service/cmd/stop/ {serviceId}	PUT	{NoneServiceId returned by the GET service endpoint}	Used to start a currently stopped service.
/fanoutagent/ service/cmd/start/ {serviceId}	PUT	{NoneServiceId returned by the GET service endpoint}	Used to start a currently stopped service.

Path	Method	Parameter(s)	Description
/fanoutagent/ shutdown	PUT	{None}	Used to shutdown a Fanout agent instance. NOTE: When this endpoint is invoked, the agent instance will shutdown. This means that all further communication with the agent will cease. Further communication will resume only after the agent is started by logging into the computer hosting the agent.
/config/ setEncryptionkey			

