



NetIQ® Identity Manager™ Driver for ID Provider 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 this Book and the Library	5
About NetIQ Corporation	7
1 Understanding the ID Provider Driver	9
Why Use the Driver?	9
Design Architecture	9
Schema Architecture	11
2 Installing Driver Files	13
3 Creating a New Driver Object	15
Creating the Driver Object in Designer	15
Importing the Current Driver Packages	15
Installing the Driver Packages	16
Configuring the Driver Settings	17
Deploying the Driver Object	19
Starting the Driver	20
Adding Packages to an Existing Driver	20
Creating ID Policies	21
Default Policies	21
Creating an ID Policy	22
Managing the Access Control List	22
Activating the Driver	23
Adding Packages to an Existing Driver	23
4 Upgrading an Existing Driver	25
What's New in Version 4.0.0	25
Upgrade Procedure	25
5 Configuring ID Clients	27
ID Client	27
Standalone Client	28
6 Managing the ID Provider Driver	29
7 Troubleshooting	31
A Driver Properties	33
Driver Configuration	33
Driver Module	34
Driver Object Password	34
Authentication	34
Startup Option	35

Driver Parameters	35
ECMAScript	36
Global Configurations.....	36
Global Configuration Values	37

About this Book and the Library

The *Identity Manager Driver for ID Provider Implementation Guide* explains the purpose of the ID Provider driver and how to implement the driver.

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/)

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 **Add Comment** 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 community.netiq.com.

1 Understanding the ID Provider Driver

The Identity Manager driver for ID Provider enables you to create and maintain a central source of unique IDs that can be consumed by client applications or systems. When the driver receives an ID request from a client, it generates an ID based on policies you define, passes it to the client, and then stores it in the Identity Vault.

- ◆ [“Why Use the Driver?” on page 9](#)
- ◆ [“Design Architecture” on page 9](#)
- ◆ [“Schema Architecture” on page 11](#)

Why Use the Driver?

There are many different reasons why you would want to use the ID Provider driver. For example:

- ◆ It is convenient for administrators to have one basic ID for each object in the system, and to have complete control of the ID. No other system can change this ID.
- ◆ You can use the ID Provider driver in conjunction with the WorkOrder driver to verify that each WorkOrder ID is unique.
- ◆ You can use the driver to help manage UIDs and GIDs in Linux.

Design Architecture

Identity Manager drivers listen for events and then apply the proper Identity Manager policies for the event. That information is then passed to the Identity Manager engine that executes the policies.

The ID Provider driver is different from all other Identity Manager drivers. It also listens for events, but it has two sets of policies: the Identity Manager policies and the ID Provider policies. The ID Provider policies allow the driver to generate and assign unique IDs to objects.

The driver has three major components:

- ◆ **ID Client:** The ID client communicates with the ID Provider driver to obtain a unique ID. The client can be another Identity Manager driver (for example, the WorkOrder driver) or a standalone Java application.
- ◆ **ID Provider Driver:** The driver receives ID requests from clients, generates unique IDs that are stored in the Identity Vault, and passes the unique IDs back to the client. The driver uses LDAP to access the Identity Vault and uses Java RMI (Remote Method Invocation) to communicate with ID clients.
- ◆ **Identity Vault:** The Identity Vault provides the location for storing unique IDs and also contains the policies used to generate the IDs. All IDs and policies are stored in the ID Policy Container.

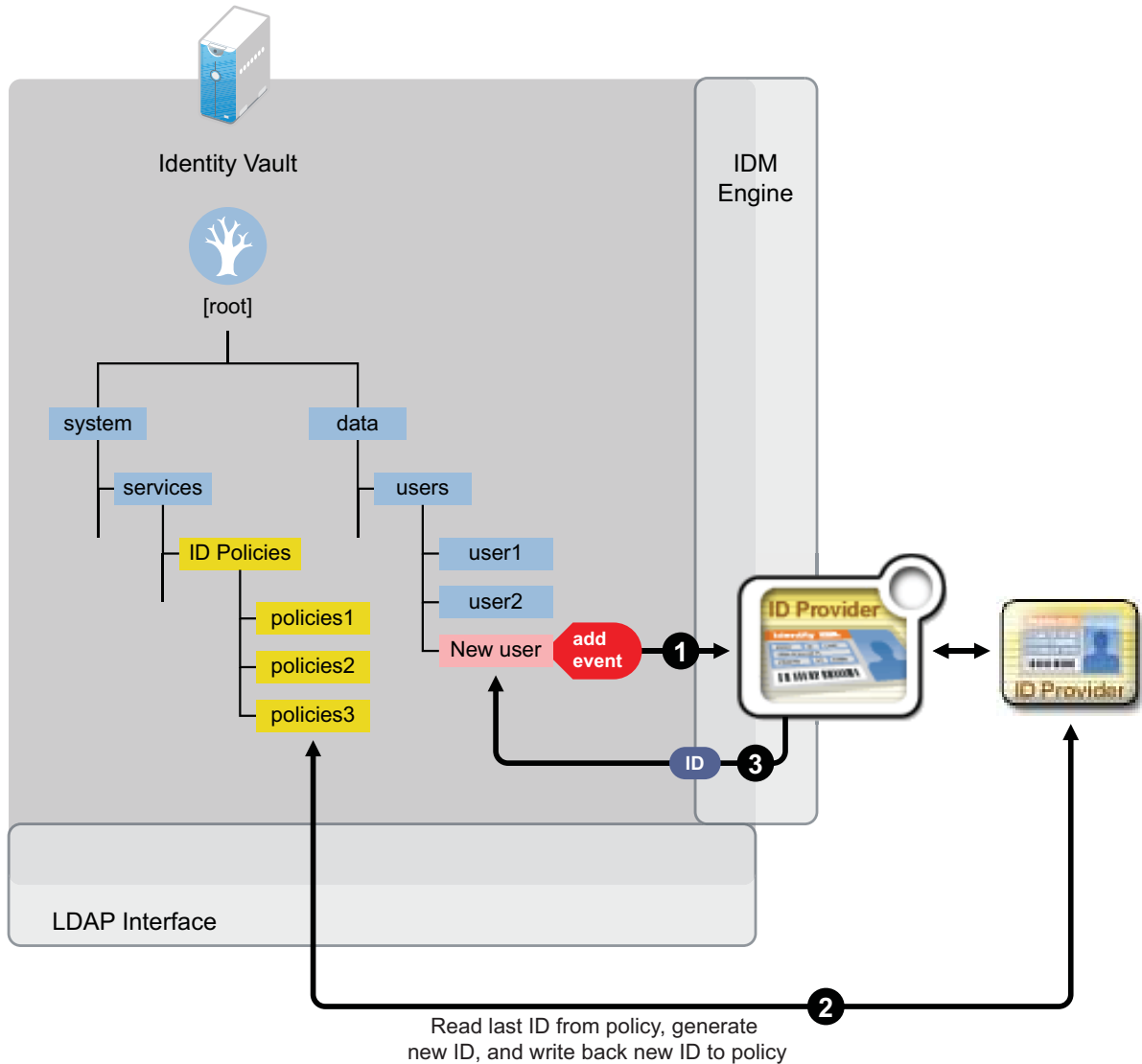
The ID Provider driver can be used in two different scenarios:

- ◆ [“Scenario 1: Using the Identity Vault to Store the ID Provider Policies” on page 10](#)
- ◆ [“Scenario 2: Using an LDAP Database to Store the ID Provider Policies” on page 11](#)

Scenario 1: Using the Identity Vault to Store the ID Provider Policies

This is the most commonly used scenario for this driver. The ID Provider policies are created and stored in the Identity Vault when the driver is created and configured. [Figure 1-1](#) shows how a unique ID is generated.

Figure 1-1 Identity Vault Stores the ID Provider Policies



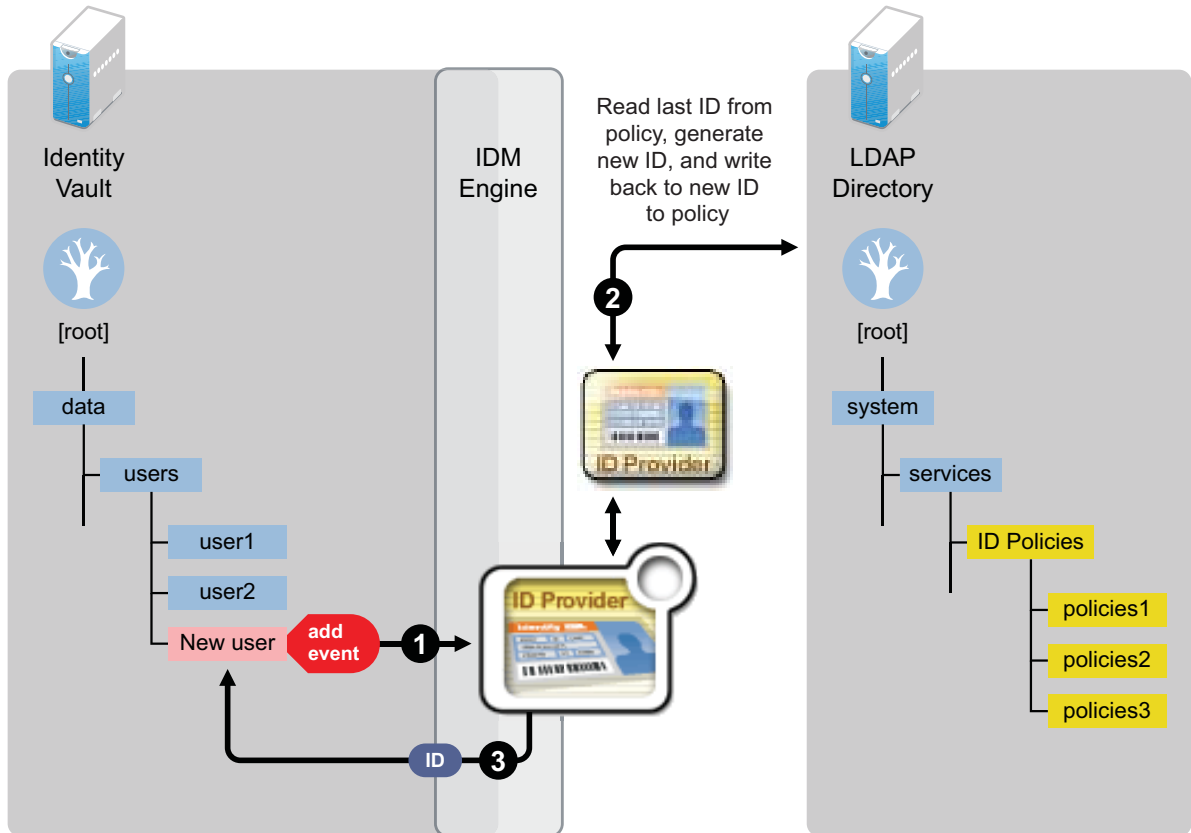
1. A new User object is created in the Identity Vault, then the ID Provider driver picks up the Create event.
2. The ID Provider driver reads the last ID that was generated from the ID Provider policies in the Identity Vault and generates a new ID. The ID is then written back to the ID Provider policies in the Identity Vault to track the unique IDs.
3. The ID Provider driver then assigns the new ID to the new User object.

All events are tracked and stored in the Identity Vault.

Scenario 2: Using an LDAP Database to Store the ID Provider Policies

This scenario allows you to use an LDAP database to store the ID Provider policies instead of using the Identity Vault. Figure 1-2 shows how a unique ID is generated with the LDAP database.

Figure 1-2 LDAP Database Stores the ID Provider Policies



1. A new User object is created in the Identity Vault, then the ID Provider driver picks up the Create event.
2. The ID Provider driver reads the last ID that was generated from the ID Provider policies in the LDAP database and generates a new ID. The ID is then written back to the ID Provider policies in the LDAP database to track the unique IDs.
3. The ID Provider driver then assigns the new ID to the new User object in the Identity Vault.

Schema Architecture

The Identity Vault's schema must be extended to support the ID Provider driver functionality. The following two tables describe the schema attributes and classes.

Table 1-1 Schema Attributes

Attribute Name	Syntax	Attribute Flags	Description
DirXML-IDPoIName	Case Ignore String	Single valued Synchronize immediately	ID Policy object name

Attribute Name	Syntax	Attribute Flags	Description
DirXML-IDPoLastID	Numeric String	Single-valued Synchronize immediately	Last delivered ID
DirXML-IDPoMin	Numeric String	Single-valued	Minimum value for an ID
DirXML-IDPoMax	Numeric String	Single-valued	Maximum value for an ID
DirXML-IDPoPrefix	Case Ignore String	Single-valued	Prefix for a new ID
DirXML-IDPoFill	Boolean	Single-valued	True: Fill ID with 0 up to maximum length False or Empty: Do nothing
DirXML-IDPoArea	Case Ignore String	Single-valued	Exclude/Include list for generated IDs
DirXML-IDPoAreaEI	Boolean	Single-valued	True: IDPoArea = Include list False or Empty: IDPoArea = Exclude list
DirXML-IDPoAccessControl	Boolean	Single-valued	True: IDPoACL list is used False or Empty: IDPoACL list is not used
DirXML-IDPoACL	Case Ignore String	Single-valued	Comma-delimited list of ID clients to be allowed to request an ID from the ID server
DirXML-IDPolicyContainerDN	Distinguished Name	Single-valued	Link to the ID Policy Container

Table 1-2 Schema Classes

Class Name	Contained By	Attributes Contained
ID Policy Container	Country, Domain, Locality, Organization, Organizational Unit, Tree Root	OU
ID Policy	ID Policy Container	IDPoACL IDPoAccessControl IDPoArea IDPoAreaEI IDPoFill IDPoLastID IDPoMax IDPoMin IDPoName IDPoPrefix

2 Installing Driver Files

The ID Provider Driver is a service driver that is included with the base Identity Manager product. The driver is installed when the Identity Manager engine and drivers are installed. For the installation instructions, see [“Understanding the Installation and Configuration Process”](#) in the *NetIQ Identity Manager Setup Guide for Linux* or [“Planning Your Installation ”](#) in the *NetIQ Identity Manager Setup Guide for Windows*.

3 Creating a New Driver Object

After the ID Provider driver files are installed on the server where you want to run the driver (see [Chapter 2, “Installing Driver Files,” on page 13](#)), you must create the driver in the Identity Vault. The following sections provide instructions:

- ♦ [“Creating the Driver Object in Designer” on page 15](#)
- ♦ [“Adding Packages to an Existing Driver” on page 20](#)
- ♦ [“Creating ID Policies” on page 21](#)
- ♦ [“Activating the Driver” on page 23](#)
- ♦ [“Adding Packages to an Existing Driver” on page 23](#)

Creating the Driver Object in Designer

You can run the driver as a native Java module or as an Identity Manager driver on any supported platform.

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

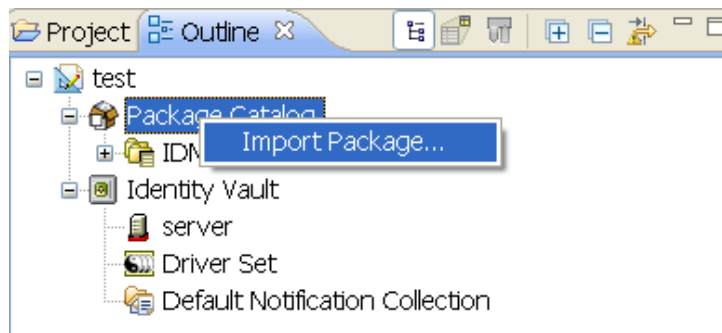
- ♦ [“Importing the Current Driver Packages” on page 15](#)
- ♦ [“Installing the Driver Packages” on page 16](#)
- ♦ [“Configuring the Driver Settings” on page 17](#)
- ♦ [“Deploying the Driver Object” on page 19](#)
- ♦ [“Starting the Driver” on page 20](#)

Importing the Current Driver Packages

The driver packages contain the items required to create a driver, such as policies, entitlements (not available in Beta), filters, and Schema Mapping policies. These packages are only available in Designer. You can upgrade any package that is installed if there is a newer version of the package available. It is recommended to have the latest packages in the Package Catalog before creating a new driver object. For more information on upgrading packages, see [“Upgrade Settings:”](#) in the *NetIQ Designer for Identity Manager Administration Guide*.

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

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



- 6 Select any ID Provider driver packages
or
Click **Select All** to import all of the packages displayed.
By default, only the base packages are displayed. Deselect **Show Base Packages Only** to display all packages.
- 7 Click **OK** to import the selected packages, then click **OK** in the successfully imported packages message.
- 8 After the current packages are imported, continue with “[Installing the Driver Packages](#)” on [page 16](#).

Installing the Driver Packages

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

- 1 In Designer, open your project.
- 2 In the Modeler, right-click the driver set where you want to create the driver, then click **New > Driver**.
- 3 Select **ID Provider Base**, then click **Next**.
- 4 On the ID Provider page, specify a name for the driver, then click **Next**.
- 5 On the ID Provider page, fill in the following fields, then click **Next**:
 - LDAP server:** Specify the IP address or DNS name of the LDAP server that contains the ID policies.
 - Policy Container DN:** Specify the LDAP DN of the policy container.
- 6 On the ID Provider page, fill in the following fields, then click **Next**:
 - Authentication ID:** Specify the LDAP DN of a user with read/write access to the ID Policy container and its child objects.
 - Authentication Password:** Specify the password of the user named in the **Authentication ID** field.
- 7 Fill in the following fields for Remote Loader information:
 - Connect To Remote Loader:** Select **Yes** or **No** to determine if the driver will use the Remote Loader. For more information, see the [Configuring the Remote Loader and Drivers](#) in the [NetIQ Identity Manager Setup Guide for Linux](#) or [Configuring the Remote Loader and Drivers](#) in the [NetIQ Identity Manager Setup Guide for Windows](#).

If you select **No**, skip to [Step 8](#). If you select **Yes**, use the following information to complete the configuration of the Remote Loader, then click **Next**:

Host Name: Specify the IP address or DNS name of the server where the Remote Loader is installed and running.

Port: Specify the port number for this driver. Each driver connects to the Remote Loader on a separate port. The default value is 8090.

Remote Loader Password: Specify a password to control access to the Remote Loader. It must be the same password that is specified as the Remote Loader password on the Remote Loader.

Driver Password: Specify a password for the driver to authenticate to the Identity Manager server. It must be the same password that is specified as the Driver Object Password on the Remote Loader.

- 8 Review the summary of tasks that will be completed to create the driver, then click **Finish**.
- 9 After the driver packages are installed, if you want to change the configuration of the Role-Based Entitlement driver, continue to [“Configuring the Driver Settings” on page 17](#).


or

If you do not want to change the configuration of the driver, continue to [“Deploying the Driver Object” on page 19](#).

Configuring the Driver Settings

After you import the driver configuration file, the ID Provider driver will run. However, there are many configuration settings that you can use to customize and optimize the driver. The settings are divided into categories such as Driver Configuration, Engine Control Values, and Global Configuration Values (GCVs). The settings are described in [Appendix A, “Driver Properties,” on page 33](#).

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

- 1 Open your project.
- 2 In the Modeler, right-click the driver icon  or the driver line, then select **Properties**.
- 3 Make the changes you want, then continue to [“Starting the Driver” on page 20](#).

If you want to make additional changes to the driver, the following sections contain information about the driver parameters.

- ♦ [“ID Policy Repository” on page 17](#)
- ♦ [“Client Options” on page 18](#)
- ♦ [“Server Options” on page 19](#)

ID Policy Repository

The ID policy repository parameters contain information about the location and how to access any ID policies.

Parameter	Default Value	Description
LDAP Server	127.0.0.1	The IP address or DNS name of the LDAP server holding the ID policies
LDAP Port	636	The TCP port that the LDAP server listens on. The value is usually 389 for non-SSL connections and 636 for SSL connections.
Use SSL	True	Specify whether or not you want to use SSL.
Always trust	True	Specify whether or not you want to trust all servers. If this option is set to True, the ID provider trusts all LDAP servers even if the server certificate is untrusted.
Policy Container DN	LDAP DN for the policy container under the driver object. For example cn=id-policies,cn=id-provider,cn=driverset1,dc=idm,dc=services,dc=system.	Specify or browse to the DN of the policy container in your tree. The policy container can only be created under the ID Provider driver.

Client Options

The client options are for the ID Provider clients. For more information, see [Chapter 5, “Configuring ID Clients,”](#) on page 27.

Parameter	Default Value	Description
Client name	ID-Provider Driver	This is the name the driver uses when it acts as an ID client and requests an ID from the provider. This is useful for tracing and if access control is enabled on any of the ID policies. If access control is enabled, a list of ID client names can be specified that are allowed to obtain an ID from the policy. If the client name associated with the request is not in that list, the provider does not issue an ID.
ID Generation Map	workforceID=wfid	Provide a comma-separated list of attribute=policy pairs. For example, workforceID=wfid,uniqueID=uid. This example configures the driver to request IDs from the wfid policy and stores them in the workforceID attribute whenever a new object is created or whenever someone tries to change this attribute. Similarly, IDs from the UID policy are used from the uid attribute. The driver only issues IDs for any attribute if that attribute and the object class holding the attribute are in both the Subscriber, Publisher, Filter, and are set to synchronize. Attribute names must be in the Identity Namespace (not LDAP) and must be case-exact.


Server Options

These options allow you to set up clients other than the ID Provider driver by using Java Remote Method Invocation (RMI). It also allows you to set ID Provider trace level.

Parameter	Default Value	Description
Start RMI	True	Controls whether the ID provider starts an RMI service or not. You only need a running RMI service if you request IDs from other clients than the driver (for example, DirXMLScript policies.) If all IDs are managed through this driver's filter and ID Generation Map settings, then no RMI service is needed.
RMI server	172.17.2.117	Specify the IP address the RMI server binds to. Leave this field empty if you want the server to bind to all IP addresses.
RMI port	1199	Specify the TCP port the RMI service listens on. The defined standard port for RMI is 1099. If that port is already in use (you see errors in the trace when you start the driver), use a different port higher than 1023. This configuration assumes a port of 1199 to avoid common port conflicts.
Use legacy ID-server schema?	False	Enables the backward compatibility mode when migrating an existing ID-Server configuration to run with the new ID Provider driver. Setting this to True allows you to keep using legacy ID policies, which do not use the new schema that ships with the ID Provider.
Trace level	ALL	This is not the driver trace level, but the ID Provider trace level. The levels are: <ul style="list-style-type: none">◆ OFF: Tracing is turned off.◆ FATAL: Displays only fatal messages.◆ ERROR: Displays only error messages.◆ WARN: Displays only warning messages.◆ INFO: Displays only informational messages.◆ DEBUG: Displays only debug messages.◆ ALL: Displays all messages.

Deploying the Driver Object

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

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

- 6 Click **Define Security Equivalence** to assign rights to the driver.

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

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

6b Click **OK** twice.

For more information about defining a Security Equivalent User in objects for drivers in the Identity Vault, see [Establishing a Security Equivalent User](#) in the *NetIQ Identity Manager Security Guide*.

- 7 Click **Exclude Administrative Roles** to exclude users that should not be synchronized.

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

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


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

- 8 Click **OK**.

Starting the Driver

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


To start the driver:

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

For information about management tasks for the driver, see [Chapter 6, "Managing the ID Provider Driver,"](#) on page 29.

Adding Packages to an Existing Driver

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

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

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

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

Package Management			
Installed Packages			
Package	Versi...	Upgra...	Operation
Password Synchronization Notificatio...	0.2.0		Select Operation...
Provisioning Notification Templates	0.2.0		Install
Password Management Notification T...	0.2.0		Install
Password Expiration Notification Tem...	0.2.0		Install
Job Default Notification Templates	0.2.0		Install

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

Package Management			
Installed Packages			
Package	Versi...	Upgra...	Operation
Job Default Notification Templates	0.2.0		Select Operation...
Password Expiration Notification Tem...	0.2.0		Select Operation...
Password Management Notification T...	0.2.0		Select Operation...
Password Synchronization Notificatio...	0.2.0		Select Operation...
Provisioning Notification Templates	0.2.0		Select Operation...

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

Creating ID Policies

An ID Policy container is a repository for ID policies and is used in conjunction with the ID Provider driver. An ID policy allows the ID Provider driver to generate unique IDs. When the ID Provider driver receives an ID request from a client, it generates an ID based on the ID policy specified in the request and passes it to the client.

- ♦ [“Default Policies” on page 21](#)
- ♦ [“Creating an ID Policy” on page 22](#)
- ♦ [“Managing the Access Control List” on page 22](#)

Default Policies

By default, there are three ID policies that are created when the driver is imported. The three policies are sample policies. You can use these policies or create your own. The default policies are:

- ♦ **pid**: The pid policy generates unique IDs between 100000 to 2000000000. It also adds the prefix of “PID” to each unique ID.

- ♦ **wfid:** The wfid policy generates unique IDs between 10000000 to 99999999. It also adds the prefix of “WFID” to each unique ID for the workforce ID.
- ♦ **woid:** The woid policy generates unique IDs between 100000 to 2000000000. It also adds the prefix of “WOID” to each unique ID.

Creating an ID Policy

To create an ID policy:

- 1 In Designer, right-click the ID Policy container in the **Outline** tab, then click **New > ID Policy**.
The ID Policy container is created when the ID Provider driver is created. The ID Policy container can only reside under the ID Provider driver.
- 2 Specify the name for the ID policy, then click **OK**.
- 3 Double-click the ID policy to access the properties page.
- 4 Use the following information to create your ID policy:

Policy Name: Specify the name of the ID policy.

Policy’s Last ID: The last ID number that was used by this ID policy. If you have deployed this ID policy, use the **Connect** icon to update this field to the last ID number that was stored in the Identity Vault for this ID policy.

NOTE: Only the ID Provider driver can update the last value stored in the Identity Vault.

Constraints Minimum/Maximum: Numbers must be between 0 and 2147483647. If you have a fixed system that can only handle eight digits, set the **Maximum** to 99999999.

Constraints Exclude/Include: Allows you to include or exclude a set of numbers that you type in. Numbers can be typed in a coma-delimited list and you can use ranges, such as 10,100,1000,5000-10000,1099, etc.

Constraints Prefix: Allows you to give a prefix to the IDs that are generated using this ID policy. If you create multiple ID policies, a prefix is useful to see which ID policies are being used. An example is WFID, for workforce IDs.

Constraints Fill Yes/No: If you choose **Yes**, the ID is filled with leading zeros (0) up to the maximum length. This helps keep generated IDs at the same length. If you select **No**, it does nothing and the ID lengths increment over time.

Access Control Enabled: Check this box if you want to enable access control list.

Access Control ACL: Type in the access control lists you want to use. Access control must be enabled before you can type in ACLs. For more information, see [“Managing the Access Control List” on page 22](#).

- 5 Click **OK** to save the information.

Managing the Access Control List

The Access Control List (ACL) is also called the Object Trustee property. Whenever you make a trustee assignment, the trustee is added as a value to the Object Trustees (ACL) property of the target.

The value for the ACL parameter must match the value that the ACL client is using.


Activating the Driver

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

If you create the driver in a driver set that has not been previously activated, the driver will run in the evaluation mode for 90 days. You must activate the driver during the evaluation period; otherwise, the driver will be disabled. If you try to run the driver, 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 Drivers](#) in the *NetIQ Identity Manager Overview and Planning Guide*.

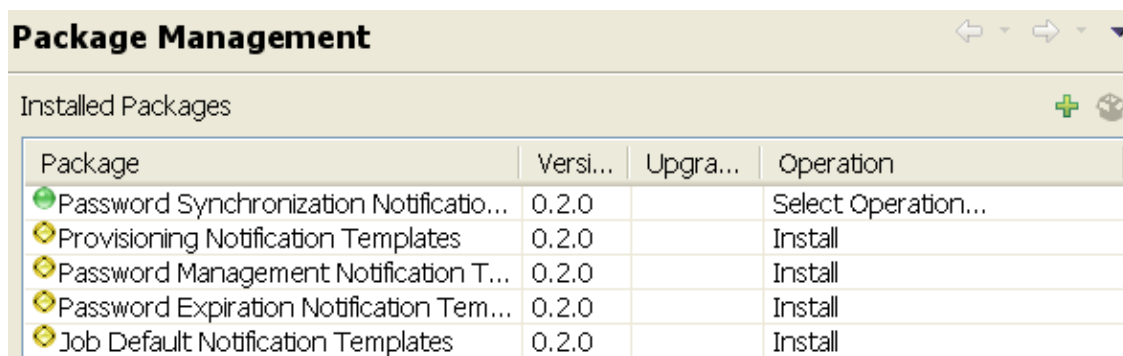
Adding Packages to an Existing Driver

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

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

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

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



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

Package Management			
Installed Packages			
Package	Versi...	Upgra...	Operation
Job Default Notification Templates	0.2.0		Select Operation...
Password Expiration Notification Tem...	0.2.0		Select Operation...
Password Management Notification T...	0.2.0		Select Operation...
Password Synchronization Notificatio...	0.2.0		Select Operation...
Provisioning Notification Templates	0.2.0		Select Operation...

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

4 Upgrading an Existing Driver

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

- ♦ “What’s New in Version 4.0.0” on page 25
- ♦ “Upgrade Procedure” on page 25

What’s New in Version 4.0.0

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

Upgrade Procedure

The driver upgrade process involves updating the driver files and upgrading the installed driver packages. However, there is no separate procedure for updating the driver shim. The driver shim is updated with a new version of Identity Manager engine.

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

- 1 Download the latest available packages.

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

- 2g Click **Apply**.

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

- 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 “Upgrading the Identity Manager Drivers” in the *NetIQ Identity Manager Setup Guide for Linux* or “Upgrading the Identity Manager Drivers” in the *NetIQ Identity Manager Setup Guide for Windows*.

5 Configuring ID Clients

An ID client can be run as a standalone Java process or included in another Identity Manager driver. All clients must use the Java RMI (Remote Method Invocation) interface to request a new ID from the ID Provider driver.

- ♦ [“ID Client” on page 27](#)
- ♦ [“Standalone Client” on page 28](#)

ID Client

The ID client can be used inside of DirXML style sheets calling the getNextID function of the com.novell.ncs.idsrv.IDClient Java class.

```
xmlns:id=http://www.novell.com/nxsl/java/com.novell.idm.idprovider.IDClient
```

To obtain the next available ID from an ID Policy object in the Identity Vault, the ID client uses the following parameters to communicate with the ID Provider driver:

Parameter	Description	Sample
\$RMIServer	RMI server host address.	localhost
\$RMIPort	RMI server port.	1099
\$UIRule	ID Policy object name to retrieve an ID from.	uniqueCN
\$IDClient	ID Client name to identify this client at the RMI server.	Client-No2
\$Tracelevel	Trace level. You use the trace level setting to see specific trace information in the DirXML ID Servers main screen. The trace level is a bit mask and can be combined. Trace values and levels: 0 = off 1 = low 2 = medium 3 = high 4 = exceptions	1

To generate the IDs by the XPATH expression, use the following command:

```
id: getNextID('IP_OF_RMI_SERVER', 'PORT', 'POLICY_NAME', 'CLIENT_NAME', TRACE_LEVEL
```

The XPATH expression will look like this:

```
<arg-string>  
  <token-xpath  
expression="id:getNextID('172.17.1.3', '1199', 'MYUID', 'DBUSERS', 0)"/>  
</arg-string>
```

Standalone Client

The standalone client is run as a Java process that calls the main function of the `com.novell.ncs.idsrv.IDClient` Java class.

```
%JRE_HOME%\java -noverify -classpath %CLASSPATH%  
com.novell.idm.idprovider.IDClient <parameters>
```

To obtain the next available ID from an ID Policy objects in the Identity Vault, the client uses the following parameters to communicate with the driver:

Parameter	Description	Sample
-h	RMI server host address.	-h localhost
-p	RMI server port.	-p 1099
-o	ID Policy object name to retrieve an ID from.	-o uniqueCN
-c	ID Client name to identify this client at the RMI server.	-c Client-No1
-t	Trace level.	-t 1
	You use the trace level setting to see specific trace information in the DirXML ID Servers main screen.	
	The trace level is a bit mask and can be combined.	
	Trace values and levels:	
	0 = off	
	1 = low	
	2 = medium	
	3 = high	
	4 = exceptions	
-m	Remote RMI server command to be executed at the RMI server console	-m reinitialize

```
%JRE_HOME%\java -noverify -classpath %CLASSPATH%  
com.novell.idm.idprovider.IDClient -h localhost -p 1099 -o Policy -t 1 -c Client -  
l 1
```

6 Managing the ID Provider Driver

As you work with the ID Provider 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
- ◆ Upgrading an existing 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](#).

7 Troubleshooting

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

A Driver Properties

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


- ♦ “[Driver Configuration](#)” on page 1
- ♦ “[Global Configuration Values](#)” on page 5

Driver Configuration

In Identity Console:

- 1 Click the **Identity Manager Administration** tile.
- 2 Select the driver set that contains the driver whose properties you want to edit.
- 3 Click the driver icon to display the driver’s properties page.

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](#)” on page 2
- ♦ “[Driver Object Password](#)” on page 2
- ♦ “[Authentication](#)” on page 2
- ♦ “[Startup Option](#)” on page 3
- ♦ “[Driver Parameters](#)” on page 3
- ♦ “[ECMAScript](#)” on page 4
- ♦ “[Global Configurations](#)” on page 5

Driver Module

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

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

The name of the Java class is:

```
com.novell.idm.dirxml.driver.idprovider.IDProviderShim
```

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

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

Driver Object Password

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

Authentication

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

Authentication ID: Specify a user application ID. This ID is used to pass Identity Vault subscription information to the application. For example, `Administrator`.

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

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

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

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

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

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

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

Startup Option

The startup options allow 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 Identity Console.

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

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

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 driver parameters are divided into categories:

- ♦ [“ID Policy Repository” on page 3](#)
- ♦ [“Client Options” on page 4](#)
- ♦ [“Server Options” on page 4](#)

ID Policy Repository

LDAP server: The IP address or DNS name of the LDAP server that contains the ID policies.

LDAP port: The TCP port of the LDAP server. The default port is 389 for a non-SSL connection and 636 for an SSL connection.

Use SSL: Select **True** to use an SSL/TLS connection to the LDAP server.

Always trust: If this options is set to **True**, the ID Provider driver trusts all LDAP servers even if their certificates are untrusted.

Policy Container DN: Specify the DN of the policy container in your Identity Vault.

Client Options

Client name: The name the driver uses when it acts as an ID client and requests an ID from the provider. This is useful for tracing, and if access control is enabled on any of the ID policies. If access control is enabled, the ID client names that obtain an ID from the policy are specified. If the client name associated with the request is not in the list, the provider does not issue an ID.

ID Generation Map: Specify a comma-separated list of attribute=policy pairs. For example:
`workforceID=wfid,uniqueID=uid.`

This example configures the driver to request IDs from the wfid policy and store them in the workforceID attribute whenever a new object is created or when someone tries to change this attribute. IDs from the uid policy are used for the uniqueID attribute. The driver only issues IDs from an attribute if that attribute and the object class holding the attribute are in the Subscriber channel and the Publisher channel of the filter and are set to synchronize.

NOTE: Attribute names must be in the Identity Manager namespace (not LDAP) and must be case-exact.

Server Options

Start RMI?: Controls whether the ID Provider starts an RMI service or not. An RMI service is needed if you request IDs from other clients than the driver for example, from DirXML Script policies or style sheets. If all IDs are managed through this driver's filter and ID Generation Map settings, no RMI service is needed.

RMI server: The IP address of the RMI server. Leave this field blank for the server to bind to all available addresses.

RMI port: The TCP port of the RMI service. The default port for RMI is 1099. If that port is in use, change to a different port that is lower than 1024. If the port is in conflict, you see errors in the trace when the driver starts. The configuration assumes a port of 1099 to avoid common port conflicts.

RMI Service port: The TCP port for the RMI ID Provider service. The server uses an ephemeral port if the value of this parameter is zero.

RMI ID Provider Service IP: The IP address of the server where the ID Provider service is available. This parameter is useful in NAT environments where ID Provider service has to be on a specific network interface. The IP address of the server where the client can access the ID Provider Service. In a NAT environment, this is the public IP address through which the clients can connect to the Identity Vault server.

Use legacy ID Server schema: Enables a backward-compatibility mode when migrating an existing ID server configuration to run with the new ID Provider shim. True allows you to use legacy ID policies which do not use the schema that ships with the ID Provider driver.

Trace level: Select On to enable the ID Provider trace level, not the driver trace level.

ECMAScript

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

Global Configurations

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

Global Configuration Values

Global configuration values (GCVs) allow you to specify settings for the Identity Manager features such as driver heartbeat, as well as settings that are specific to the function of an individual driver configuration. The ID Provider driver does not include any preconfigured GCVs.

To access the driver's GCVs in Identity Console:

- 1 Click the **Identity Manager Administration** tile.
- 2 Select the driver set that contains the driver whose properties you want to edit.

- 3 Locate the driver icon, then click the driver icon to display the driver's properties page.
- 4 Click **Global Config Values** drop down to display the GCV page.

In Designer:

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

