

ID Provider Driver Implementation Guide

Novell[®] Identity Manager

4.0

October 15, 2010

www.novell.com



Legal Notices

Novell, Inc., makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc., makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. See the [Novell International Trade Services Web page \(http://www.novell.com/info/exports/\)](http://www.novell.com/info/exports/) for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Copyright © 2008-2010 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.
www.novell.com

Online Documentation: To access the latest online documentation for this and other Novell products, see the [Novell Documentation Web page \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

Novell Trademarks

For Novell trademarks, see [the Novell Trademark and Service Mark list \(http://www.novell.com/company/legal/trademarks/tmlist.html\)](http://www.novell.com/company/legal/trademarks/tmlist.html).

Third-Party Materials

All third-party trademarks are the property of their respective owners.

Contents

About This Guide	7
1 Understanding the ID Provider Driver	9
1.1 Why Use the Driver?	9
1.2 Design Architecture	9
1.3 Schema Architecture	11
2 Installing Driver Files	15
3 Creating a New Driver	17
3.1 Creating the Driver in Designer	17
3.1.1 Importing the Current Driver Packages	17
3.1.2 Installing the Driver Packages	18
3.1.3 Configuring the Driver Settings	19
3.1.4 Deploying the Driver	22
3.1.5 Starting the Driver	23
3.2 Creating the Driver in iManager	23
3.3 Creating ID Policies	23
3.3.1 Default Policies	23
3.3.2 Creating an ID Policy	23
3.3.3 Managing the Access Control List	24
3.4 Activating the Driver	24
4 Upgrading an Existing Driver	25
4.1 What's New in Version 4.0	25
4.2 Upgrade Procedure	25
5 Configuring ID Clients	27
5.1 ID Client	27
5.2 Standalone Client	28
6 Managing the ID Provider Driver	29
7 Troubleshooting	31
A Driver Properties	33
A.1 Driver Configuration	33
A.1.1 Driver Module	34
A.1.2 Driver Object Password (iManager Only)	34
A.1.3 Authentication	34
A.1.4 Startup Option	35
A.1.5 Driver Parameters	35
A.1.6 ECMAScript	36

A.1.7	Global Configurations	36
A.2	Global Configuration Values	36

About This Guide

This guide explains the purpose of the ID Provider driver and how to implement the driver.

- ◆ [Chapter 1, “Understanding the ID Provider Driver,” on page 9](#)
- ◆ [Chapter 2, “Installing Driver Files,” on page 15](#)
- ◆ [Chapter 3, “Creating a New Driver,” on page 17](#)
- ◆ [Chapter 4, “Upgrading an Existing Driver,” on page 25](#)
- ◆ [Chapter 5, “Configuring ID Clients,” on page 27](#)
- ◆ [Chapter 6, “Managing the ID Provider Driver,” on page 29](#)
- ◆ [Chapter 7, “Troubleshooting,” on page 31](#)
- ◆ [Appendix A, “Driver Properties,” on page 33](#)

Audience

This guide is intended for Identity Manager administrators.

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 the most recent version of this guide, visit the [Identity Manager Drivers Documentation Web site \(http://www.novell.com/documentation/idm40drivers\)](http://www.novell.com/documentation/idm40drivers).

Additional Documentation

For documentation on Identity Manager, see the [Identity Manager Documentation Web site \(http://www.novell.com/documentation/idm40/index.html\)](http://www.novell.com/documentation/idm40/index.html).

Understanding the ID Provider Driver

1

The ID Provider driver 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.

- ♦ [Section 1.1, “Why Use the Driver?” on page 9](#)
- ♦ [Section 1.2, “Design Architecture,” on page 9](#)
- ♦ [Section 1.3, “Schema Architecture,” on page 11](#)

1.1 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.

1.2 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 Metadirectory 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.

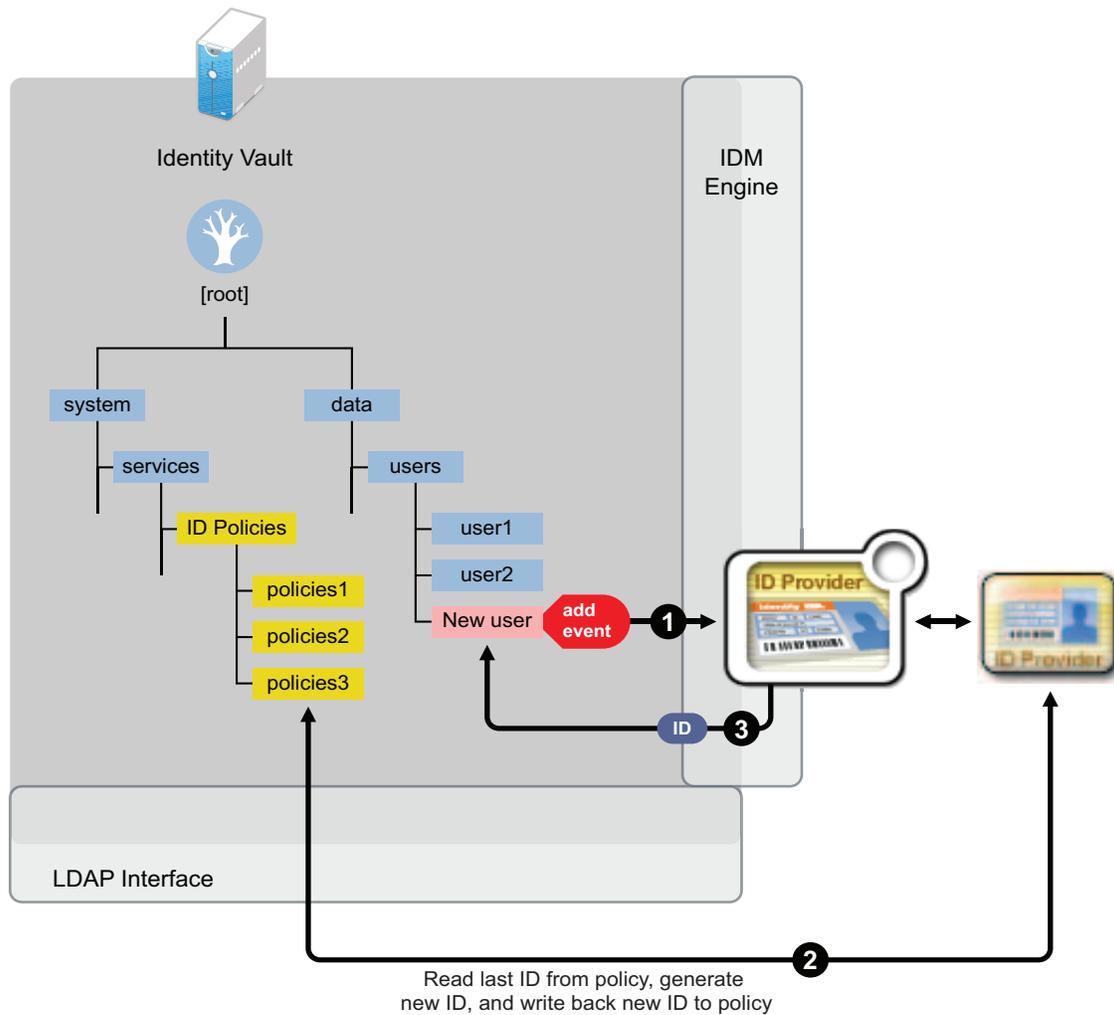
The ID Provider driver can be used in two different scenarios:

- ♦ “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



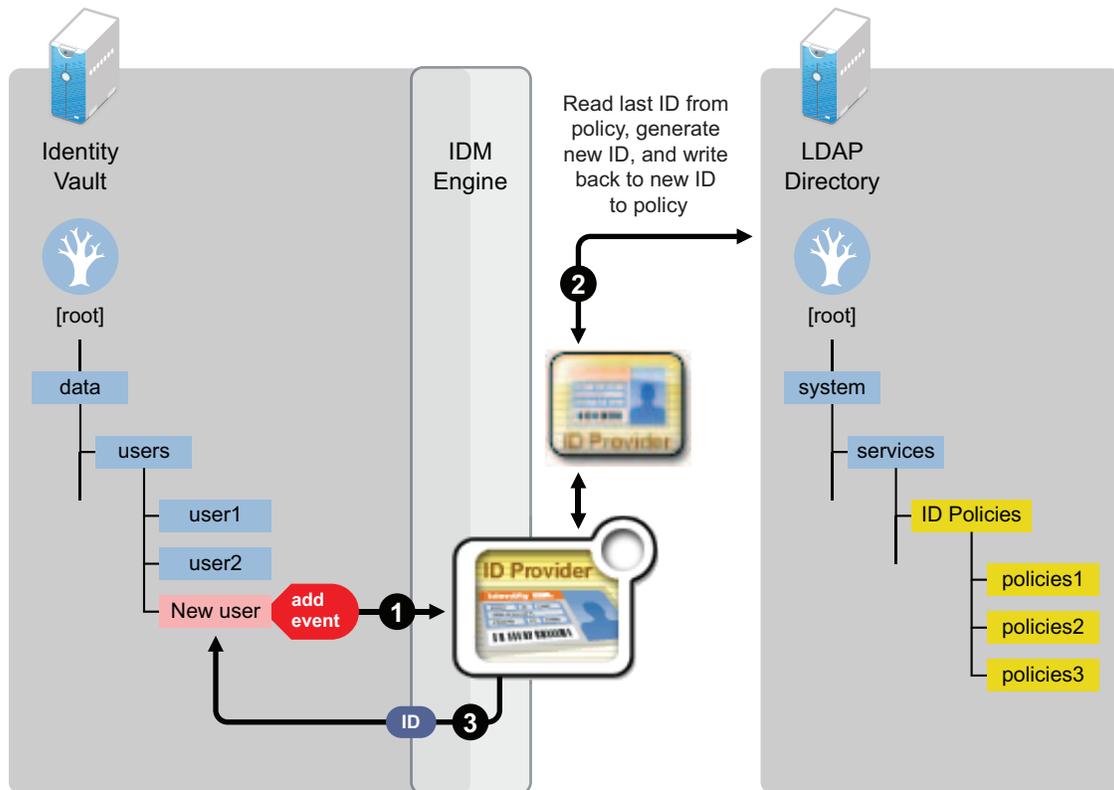
1. A new User object is created in the Identity Vault, then the ID Provider driver picks up the Create event.
2. The ID Provider driver reads the last ID that was generated from the ID Provider policies in the Identity Vault and generates a new ID. The ID is then written back to the ID Provider policies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

1.3 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-IDPolName	Case Ignore String	Single valued Synchronize immediately	ID Policy object name
DirXML-IDPolLastID	Numeric String	Single-valued Synchronize immediately	Last delivered ID
DirXML-IDPolMin	Numeric String	Single-valued	Minimum value for an ID
DirXML-IDPolMax	Numeric String	Single-valued	Maximum value for an ID
DirXML-IDPolPrefix	Case Ignore String	Single-valued	Prefix for a new ID
DirXML-IDPolFill	Boolean	Single-valued	True: Fill ID with 0 up to maximum length False or Empty: Do nothing
DirXML-IDPolArea	Case Ignore String	Single-valued	Exclude/Include list for generated IDs
DirXML-IDPolAreaEI	Boolean	Single-valued	True: IDPolArea = Include list False or Empty: IDPolArea = Exclude list
DirXML-IDPolAccessControl	Boolean	Single-valued	True: IDPolACL list is used False or Empty: IDPolACL list is not used
DirXML-IDPolACL	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

Class Name	Contained By	Attributes Contained
ID Policy	ID Policy Container	IDPolACL IDPolAccessControl IDPolArea IDPolAreaEI IDPolFill IDPolLastID IDPolMax IDPolMin IDPolName IDPolPrefix

Installing Driver Files

2

The ID Provider Driver is a service driver that is included with the base Identity Manager product. The driver is installed when the Metadirectory engine and drivers are installed. For the installation instructions, see “[Installation](#)” in the *Identity Manager 4.0 Integrated Installation Guide*.

Creating a New Driver

3

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 15](#)), you must create the driver in the Identity Vault. You do so by installing driver packages and then modifying the driver configuration to suit your environment. The following sections provide instructions:

- ♦ [Section 3.1, “Creating the Driver in Designer,” on page 17](#)
- ♦ [Section 3.2, “Creating the Driver in iManager,” on page 23](#)
- ♦ [Section 3.3, “Creating ID Policies,” on page 23](#)
- ♦ [Section 3.4, “Activating the Driver,” on page 24](#)

3.1 Creating the Driver in Designer

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

You create the ID Provider driver by installing the driver packages and then modifying the configuration to suit your environment. After you create and configure the driver, you need to deploy it to the Identity Vault and start it.

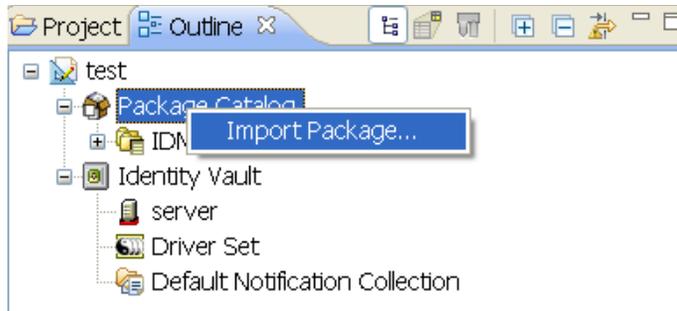
- ♦ [Section 3.1.1, “Importing the Current Driver Packages,” on page 17](#)
- ♦ [Section 3.1.2, “Installing the Driver Packages,” on page 18](#)
- ♦ [Section 3.1.3, “Configuring the Driver Settings,” on page 19](#)
- ♦ [Section 3.1.4, “Deploying the Driver,” on page 22](#)
- ♦ [Section 3.1.5, “Starting the Driver,” on page 23](#)

3.1.1 Importing the Current Driver Packages

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

To verify 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 [Section 3.1.2, “Installing the Driver Packages,”](#) on page 18.

3.1.2 Installing the Driver Packages

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

- 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:
 - 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 Click *Next*.
- 7 On the ID Provider page, fill in the following fields:
 - 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.
- 8 Click *Next*.
- 9 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 [Identity Manager 4.0 Remote Loader Guide](#).

If you select *No*, skip to [Step 10](#). If you select *Yes*, use the following information to complete the configuration of the Remote Loader.

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

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

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

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

- 10 Review the summary of tasks that will be completed to create the driver, then click *Finish*.
- 11 After the driver packages are installed, if you want to change the configuration of the Role-Based Entitlement driver, continue to [Section 3.1.3, “Configuring the Driver Settings,”](#) on page 19.

or

If you do not want to change the configuration of the driver, continue to [Section 3.1.4, “Deploying the Driver,”](#) on page 22.

3.1.3 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 [Section 3.1.5, “Starting the Driver,”](#) on page 23.

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

- ♦ [“ID Policy Repository”](#) on page 19
- ♦ [“Client Options”](#) on page 20
- ♦ [“Server Options”](#) on page 21

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,cd=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.

Parameter	Default Value	Description
ID Generation Map	workforceID=wfid	<p>Provide a comma-separated list of attribute=policy pairs.</p> <p>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.</p> <p>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.</p> <p>Attribute names must be in the Identity Namespace (not LDAP) and must be case-exact.</p>

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.

Parameter	Default Value	Description
Trace level	ALL	<p>This is not the driver trace level, but the ID Provider trace level. The levels are:</p> <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.

3.1.4 Deploying the Driver

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

- 1 In Designer, open your project.
- 2 In the Modeler, right-click the driver icon  or the driver line, then select *Live > Deploy*.
- 3 If you are authenticated to the Identity Vault, skip to [Step 5](#); otherwise, specify the follow 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 the deployment summary, then click *Deploy*.
- 6 Read the successful message, then click *OK*.
- 7 Click *Define Security Equivalence* to assign rights to the driver.

The driver requires rights to objects within the Identity Vault 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.

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

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

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

3.1.5 Starting the Driver

When a driver is created, it is stopped by default. To make the driver work, you must start the driver and cause events to occur. Identity Manager is an event-driven system, so after the driver is started, it 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.

3.2 Creating the Driver in iManager

Drivers are created with packages, and iManager does not support packages. In order to create or modify drivers, you must use Designer. See [Section 3.1, “Creating the Driver in Designer,”](#) on page 17.

3.3 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.

- ♦ [Section 3.3.1, “Default Policies,”](#) on page 23
- ♦ [Section 3.3.2, “Creating an ID Policy,”](#) on page 23
- ♦ [Section 3.3.3, “Managing the Access Control List,”](#) on page 24

3.3.1 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.

3.3.2 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 [Section 3.3.3, "Managing the Access Control List,"](#) on page 24.

- 5 Click *OK* to save the information.

3.3.3 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.

3.4 Activating the Driver

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

For information on activation, refer to the "[Activating Novell Identity Manager Products](#)" in the *Identity Manager 4.0 Framework Installation Guide*.

Upgrading an Existing Driver

4

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

- ♦ [Section 4.1, “What’s New in Version 4.0,” on page 25](#)
- ♦ [Section 4.2, “Upgrade Procedure,” on page 25](#)

4.1 What’s New in Version 4.0

Driver content is delivered in packages instead of through a driver configuration file.

4.2 Upgrade Procedure

The process for upgrading the ID Provider driver is the same as for other Identity Manager drivers. For details instructions, see [“Upgrading Drivers to Packages”](#) in the *Identity Manager 4.0 Framework Installation Guide*.

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.

- ♦ [Section 5.1, “ID Client,” on page 27](#)
- ♦ [Section 5.2, “Standalone Client,” on page 28](#)

5.1 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
\$UIDRule	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

```
<xsl:variable name="RMIServer" select="'192.168.65.100'"/>
<xsl:variable name="RMIPort" select="'1099'"/>
<xsl:variable name="UIDRule" select="'Unix UID'"/>
<xsl:variable name="IDClient" select="'NIS-SUB-A-CST'"/>
<xsl:variable name="GIDRule" select="'Unix GID'"/>
<xsl:variable name="Tracelevel" select="'9'"/>
```

```
<xsl:variable name="uid"
select="id:getNextID($RMIServer,$RMIPort,$UIDRule,$IDClient,$Tracelevel)"/>
```

5.2 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. 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	-t 1
-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
```

Managing the ID Provider Driver

6

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 *Identity Manager 4.0 Common Driver Administration Guide*.

Troubleshooting

7

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 *Identity Manager 4.0 Common Driver Administration Guide*.

Driver Properties

A

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 *Identity Manager 4.0 Common Driver Administration Guide* for information about the common properties.

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

- ♦ [Section A.1, “Driver Configuration,” on page 33](#)
- ♦ [Section A.2, “Global Configuration Values,” on page 36](#)

A.1 Driver Configuration

In iManager:

- 1 In iManager, 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 ID Provider 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 properties page opens with the *Driver Configuration* tab displayed.

In Designer:

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

The Driver Configuration options are divided into the following sections:

- ♦ [Section A.1.1, “Driver Module,” on page 34](#)
- ♦ [Section A.1.2, “Driver Object Password \(iManager Only\),” on page 34](#)
- ♦ [Section A.1.3, “Authentication,” on page 34](#)
- ♦ [Section A.1.4, “Startup Option,” on page 35](#)
- ♦ [Section A.1.5, “Driver Parameters,” on page 35](#)
- ♦ [Section A.1.6, “ECMAScript,” on page 36](#)
- ♦ [Section A.1.7, “Global Configurations,” on page 36](#)

A.1.1 Driver Module

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

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

The 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.

A.1.2 Driver Object Password (iManager Only)

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

A.1.3 Authentication

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

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

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

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

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

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

A.1.4 Startup 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 iManager.

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

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

A.1.5 Driver Parameters

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

- ♦ [“ID Policy Repository” on page 35](#)
- ♦ [“Client Options” on page 35](#)
- ♦ [“Server Options” on page 36](#)

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.

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.

A.1.6 ECMAScript

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

A.1.7 Global 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.

A.2 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.

In iManager:

- 1 In iManager, 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 ID Provider 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 properties page opens with the *Driver Configuration* tab displayed.

In Designer:

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

