AUTHORIZED DOCUMENTATION

Administration Guide

Novell®
Identity Manager

3.5.1

December 22, 2009

www.novell.com

Legal Notices

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

Online Documentation: To access the online documentation for this and other Novell products, and to get updates, see www.novell.com/documentation.

Contents

About This Guide

Novell® Identity Manager is a data sharing and synchronization service that enables applications, directories, and databases to share information. It links scattered information and enables you to establish policies that govern automatic updates to designated systems when identity changes occur. Identity Manager provides the foundation for account provisioning, security, user self-service, authentication, authorization, automated workflow and Web services. It allows you to integrate, manage, and control your distributed identity information so you can securely deliver the right resources to the right people.

This guide provides an overview of the Identity Manager technologies, and also describes administration and configuration functions.

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 online documentation, or go to http://www.novell.com/documentation/feedback.html and enter your comments there.

Documentation Updates

For the most recent version of this document, see the Identity Manager Documentation Web site (http://www.novell.com/documentation/idm35).

Additional Documentation

For documentation on installing and upgrading Identity Manager, see the *Installation Guide* (http://www.novell.com/documentation/idm35).

For documentation on Identity Manager policies and filters, see the Policy Guides (http://www.novell.com/documentation/idm35).

For documentation on design and deployment practices, see the *Designer 2.1 for Identity Manager: Administration Guide* (http://www.novell.com/documentation/designer21/).

For documentation on password policies, password self-service, and managing passwords, see the *Password Management Administration Guide* (http://www.novell.com/documentation/password_management/index.html).

For documentation on using the Identity Manager drivers, see the Identity Manager Driver Documentation Web site (http://www.novell.com/documentation/idm35drivers/index.html).

Documentation Conventions

In this documentation, a greater-than symbol (>) is used to separate actions within a step and items within a cross-reference path.

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark.

Overview of the Identity Manager Architecture

1

Identity Manager has three major components.

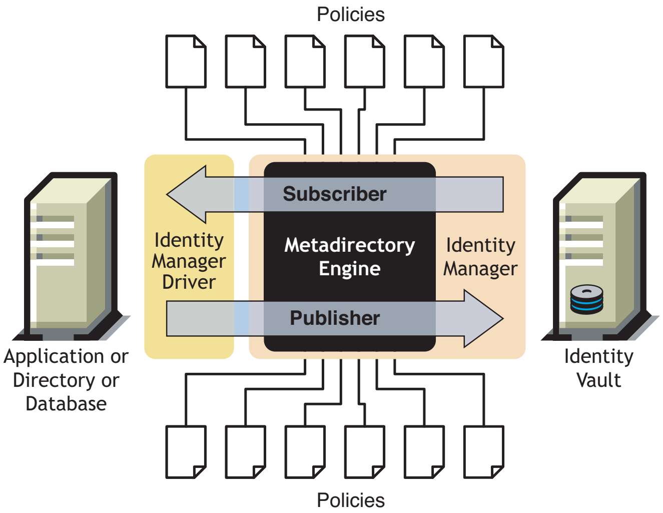
- Section 1.1, “Identity Manager,” on page 13
- Section 1.2, “User Application,” on page 21
- Section 1.3, “Designer,” on page 21

1.1 Identity Manager

Identity Manager provides for the synchronization of data between the Identity Vault and the connected system. The connected system consists of applications, directories, databases, or files.

Identity Manager includes of several components. The following illustration shows the basic components and their relationships:

Figure 1-1 *Identity Manager Components*



Integrated Architecture (http://www.novell.com/documentation/idm35/idm_log/data/b8ilr2g.html) The Metadirectory engine is the key module in the Identity Manager architecture. It provides the interface that allows Identity Manager drivers to synchronize information with the Identity Vault, allowing even disparate data systems to connect and share data.

The Metadirectory engine processes Identity Vault data and Identity Vault events by using an XML format. The Metadirectory engine employs a rules processor and a data transformation engine to manipulate the data as it flows between two systems:

1. Reads the filter for all Identity Manager drivers.
2. Registers the drivers for the appropriate Identity Vault events.

3. Filters the data according to each driver's specifications.
4. Sets up a cache for the Identity Vault events passing through to each driver.

When the Identity Vault initializes, it does the following:

- After an event is cached, the driver that owns the cache reads the event.
- The driver receives the Identity Vault data in eDirectory native format, translates it into XDS format (the XML vocabulary used by Identity Manager that can be transformed by a policy), and sends the event to the Metadirectory engine. The engine reads all the policies in the connected system driver and creates XML-formatted data according to those policies, then sends the data to the connect system driver. It then sends the data to the connected system. For more information on policies, see *Understanding Policies for Identity Manager 3.5.1* (http://www.novell.com/documentation/idm35/policy/data/bookinfo.html).
- The Publisher portion of the driver performs the gathering and sending of updates from the connected system to the Identity Vault. When the connect system driver is informed of changes to the information the two systems are sharing, the connected system driver gathers the information, ensures that it has been filtered to the correct set of data, converts the data to XDS format, and sends the data to the engine.

1.1.1 Metadirectory Engine

The Metadirectory engine can be broken down into two components: the eDirectory interface and the synchronization engine.

eDirectory Interface

The eDirectory interface built into the Metadirectory engine is used to detect events that take place in eDirectory. This interface guarantees the delivery of events to Identity Manager by using the event cache. The eDirectory interface supports multiple driver loading, which means that only one instance of Identity Manager is running for that eDirectory server, but it can communicate with multiple connected systems. Loopback detection is built into this interface to prevent event loops from occurring between the Identity Vault and the connected system. Although the interface contains loopback protection, developers are encouraged to also build loopback detection into the individual connected system drivers.

Synchronization Engine

The synchronization engine applies the Identity Manager policies to each event presented to it. The policies are created in the Policy Builder using DirXML Script. The Policy Builder allows you to create policies through a GUI interface instead of using XML documents or stylesheets written in XSLT. You can use still use style sheets, but the Policy Builder is simpler to use. For more information about the Policy Builder or DirXML Script, see *Understanding Policies for Identity Manager 3.5.1* (http://www.novell.com/documentation/idm35/policy/data/bookinfo.html).

The synchronization engine applies each type of policy to the source document. The ability to complete these transformations is one of the most powerful capabilities of Identity Manager. Data is transformed in real time as it is shared between the Identity Vault and the connected systems.

1.1.2 Driver Configuration Files

Driver configurations are preconfigured XML files that are included with Identity Manager. You can import these configuration files through the wizards in iManager and Designer.

These driver configurations contain sample policies. They are not intended for use in a production environment, but rather as templates for you to modify.

1.1.3 Identity Manager Event Cache

All of the events generated through eDirectory are stored in an event cache until they are successfully processed. This guarantees that no data is lost because of a bad connection, loss of system resources, unavailability of a driver, or any other network failure.

1.1.4 Driver Shim

The driver shim serves as a conduit for information between the connected system and the Identity Vault. The shim is written in Java*, C, or C++.

The communication between the Metadirectory engine and the driver shim is in the form of XML documents that describe events, queries, and results.The driver shim is commonly referred to as the driver. It is the conduit through which information is transferred between the Identity Vault and the connected system.

The following object events are supported by the shim:

- Add (creation)
- Modify
- Delete
- Rename
- Move
- Query

In addition, the shim must support a defined query capability so that Identity Manager can query the connected system.

When an event occurs in the Identity Vault that causes an action in the connected system, Identity Manager creates an XML document that describes the Identity Vault event and submits it through the Subscriber channel to the driver shim.

When an event occurs in the connected system, the driver shim generates an XML document that describes the connected system event. The driver shim then submits the XML document to Identity Manager through the Publisher channel. After processing the event through any Publisher policies, Identity Manager causes the Identity Vault to take the appropriate action.

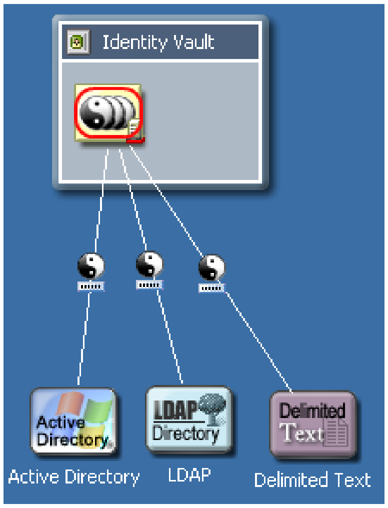
1.1.5 Driver Set

A driver set is a container object that holds Identity Manager drivers. A driver set can be associated with one server at a time. As a result, all running drivers must be grouped into the same driver set.

The driver set object must exist in a full read/write replica on any server that is using it, so we recommend partitioning the driver set. This is recommended so that if replicas of users are moved to another server, the driver objects are not.

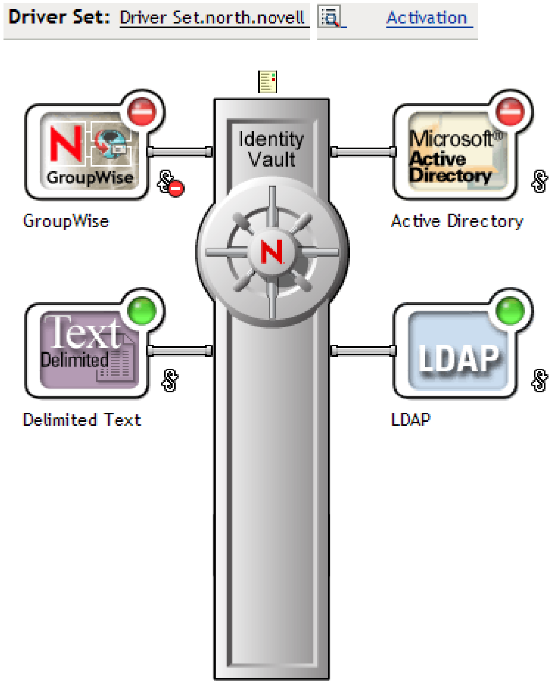
The following image shows how the driver set is displayed in Designer.

***Figure 1-2** Driver Set in Designer*



The following image shows how the driver set is displayed in iManager.

***Figure 1-3** Driver Set in iManager*



From the Modeler in Designer (shown above Figure 1-2) or the Overview page in iManager (shown above Figure 1-3), you can:

- View and modify the driver set and its properties
- View the drivers within the driver set
- Change the status of a driver
- Associate a driver set with a server
- Add or remove drivers
- View activation information for the driver set
- View the status log for the driver set

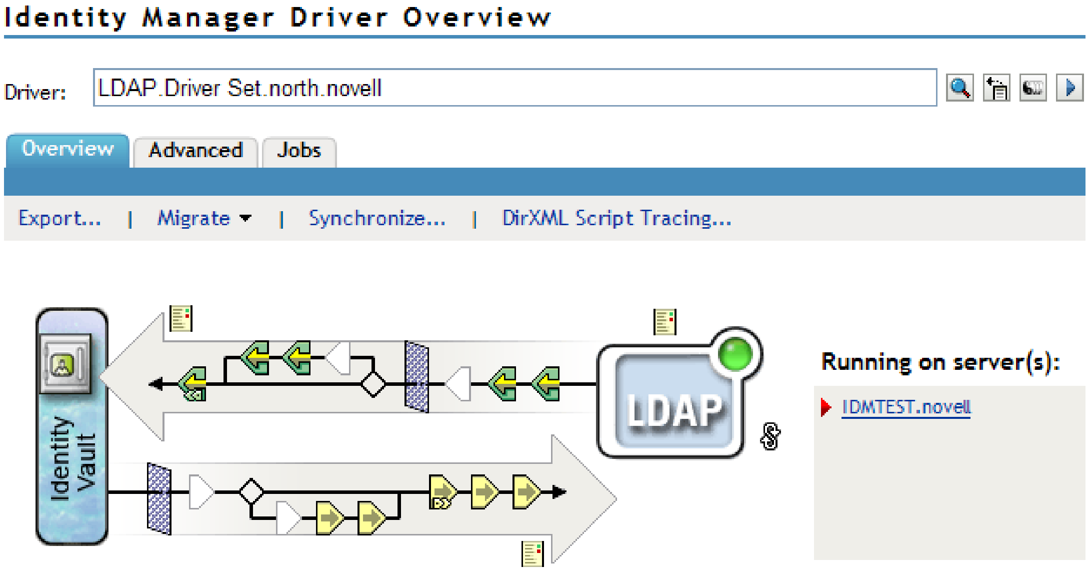
1.1.6 Driver Object

A Driver object represents a driver that connects to the connected system that integrates with the Identity Vault. The following components comprise the driver object and its configuration parameters:

- A Driver object in the eDirectory tree contained by a driver set object.
- A Subscriber channel object contained by the Driver object.
- A Publisher object contained by the Driver object.
- Several policy objects that are referenced by the Driver, Subscriber, and Publisher objects.
- An executable driver shim that is referenced by the Driver object.
- Shim-specific parameters that are configured by the administrator.
- An eDirectory password for the Driver object. The password can be used by the shim to authenticate a remote part of the shim.
- Authentication parameters used to connect to and authenticate to the connected system.
- Entitlements, although they are not part of every driver. Entitlements can be enabled during the creation of the driver or added later.
- A startup option for the driver that includes the following:
    - Disabled: The driver does not run.
    - Manual: The driver must be started manually through iManager.
    - Auto start: The driver starts automatically when the Identity Vault starts.
- A reference to a Schema Mapping policy.
- An XML representation of the connected system's schema. This is typically obtained automatically from the connected system through the shim.

In iManager, you can access the Identity Manager Driver Overview page and modify an existing driver's parameters, policies, style sheets, and entitlements. The Identity Manager Driver Overview is shown below.

***Figure 1-4** Identity Manager Driver Overview*



Driver: LDAP.Driver Set.north.novell

In addition, the Driver object is used for eDirectory rights checking. The Driver object must be granted sufficient eDirectory rights to any object it reads or writes. You can do this by making the Driver object a Trustee of the eDirectory objects the driver synchronizes with, or by granting Security Equivalences to the Driver object.

See eDirectory Rights (http://www.novell.com/documentation/edir88/index.html?page=/documentation/edir88/edir88/data/fbachifb.html) in the *Novell eDirectory 8.8 Administration Guide* for more information on rights assignments.

1.1.7 Publisher and Subscriber Channels

Identity Manager drivers contain two channels for processing data: the Publisher channel and Subscriber channel. The Publisher channel sends events from the connected system to the Identity Vault. The Subscriber channel sends events from the Identity Vault to the connected system. Each channel contains its own policies that define how to process and transform data.

Figure 1-5 Publisher and Subscriber Channels in Designer

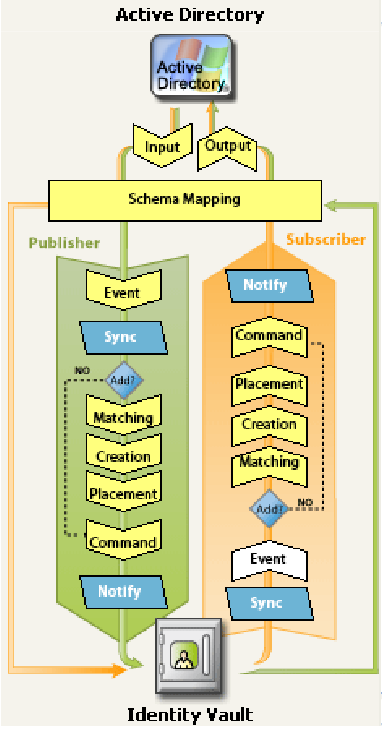
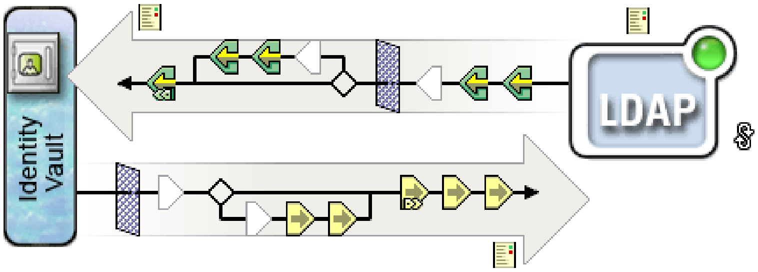


Figure 1-6 The Publisher and Subscriber channels in iManager



1.1.8 Events and Commands

The distinction between events and commands in Identity Manager is important. If an event is being sent to a driver, the event is a command. If the event is being sent to Identity Manager, the event is a notification. When the driver sends an event notification to Identity Manager, the driver is informing Identity Manager of a change that occurred in the connected system. Based on configurable rules, the Metadirectory engine then determines what commands, if any, must be sent to the Identity Vault.

When Identity Manager sends a command to the driver, Identity Manager has already taken an Identity Vault event as input, applied the appropriate policies, and determined that the change in the connected system represented by the command is necessary.

1.1.9 Policies and Filters

Policies and filters give you the ability to control how data flows from one system to another. It is through the rules in policies that you define how managing Identity Vault classes, attributes, and events are translated for use in the connected system, and visa-versa. For detailed information on policies and filters, refer to *Understanding Policies for Identity Manager 3.5.1* (http://www.novell.com/documentation/idm35/policy/data/bookinfo.html).

1.1.10 Associations

Most other identity management products require the connected system to store an identifier of some sort to map objects from a connected system to the directory. With Identity Manager, no changes are required of the connected system. Each object in the Identity Vault contains an association table that maps the Identity Vault object with a unique identifier in the connected systems. The table is reverse-indexed so that the connected system does not need to supply an Identity Vault identifier (such as a distinguished name) to the driver when updating the Identity Vault.

The creation of an association between two objects happens when an event occurs to an object that has not yet been associated with another object in the Identity Vault. For an association to be created, the minimum set of definable criteria must match between each object. For example, you can create a policy stating that if any two of four attributes match by more than 90% (full name, telephone number, employee ID, and e-mail address), the object will be associated.

Matching policies define the criteria for determining if two objects are the same. If no match is found for the changed object, a new object can be created. For this to occur, all of the minimum creation criteria must be met. These criteria are defined by a Create policy. Finally, the Placement policy defines where, in the naming hierarchy, the new object is created.

Associations can be created in one of two ways:

- As a match between objects
- As a new creation of an object in a specific location

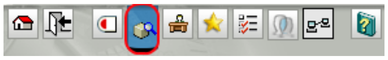
After an association between objects is formed, this association remains in effect until the objects are deleted or the association is deleted by an administrator.

Association Table

In Identity Manager, associations refer to the matching of objects in eDirectory with objects residing in connected systems. When Identity Manager is initially installed, the eDirectory schema is extended. Part of this extension is a new attribute tied to the base class of all eDirectory objects. This attribute is an association table. Association tables keep track of all the connected system objects that an eDirectory object is linked to. This table is built and maintained automatically, so there is rarely a reason to manually edit this information, although it is often helpful to view it.

The association attribute on the object can be viewed in iManager.

1 In iManager select the *View Objects* icon in the tool bar.



2 Browse to and select the object, then select *Modify Object*.

3 Select the Identity Manager tab.

The association attribute is displayed on the Identity Manager tab.

1.1.11 Identity Manager and Clustering

If you deploy Identity Manager in a clustered environment, Novell® will support Identity Manager running in the cluster, although in most situations, the cluster itself is not supported. The following two scenarios describe the extent of support given:

- If you run the Identity Manager engine or remote loader on SLES, and use Heartbeat to manage High Availability, everything is supported.
- If you run the Identity Manager engine or remote loader in a clustered environment on any other supported platform, support is extended for everything except the cluster management system.

NOTE: SLES is the only platform that is fully supported in a clustered environment.

For more information on how to configure a cluster with Identity Manager, refer to these resources:

- Configuring a Linux High Availability Cluster for IDM 3 and eDirectory 8.8 AppNote, at the Novell Cool Solutions Web site. (http://www.novell.com/coolsolutions/appnote/18591.html)
- Clustering eDirectory and IDM on Windows 2003 AppNote, at the Novell Cool Solutions Web site. (http://www.novell.com/coolsolutions/appnote/14856.html)
- High Availability on PolyServe Clusters AppNote, at the Novell Cool Solutions Web site. (http://www.novell.com/coolsolutions/appnote/16131.html)
- Setting Up an Identity Manager Cluster on Windows, at the Novell Support Web site. (http://www.novell.com/support/search.do?cmd=displayKC&docType=kc&externalId=3575742&sliceId=SAL_Public&dialogID=310596&stateId=1%200%20308676)

1.2 User Application

The User Application is a provisioning solution. It is an add-on product for Identity Manager. The User Application integrates powerful approval workflow with Identity Manager. This allows organizations to make provisioning decisions based on human input in addition to automated rules where no manual intervention is required. For information, see the User Application Documentation (http://www.novell.com/documentation/idm35).

1.3 Designer

Designer is a standalone client application. It consists of a Modeler space, a Palette, views, Policy Builder, a document generator, and other functionality so that you can design, test, document, and deploy Identity Manager-based solutions in a highly productive environment. For information about Designer, see the *Designer 2.1 for Identity Manager: Administration Guide (http://www.novell.com/documentation/designer21)*.

Managing Identity Manager Drivers 2

This section contains information that will help you create and manage your Identity Manager driver. Topics include:

- Section 2.1, "Creating and Configuring a Driver," on page 23
- Section 2.2, "Managing DirXML 1.1a Drivers in an Identity Manager Environment," on page 24
- Section 2.3, "Upgrading a Driver Configuration from DirXML 1.1a to Identity Manager 3.5.1 Format," on page 25
- Section 2.4, "Upgrading a Driver to the Identity Manager 3.5.1 Format," on page 26
- Section 2.5, "Starting, Stopping, or Restarting a Driver," on page 26
- Section 2.6, "Driver Parameters," on page 26
- Section 2.7, "Using the Inspector Tools," on page 26
- Section 2.8, "Using Global Configuration Values," on page 32
- Section 2.9, "Using the DirXML Command Line Utility," on page 32
- Section 2.10, "Viewing Version Information," on page 32
- Section 2.11, "Using Named Passwords," on page 37
- Section 2.12, "Reassociating a Driver Object with a Server," on page 44
- Section 2.13, "Adding Driver Heartbeat," on page 44
- Section 2.14, "Viewing Identity Manager Processes," on page 45

2.1 Creating and Configuring a Driver

For each Identity Manager driver you plan to use, you should create a driver object and import a driver configuration. The driver object contains configuration parameters and policies for that driver. As part of creating a driver object, you import a driver-specific configuration file. Driver configurations contain a default set of policies. These policies are intended to give you a good start as you implement your data sharing model. Most of the time, you will set up a driver using the shipping default configuration, and then modify the driver configuration according to the requirements of your environment.

There are two methods you can use for creating driver objects.

- The Create Driver task lets you create a single driver and import its driver configuration. For more information, refer to Section 2.1.1, "Creating a Driver Object," on page 24.
- The Import Driver task lets you create multiple drivers at the same time and import their configurations. For more information, refer to Section 2.1.2, "Creating Multiple Drivers," on page 24.

2.1.1 Creating a Driver Object

A driver configuration (XML) file creates and configures the objects needed in order for a driver to work properly. It also includes example policies you can modify for your implementation.

1. In iManager, select *Identity Manager Utilities > New Driver*.
2. Select a driver set where you want to create the driver, then click *Next*.

   If you place this driver in a new driver set, you must specify a driver set name, context, and associated server.
3. Select *Import a driver configuration from the server (.XML file)*, select the `.xml` file, then click *Next*.

   The driver configuration file is installed on the server when you install Identity Manager.
4. Follow the prompts to finish importing the driver configuration.

The necessary Identity Manager objects are created. If you didn't define security equivalences or exclude administrative users during the import, you can complete these tasks by modifying the properties of the driver object.

NOTE: If you do not enable Entitlements during the import process, the Entitlement policies are not created. If you want to use Entitlements in the future, you must create a new driver with Entitlements enabled.

2.1.2 Creating Multiple Drivers

Identity Manager provides the capability to create several drivers at once. The process is similar to creating a single driver because the driver configuration (XML) files still create and configure the objects needed in order for drivers to work properly.

To import several drivers at the same time:

1. In iManager, select *Identity Manager Utilities > Import Configurations*.
2. Select a driver set where you want to create new drivers, then click *Next*.

   If you place these drivers in a new driver set, you must specify a driver set name, context, and associated server.
3. Select the driver configurations to add to the driver set, then click *Next*.
4. Follow the prompts and specify the requested data, then click *Next*.

   When you select more than one configuration to import at a time, you are presented with the driver's configuration pages one at a time.

The necessary Identity Manager objects for each driver are created. If you didn't define security equivalences or exclude administrative users during the import, you can complete these tasks by modifying the properties of the driver object.

2.2 Managing DirXML 1.1a Drivers in an Identity Manager Environment

Existing drivers that were created for DirXML® 1.1a continue to run with Identity Manager.

The Metadirectory engine that ships with Identity Manager 3.5.1 is backward compatible with older drivers as long as the older driver shims and configurations have been updated with all the latest product updates and patches. Because the engine is backward compatible, you can run DirXML 1.1a drivers on the Identity Manager servers as long as you want to, without making any changes to them.

However, the iManager plug-ins have only limited backward compatibility. Older drivers can be viewed in the Overview of a driver set, but the driver configuration can't be viewed or edited without converting the driver. When you click a DirXML 1.1a driver in the Driver Set Overview, the Identity Manager plug-ins discover that the driver is in DirXML 1.1a format, and prompt you to convert the driver to 3.5 format using a wizard.

If you don't want to make any changes to an existing driver yet, you can cancel out of the wizard.

To edit a 1.1a driver in 1.1a format, you must use the DirXML 1.1a plug-ins. To do this, you must use a separate iManager Web server with the 1.1a plug-ins installed on it. You can't use the Identity Manager plug-ins that ship with Identity Manager to edit a driver configuration without converting the driver to Identity Manager 3.5.1 format.

2.3 Upgrading a Driver Configuration from DirXML 1.1a to Identity Manager 3.5.1 Format

The supported upgrade path from DirXML 1.1a is to install Identity Manager 3.5.1. The Identity Manager 3.5.1 installation installs new driver shims, but it does not change existing driver objects or driver configurations.

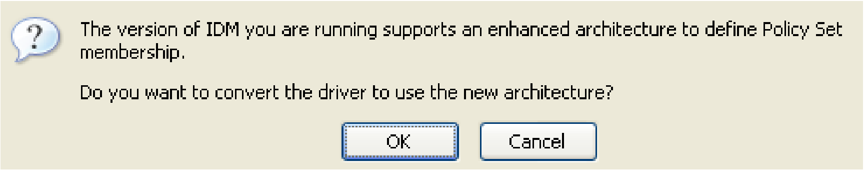
Existing driver configurations that were created for DirXML 1.1a continue to run with Identity Manager. However, the Identity Manager plug-ins let you edit only drivers that are in the Identity Manager 3.5.1 format.

IMPORTANT: Running an Identity Manager driver shim or driver configuration with a DirXML 1.1a engine is not supported.

A wizard is provided to help you convert DirXML 1.1a drivers to the Identity Manager format.

To start the wizard:

1. In iManager, click *Identity Manager* > *Identity Manager Overview*.
2. Select the driver set that contains the driver you want to convert, then click *Search*.
3. Click the icon for the driver you want to convert.
4. Read the message that is displayed, then click *OK*.



The version of IDM you are running supports an enhanced architecture to define Policy Set membership.

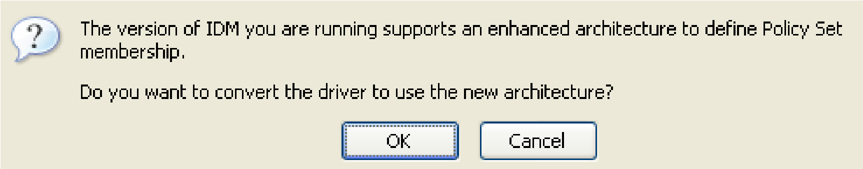
Do you want to convert the driver to use the new architecture?

5. If there is more than one driver to upgrade, repeat Step 2 through Step 4.

2.4 Upgrading a Driver to the Identity Manager 3.5.1 Format

Identity Manager 3.5.1 contains a new architecture for how policies reference one another. To take advantage of this new architecture, the driver configuration file must be upgraded. You can choose not to upgrade the driver, however, the iManager plug-ins have only limited backward compatibility. Older drivers can be viewed in the Overview of a driver set, but the driver configuration can't be viewed or edited without upgrading the driver.

1. In iManager, click *Identity Manager* > *Identity Manager Overview*.
2. Select the driver set that contains the driver you want to convert, then click *Search*.
3. Click the icon for the driver you want to convert.
4. Read the message that is displayed, then click *OK*.



The version of IDM you are running supports an enhanced architecture to define Policy Set membership.

Do you want to convert the driver to use the new architecture?

5. If there is more than one driver to upgrade, repeat Step 2 through Step 4.

2.5 Starting, Stopping, or Restarting a Driver

1. In iManager, click *Identity Manager* > *Identity Manager Overview*.
2. Browse to the driver set where the driver exists, then click *Search*.
3. Click the upper-right corner of the driver icon whose status you want to change, then click *Start driver* if the driver is stopped and *Stop driver* is the driver is running.

2.6 Driver Parameters

On the properties of each driver there are driver parameters. The parameters store information specific to the driver. The parameters stores information such as polling interval, authentication method, using SSL, or setting an heartbeat for the driver.

2.7 Using the Inspector Tools

iManager contains a set of tools that allows you to inspect objects associated with Identity Manager and Identity Manager objects.

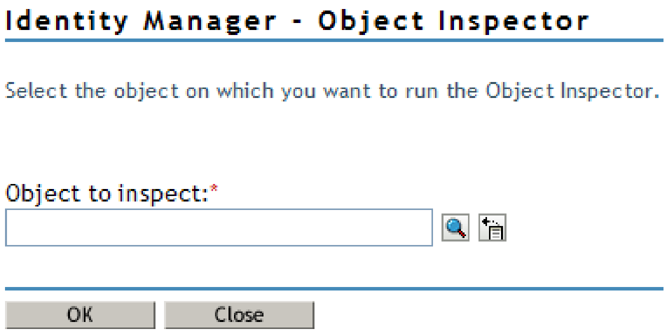
- Section 2.7.1, "Inspecting Objects," on page 27
- Section 2.7.2, "Inspecting Drivers," on page 28
- Section 2.7.3, "Inspecting the Driver's Cache File," on page 30

2.7.1 Inspecting Objects

The Identity Manager Object Inspector is a tool that an administrator can use to view detailed information about how an object participates in Identity Manager relationships. These relationships include the connected systems that are associated with the selected object, how data flows between the Identity Vault and the connected systems, the attribute values that are currently stored in the Identity Vault and on the connected systems, the connected system driver configurations, etc.

1 Click the *Object Inspector* task under the *Identity Manager* role to bring up the *Identity Manager Object Inspector* page.

This page allows you to select an object on which to run the Identity Manager Inspector.



Identity Manager - Object Inspector

Select the object on which you want to run the Object Inspector.

Object to inspect:*

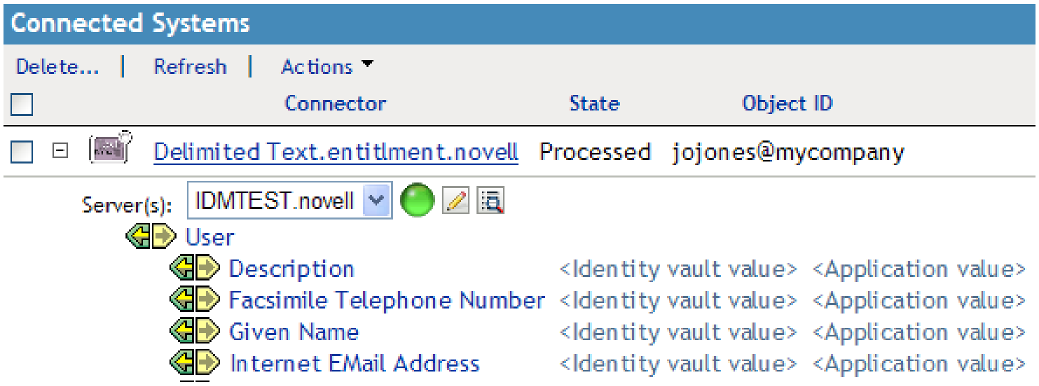
2 Specify the fully distinguished name of the object that you want to inspect, or click the *Object Selector* icon to select the desired object.

iManager keeps a record of the objects you have previously selected, so you can also use the *Object History* icon to select from a list of previously selected objects.

3 Click *OK* after you select the object.

The Object Inspector page lists information about the connected system in a table.

Figure 2-1 *Object Inspector*



Connected Systems

□	Connector	State	Object ID
□	Delimited Text.entitlment.novell	Processed	jojones@mycompany

Server(s): IDMTEST.novell

User

Description	<Identity vault value>	<Application value>
Facsimile Telephone Number	<Identity vault value>	<Application value>
Given Name	<Identity vault value>	<Application value>
Internet EMail Address	<Identity vault value>	<Application value>

- **Delete:** To delete an association with a connected system, select the check box to the left of the association and click *Delete*. To delete all associations, select the check box beneath the Delete column, then click *Delete*.
- **Refresh:** Select *Refresh* to re-read the connected system associations and refresh the table.

- **Actions:** Select a connected system by clicking the check box to the left of the association reference (you do not need to select any boxes for the Add New Association action item). Click *Actions* to expand its menu, which includes:
    - **Run Overview on Driver:** Launches the overview page for the connected system's driver in a pop-up window.
    - **Run Overview on Driver Set:** Launches the overview page for the connected system's driver set in a pop-up window.
    - **Configure Driver:** Launches the properties page for the connected system's driver in a pop-up window.
    - **Configure Driver Set:** Launches the properties page for the connected system's driver set in a pop-up window.
    - **Add New Association:** Prompts you for the parameters necessary to add new attribute values to the object's DirXML-Association attribute.
    - **Edit Association:** Prompts you to edit the parameters of the connected system's DirXML-Association attribute values.
- **Connector:** Prompts you to edit the parameters of the connected system's DirXML-Association attribute values.

    The Servers entry shows the servers that are associated with driver's driver set. Clicking the *Edit* icon to the right of the server brings up the server's properties page in a pop-up window. Clicking the *Query* icon queries the attribute values for all classes in the driver filter. The larger the filter, the longer the query takes. If the Inspector cannot communicate with the connected system, you see a message stating that the attribute cannot be queried from the application.

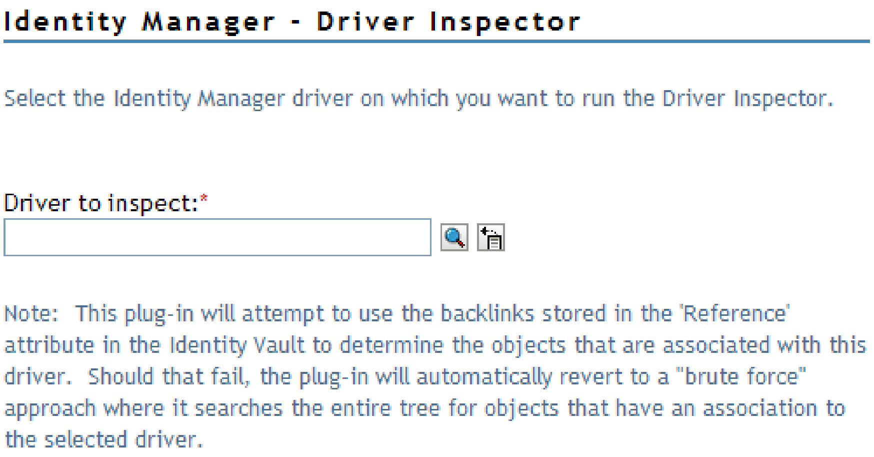
    The driver filter's associated classes (such as Group) and their attributes (such as Description and Member) are listed under the Server entry. Click the class to see all of the values for the defined attributes in that class. You can also click an attribute to see its values, or you can click the entries to the right of the attributes to see just the Identity Vault value or the application value. If no value has been defined, the entry displays No Values. If the Inspector cannot communicate with the connected system, you see a message stating that the attribute cannot be queried from the application.
- **States:** The connected system's driver states are Enabled, Disabled, Processed, Pending, Manual, and Migrate.
- **Object ID:** The identification value of the associated object to the connected system. If the connected system driver has no identification, this column displays *None*.

2.7.2 Inspecting Drivers

The Identity Manager Driver Inspector is a tool that an administrator can use to view detailed information about the objects associated with the selected driver. To access the Driver Inspector:

1 In iManager, select *Identity Manager > Driver Inspector*.

2 Browse to and select the driver to inspect, then click *OK*.



Identity Manager - Driver Inspector

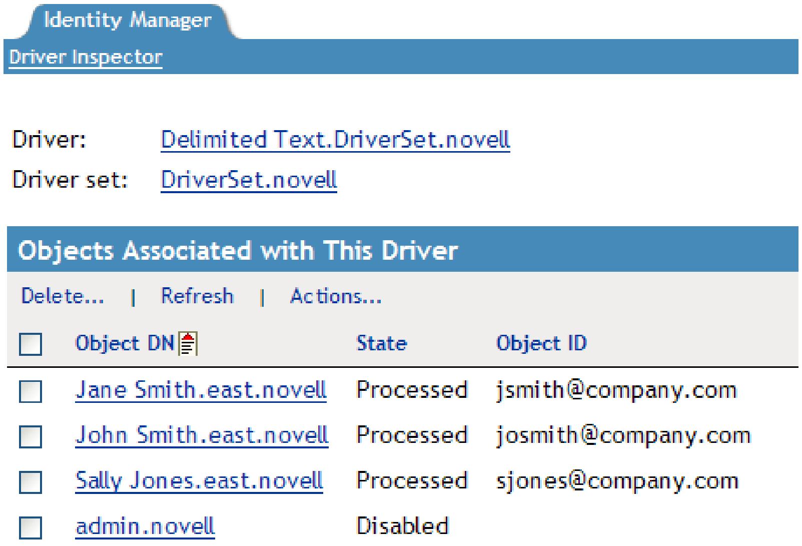
Select the Identity Manager driver on which you want to run the Driver Inspector.

Driver to inspect:*

Note: This plug-in will attempt to use the backlinks stored in the 'Reference' attribute in the Identity Vault to determine the objects that are associated with this driver. Should that fail, the plug-in will automatically revert to a "brute force" approach where it searches the entire tree for objects that have an association to the selected driver.

The Identity Manager Driver Inspector table displays information about the objects associated with the selected driver.

Figure 2-2 *Identity Manager Driver Inspector Table*



Driver: Delimited Text.DriverSet.novell

Driver set: DriverSet.novell

Objects Associated with This Driver

Delete... | Refresh | Actions...

□	Object DN	State	Object ID
□	Jane Smith.east.novell	Processed	jsmith@company.com
□	John Smith.east.novell	Processed	josmith@company.com
□	Sally Jones.east.novell	Processed	sjones@company.com
□	admin.novell	Disabled	

- **Driver:** A link to run the Driver Overview on the driver that is being inspected.
- **Driver Set:** A is a link to run the Driver Set Overview of the driver set that holds the driver.
- **Delete:** Deletes the associations of the selected objects.
- **Refresh:** Select this option to re-read all of the objects associated with the driver and refresh the displayed information.
- **Actions:** Perform actions on the objects associated with the driver. Click *Actions* to expand the menu, which includes:
  - **Show All Associations:** Displays all objects associated with the driver.
  - **Filter for Disabled Associations:** Displays all objects associated with the driver that have a Disabled state.

- **Filter for Manual Associations:** Displays all objects associated with the driver that have a Manual state.
- **Filter for Migrate Associations:** Displays all objects associated with the driver that have a Migrate state.
- **Filter for Pending Associations:** Displays all objects associated with the driver that have a Pending state.
- **Filter for Processed Associations:** Displays all objects associated with the driver that have a Processed state.
- **Filter for Undefined Associations:** Displays all objects associated with the driver that have an Undefined state.
- **Association Summary:** Displays the state of all objects associated with the driver.

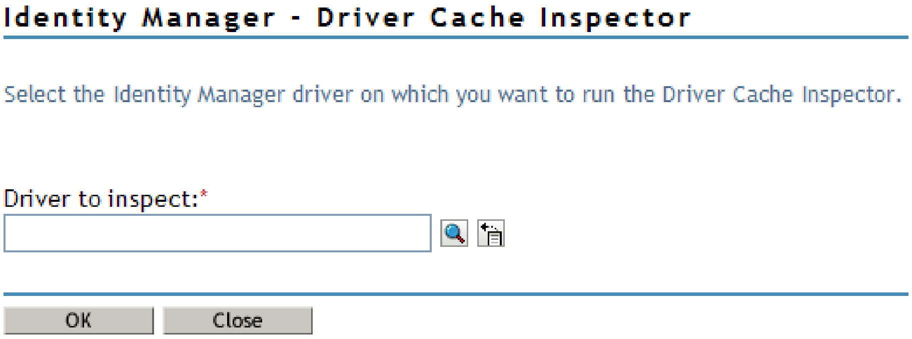
- **Object DN:** Displays the DN of the associated objects.
- **State:** Displays the association state of the object.
- **Object ID:** Displays the value of the association.

2.7.3 Inspecting the Driver's Cache File

The Identity Manager Driver Cache Inspector displays information about the cache file that stores events while the driver is stopped.

1 In iManager, select *Identity Manager* > *Driver Cache Inspector*.

2 Browse to and select the driver on which to run the Identity Manager Driver Cache Inspector.



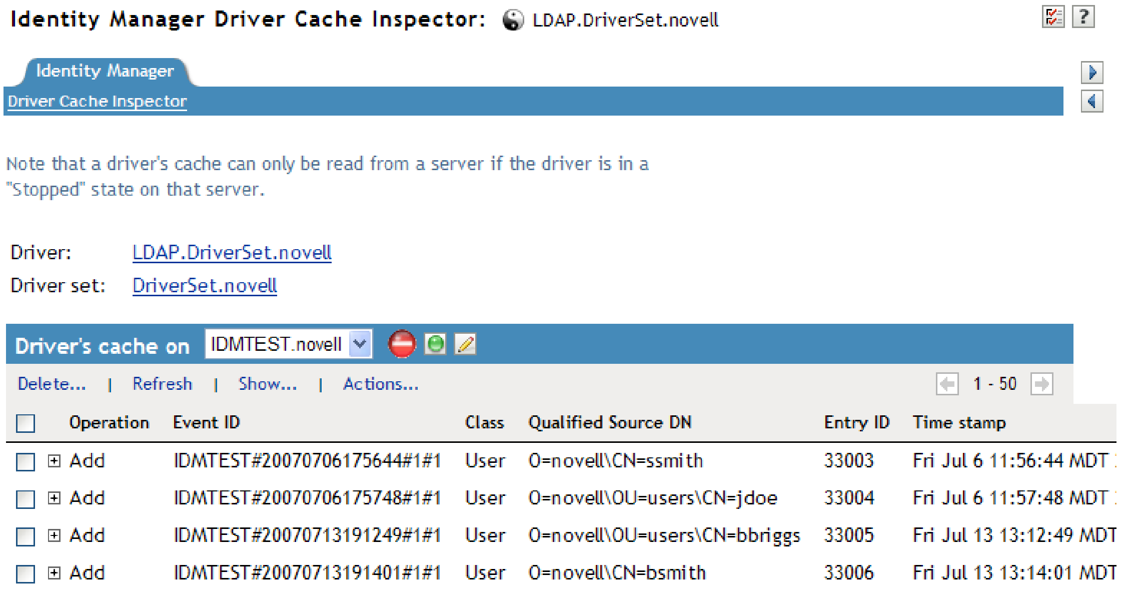
Identity Manager - Driver Cache Inspector

Select the Identity Manager driver on which you want to run the Driver Cache Inspector.

Driver to inspect:*

The Identity Manager Drive Cache Inspector page uses a table format to display information about the cache file that stores events while the driver is stopped.

Figure 2-3 Identity Manager Driver Cache Inspector



Identity Manager Driver Cache Inspector: LDAP.DriverSet.novell

Note that a driver's cache can only be read from a server if the driver is in a "Stopped" state on that server.

Driver: LDAP.DriverSet.novell

Driver set: DriverSet.novell

Driver's cache on IDMTEST.novell

Delete... | Refresh | Show... | Actions... 1 - 50

□	Operation	Event ID	Class	Qualified Source DN	Entry ID	Time stamp
□	⊞ Add	IDMTEST#20070706175644#1#1	User	O=novell\CN=ssmith	33003	Fri Jul 6 11:56:44 MDT
□	⊞ Add	IDMTEST#20070706175748#1#1	User	O=novell\OU=users\CN=jdoe	33004	Fri Jul 6 11:57:48 MDT
□	⊞ Add	IDMTEST#20070713191249#1#1	User	O=novell\OU=users\CN=bbriggs	33005	Fri Jul 13 13:12:49 MDT
□	⊞ Add	IDMTEST#20070713191401#1#1	User	O=novell\CN=bsmith	33006	Fri Jul 13 13:14:01 MDT

- **Driver:** A link to run the Driver Overview on the driver that is associated with this cache file.
- **Driver Set:** A link to run the Driver Set Overview on the driver set that holds the driver.
- **Driver's cache on:** Lists the server object that contains this instance of the cache file.
- **Start/Stop Driver icons:** Displays the current state of the driver and allows you to start or stop the driver.
- **Edit icon:** Allows you to edit the properties of the currently selected Server object.
- **Delete:** Deletes the selected items in the cache file.
- **Refresh:** Select this option to re-read the cache file and refresh the displayed information.
- **Show:** Limits the number of items to be displayed. The options are:
  - 25 per page
  - 50 per page
  - 100 per page
  - Other: Allows you to specify a desired number.
- **Actions:** Allows you to perform actions on the entries in the cache file. Click Actions to expand the menu, which includes:
  - **Expand All:** Expands all of the entries displayed in the cache file.
  - **Collapse All:** Collapses all of the entries displayed in the cache file.
  - **Go To:** Allows you to access a specified entry in the cache file. Specify the entry number, then click *OK*.
  - **Cache Summary:** Summarizes all of the events stored in the cache file.

2.8 Using Global Configuration Values

Global configuration values (GCVs) are settings that are similar to driver parameters. Global configuration values can be specified for a driver set as well as an individual driver. If a driver does not have a GCV value, the driver inherits the value for that GCV from the driver set.

GCVs allow you to specify settings for the Identity Manager features such as password synchronization and driver heartbeat, as well as settings that are specific to the function of an individual driver configuration. Some GCVs are provided with the drivers, but you can also add your own. You can refer to these values in a policy to help you customize your driver configuration.

IMPORTANT: Password synchronization settings are GCVs, but it's best to edit them in the graphical interface provided on the Server Variables page for the driver, instead of the GCV page. The Server Variables page that shows Password Synchronization settings is accessible as a tab like other driver parameters, or by clicking *Password Management > Password Synchronization*, searching for the driver, and clicking the driver name. The page contains online help for each Password Synchronization setting.

To add, remove, or edit GCVs that are not related to Identity Manager Password Synchronization:

1. In iManager, click *Identity Manager > Identity Manager Overview*.
2. Browse to and click the driver set or driver object, then click *Search*.
3. Click the upper right corner of the driver, then click *Edit properties*.
4. Select *Global Config Values*.
5. Change the default values that are set during the driver creation.
6. If you want to add additional information, click *Edit XML*.
7. Click *Enable XML editing*.
8. Add, remove, or edit the XML, then click *OK* to apply your changes.

2.9 Using the DirXML Command Line Utility

The DirXML Command Line Utility provides access for Identity Manager specific eDirectory verbs. This utility is not a replacement for iManager or Designer. The primary use of this utility is for scripting purposes. See Appendix A, "DirXML Command Line Utility," on page 307 for detailed information about the DirXML Command Line Utility. For daily tasks, use iManager or Designer.

2.10 Viewing Version Information

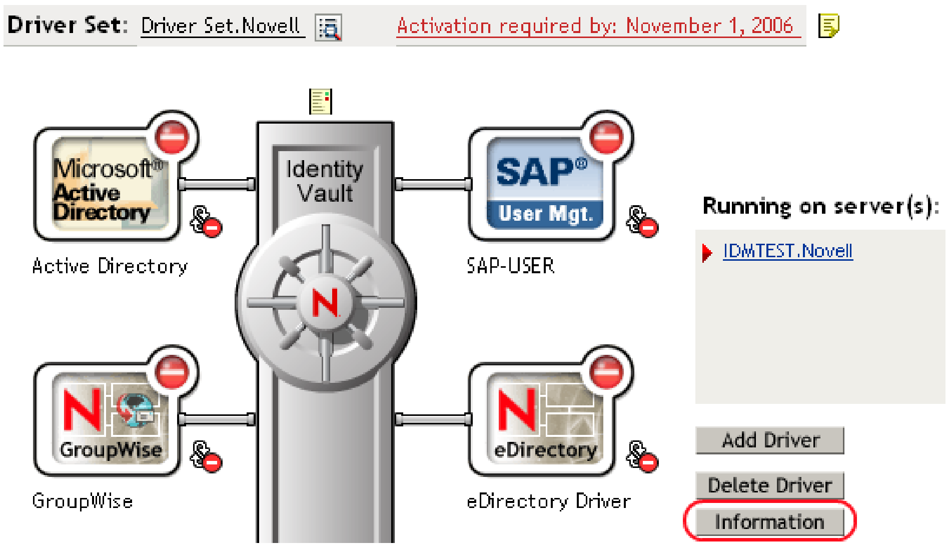
The Version Discovery Tool enables you to do the following:

- Section 2.10.1, "View a Hierarchical Display of Version Information," on page 32
- Section 2.10.2, "View the Version Information as a Text File," on page 34
- Section 2.10.3, "Saving Version Information," on page 36

2.10.1 View a Hierarchical Display of Version Information

1. In iManager click *Identity Manager > Identity Manager Overview*, then click *Search* to find your driver set.

2 Click *Information*, in the Identity Manager Overview screen.

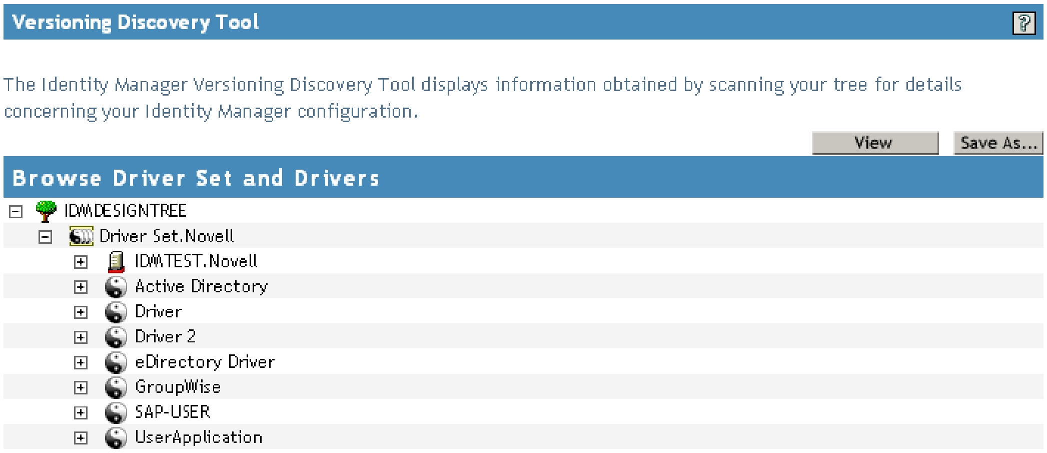


Driver Set: Driver Set.Novell

Activation required by: November 1, 2006

You can also select *Identity Manager Utilities* > *Versions Discovery*, then browse to and select the driver set, then click *OK*.

3 View a top-level or unexpanded display of versioning information.



Versioning Discovery Tool

The Identity Manager Versioning Discovery Tool displays information obtained by scanning your tree for details concerning your Identity Manager configuration.

Browse Driver Set and Drivers

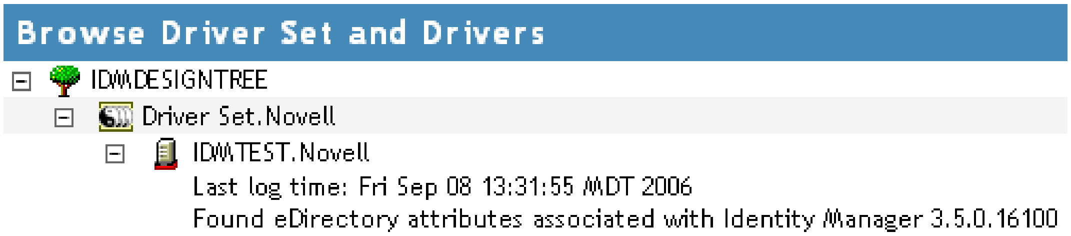
- IDMDESIGNTREE
  - Driver Set.Novell
    - IDMTEST.Novell
    - Active Directory
    - Driver
    - Driver 2
    - eDirectory Driver
    - GroupWise
    - SAP-USER
    - UserApplication

The unexpanded hierarchical view displays the following:

- The eDirectory tree that you are authenticated to
- The driver set that you selected
- Servers that are associated with the driver set

  If the driver set is associated with two or more servers, you can view Identity Manager information on each server.
- Drivers

4 View version information related to servers by expanding the server icon.



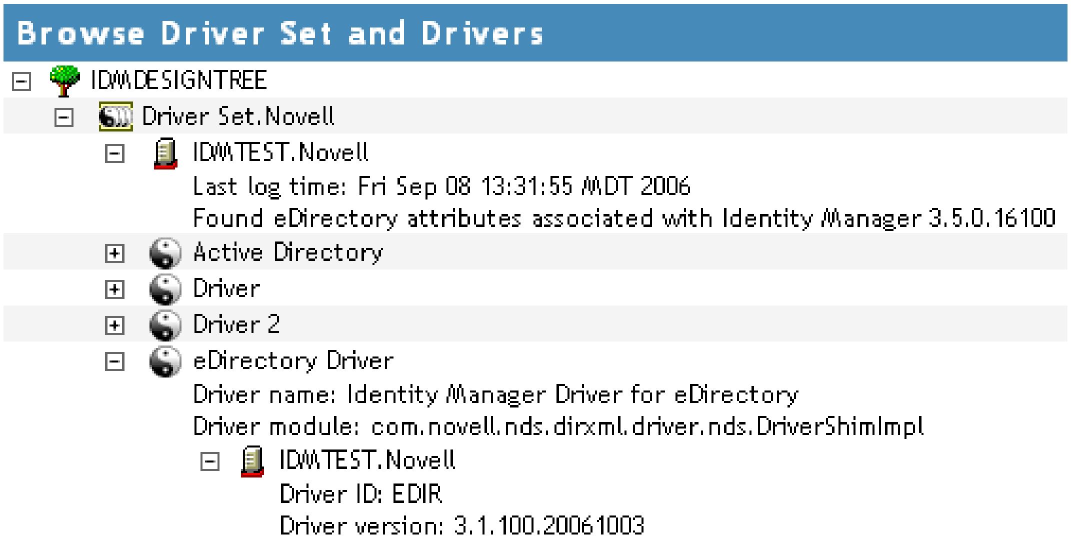
Browse Driver Set and Drivers

```
⊟ IDMDESIGNTREE
    ⊟ Driver Set.Novell
        ⊟ IDMTEST.Novell
            Last log time: Fri Sep 08 13:31:55 MDT 2006
            Found eDirectory attributes associated with Identity Manager 3.5.0.16100
```

The expanded view of a top-level server icon displays the following:

- Last log time
- Version of Identity Manager that is running on the server

5 View version information related to drivers by expanding the driver icon.



Browse Driver Set and Drivers

```
⊟ IDMDESIGNTREE
    ⊟ Driver Set.Novell
        ⊟ IDMTEST.Novell
            Last log time: Fri Sep 08 13:31:55 MDT 2006
            Found eDirectory attributes associated with Identity Manager 3.5.0.16100
        ⊞ Active Directory
        ⊞ Driver
        ⊞ Driver 2
        ⊟ eDirectory Driver
            Driver name: Identity Manager Driver for eDirectory
            Driver module: com.novell.nds.dirxml.driver.nds.DriverShimImpl
            ⊟ IDMTEST.Novell
                Driver ID: EDIR
                Driver version: 3.1.100.20061003
```

The expanded view of a top-level driver icon displays the following:

- The driver name
- The driver module (for example, com.novell.nds.dirxml.driver.nds.DriverShimImpl)

The expanded view of a server under a driver icon displays the following:

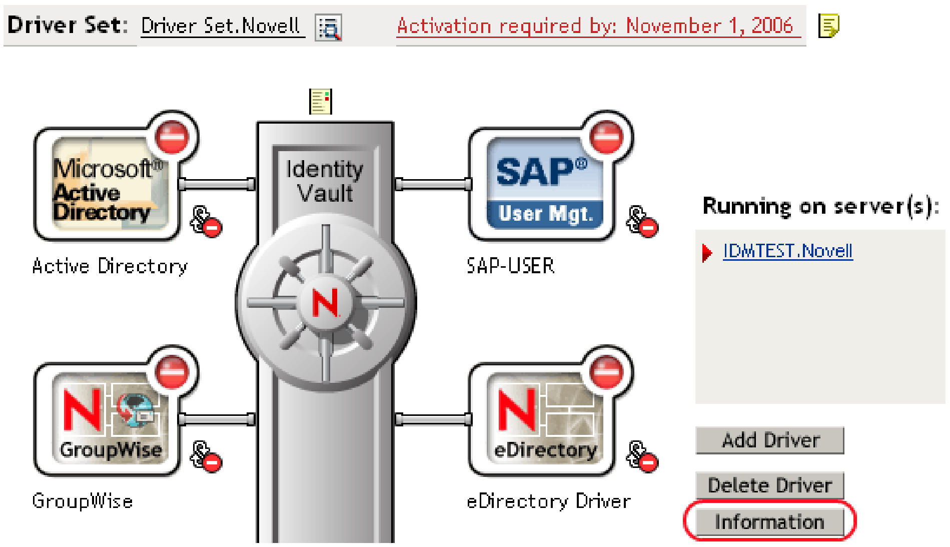
- The driver ID
- The version of the instance of the driver running on that server

2.10.2 View the Version Information as a Text File

Identity Manager publishes versioning information to a file. You can view this information in text format. The textual representation is the same information contained in the hierarchical view.

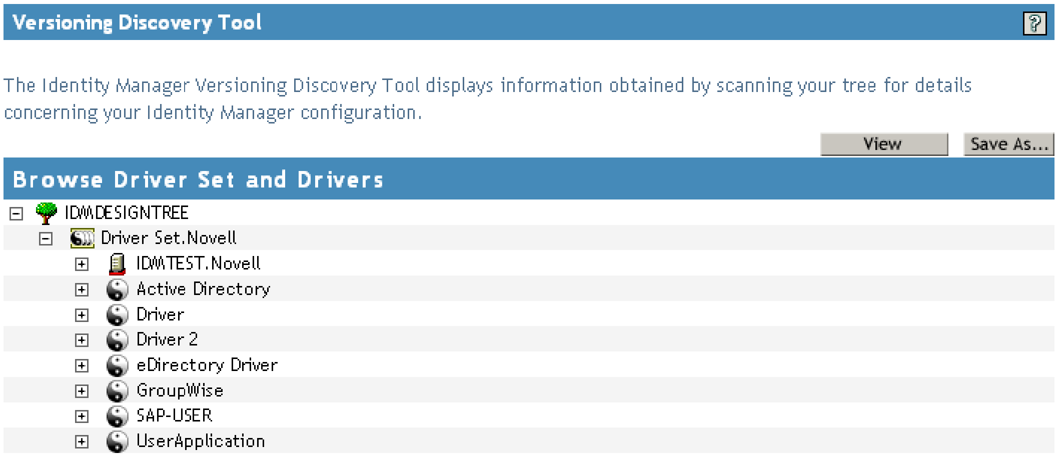
1 In iManager click *Identity Manager > Identity Manager Overview*, then click *Search* to find your driver set.

2 Click *Information* in the Identity Manager Overview screen.



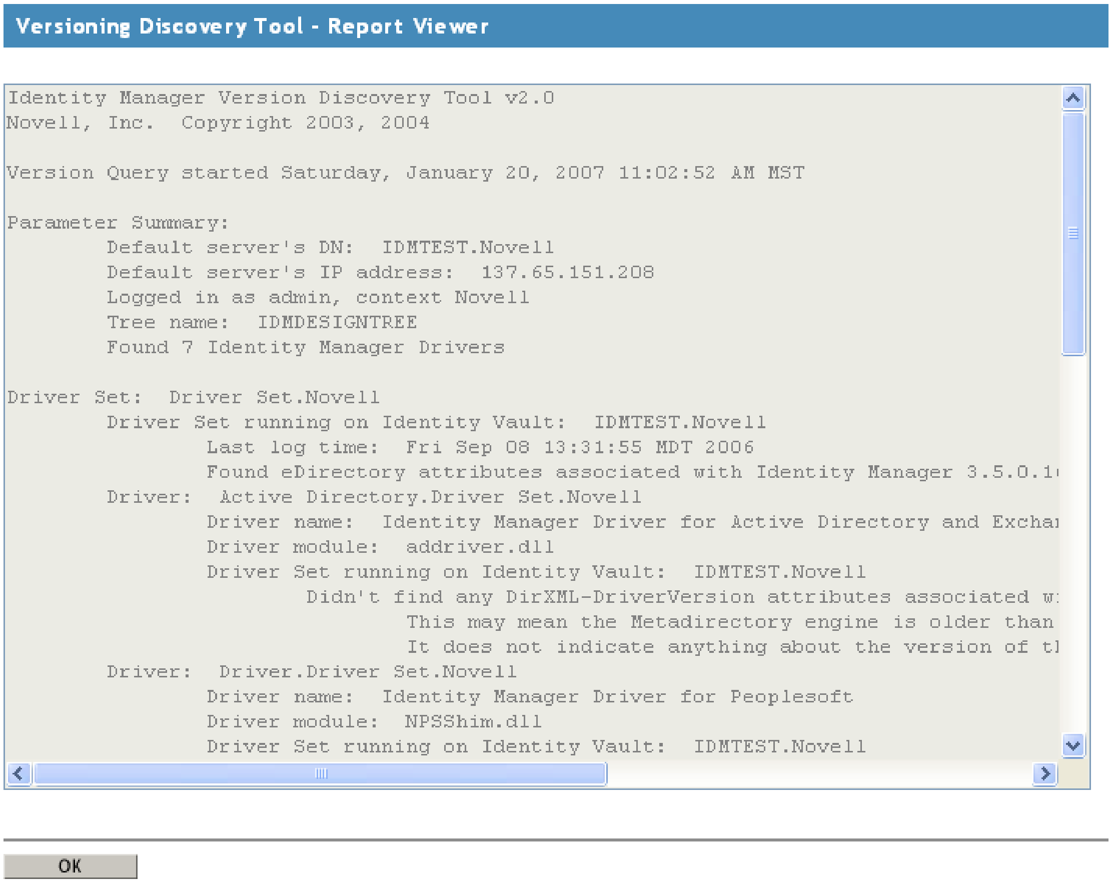
You can also select *Identity Manager Utilities > Versioning Discovery*, then browse to and select the driver set, then click *Information*.

3 In the Versioning Discovery Tool dialog box, click *View*.



The Identity Manager Versioning Discovery Tool displays information obtained by scanning your tree for details concerning your Identity Manager configuration.

The information is displayed as a text file in the Report Viewer window.

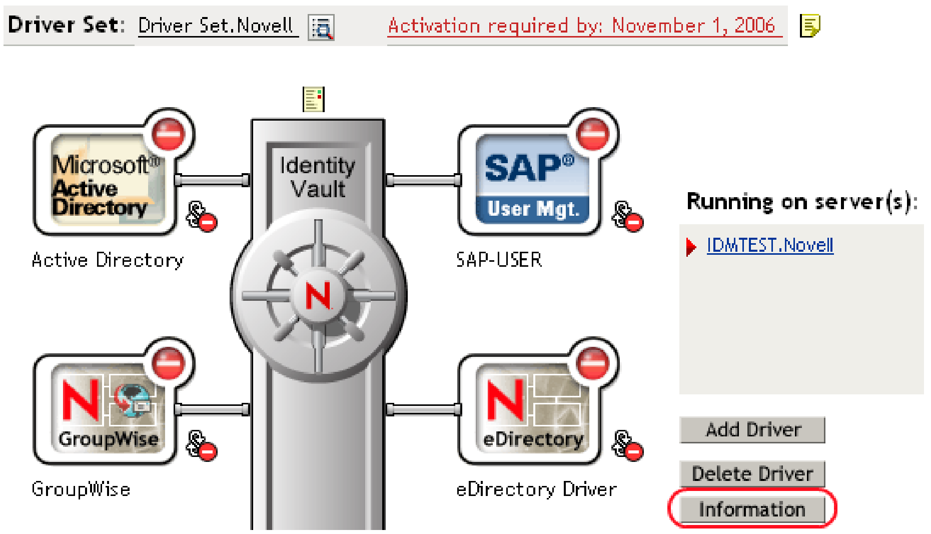


```
Identity Manager Version Discovery Tool v2.0
Novell, Inc.  Copyright 2003, 2004

Version Query started Saturday, January 20, 2007 11:02:52 AM MST

Parameter Summary:
        Default server's DN:  IDMTEST.Novell
        Default server's IP address:  137.65.151.208
        Logged in as admin, context Novell
        Tree name:  IDMDESIGNTREE
        Found 7 Identity Manager Drivers

Driver Set:  Driver Set.Novell
        Driver Set running on Identity Vault:  IDMTEST.Novell
                Last log time:  Fri Sep 08 13:31:55 MDT 2006
                Found eDirectory attributes associated with Identity Manager 3.5.0.1
        Driver:  Active Directory.Driver Set.Novell
                Driver name:  Identity Manager Driver for Active Directory and Excha
                Driver module:  addriver.dll
                Driver Set running on Identity Vault:  IDMTEST.Novell
                        Didn't find any DirXML-DriverVersion attributes associated w
                                This may mean the Metadirectory engine is older than
                                It does not indicate anything about the version of t
        Driver:  Driver.Driver Set.Novell
                Driver name:  Identity Manager Driver for Peoplesoft
                Driver module:  NPSShim.dll
                Driver Set running on Identity Vault:  IDMTEST.Novell
```

2.10.3 Saving Version Information

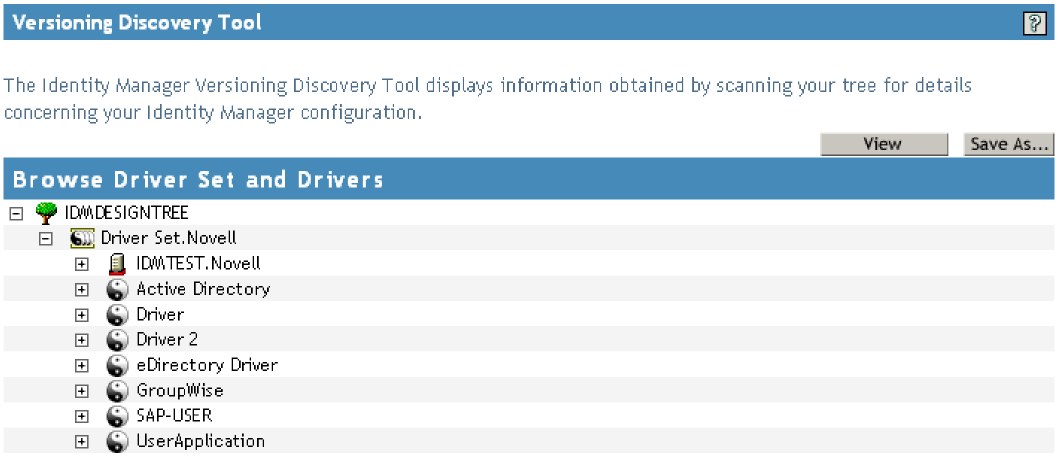
You can save version information to a text file on your local or network drive.

1. In iManager click *Identity Manager* > *Identity Manager Overview*, then click *Search* to find your driver set.
2. Click *Information* in the Identity Manager Overview screen.



You can also select *Identity Manager Utilities > Versioning Discovery*, then browse to and select the driver set, then click *Information*.

3 In the Versioning Discovery Tool dialog box, click *Save As*.



Versioning Discovery Tool

The Identity Manager Versioning Discovery Tool displays information obtained by scanning your tree for details concerning your Identity Manager configuration.

View | Save As...

Browse Driver Set and Drivers

```
⊟ IDMDESIGNTREE
  ⊟ Driver Set.Novell
    ⊞ IDMTEST.Novell
    ⊞ Active Directory
    ⊞ Driver
    ⊞ Driver 2
    ⊞ eDirectory Driver
    ⊞ GroupWise
    ⊞ SAP-USER
    ⊞ UserApplication
```

4 In the File Download dialog box, click *Save*.

5 Navigate to the desired directory, type a filename, then click *Save*.

Identity Manager saves the data to a text file.

2.11 Using Named Passwords

Identity Manager allows you to store multiple passwords securely for a particular driver. This functionality is referred to as Named Passwords. Each different password is accessed by a key, or name.

You can also use the Named Passwords feature to store other pieces of information securely, such as a user name.

To use a Named Password in a driver policy, you refer to it by the name of the password, instead of using the actual password, and the Metadirectory engine sends the password to the driver. The method described in this section for storing and retrieving Named Passwords can be used with any driver without making changes to the driver shim.

NOTE: The sample configurations provided for the Identity Manager Driver for Lotus Notes include an example of using Named Passwords in this way. The Notes driver shim has also been customized to support other ways of using Named Passwords, and examples of those methods are also included. For more information, see the section on Named Passwords in the *Identity Manager Driver Guide for Lotus Notes*.

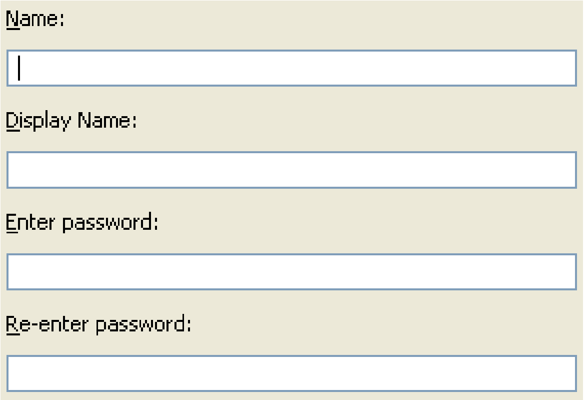
In this section:

- Section 2.11.1, "Configuring Named Passwords by Using Designer," on page 38
- Section 2.11.2, "Configuring Named Passwords by Using iManager," on page 38
- Section 2.11.3, "Using Named Passwords in Driver Policies," on page 40
- Section 2.11.4, "Configuring Named Passwords Using the DirXML Command Line Utility," on page 40

2.11.1 Configuring Named Passwords by Using Designer

1 Select the driver, then right-click and select *Properties*.

2 Select *Named Password*, click *New*.



3 Specify the *Name* of the Named Password.

4 Specify the *Display name* of the Named Password.

5 Specify the Named Password, then re-enter the password.

6 Click *OK*, twice.

2.11.2 Configuring Named Passwords by Using iManager

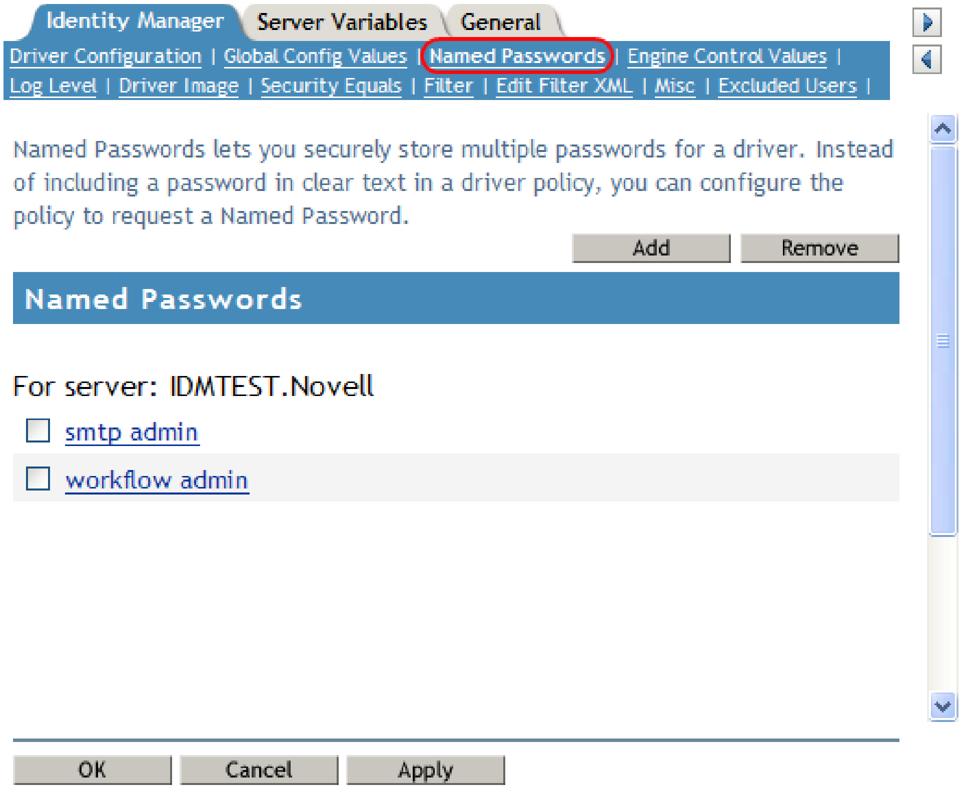
1 In iManager, click *Identity Manager* > *Identity Manager Overview*.

2 Search for the driver set, or browse and select a container that holds the driver set. A graphical representation of the driver set appears.

3. In the Identity Manager Overview screen, click the upper right corner of the driver icon, then click *Edit properties*.

4. On the Modify Object page on the Identity Manager tab, click *Named Passwords*.

   The Named Passwords page appears, listing the current Named Passwords for this driver. If you have not set up any Named Passwords, the list is empty.


Identity Manager
Server Variables
General
Driver Configuration | Global Config Values | Named Passwords | Engine Control Values |
Log Level | Driver Image | Security Equals | Filter | Edit Filter XML | Misc | Excluded Users |
Add
Remove
OK
Cancel
Apply

Named Passwords lets you securely store multiple passwords for a driver. Instead of including a password in clear text in a driver policy, you can configure the policy to request a Named Password.

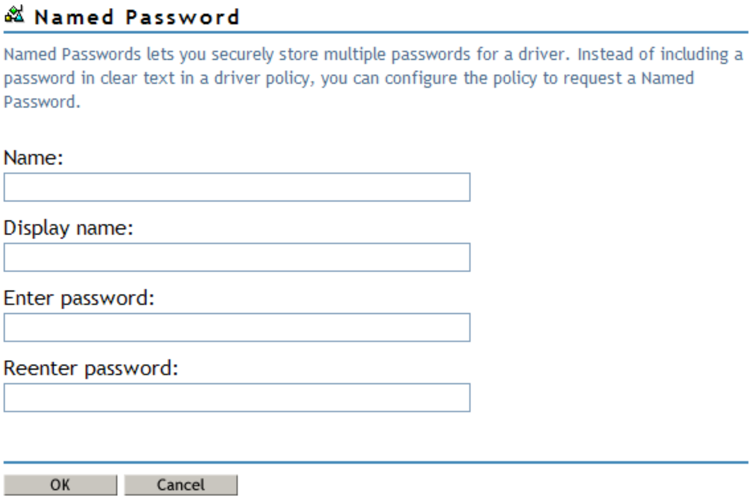
Named Passwords

For server: IDMTEST.Novell

□ smtp admin

□ workflow admin

5. To add a Named Password, click *Add*, complete the fields, and click *OK*.



Named Password

Named Passwords lets you securely store multiple passwords for a driver. Instead of including a password in clear text in a driver policy, you can configure the policy to request a Named Password.

Name:

Display name:

Enter password:

Reenter password:

6. Specify a name, display name and a password, then click *OK* twice.

Keep in mind that you can use this feature to store other kinds of information securely, such as a username.

7 A message is displayed, Do you want to restart the driver to put your changes in effect? (OK=Yes, Cancel=No) click *OK*.

8 To remove a Named Password, click *Remove*. The password is removed without prompting you to confirm the action.

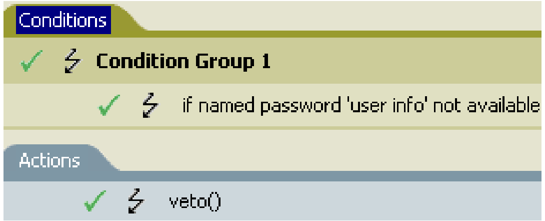
2.11.3 Using Named Passwords in Driver Policies

- "Using the Policy Builder" on page 40
- "Using XSLT" on page 40

Using the Policy Builder

Policy Builder allows you to make a call to a Named Password. Create a new rule and select Named Password as the condition. You set an action depending upon if the Named Password is available or not available. The following example shows if the Named Password userinfo is not available, then the event is vetoed.

Figure 2-4 *A Policy Using Named Password*



Conditions

✓ Condition Group 1

✓ if named password 'user info' not available

Actions

✓ veto()

Using XSLT

The following example shows how a Named Password can be referenced in a driver policy on the Subscriber channel in XSLT:

```
<xsl:value-of
select="query:getNamedPassword($srcQueryProcessor,mynamedpassword)"
xmlns:query="http://www.novell.com/nxsl/java/
com.novell.nds.dirxml.driver.XdsQueryProcessor/>
```

2.11.4 Configuring Named Passwords Using the DirXML Command Line Utility

- "Creating a Named Password in the DirXML Command Line Utility" on page 40
- "Removing a Named Password in the DirXML Command Line Utility" on page 42

Creating a Named Password in the DirXML Command Line Utility

1 Run the DirXML Command Line Utility.

For information, see Appendix A, "DirXML Command Line Utility," on page 307.

2 Enter your user name and password.

The following list of options appears.

```
DirXML commands

 1: Start driver
 2: Stop driver
 3: Driver operations...
 4: Driver set operations...
 5: Log events operations...
 6: Get DirXML version

 7: Job operations...
99: Quit

Enter choice:
```

3. Enter 3 for driver operations.

   A numbered list of drivers appears.

4. Enter the number for the driver you want to add a Named Password to.

   The following list of options appears.

   ```
   Select a driver operation for:
   driver_name

    1: Start driver
    2: Stop driver
    3: Get driver state
    4: Get driver start option
    5: Set driver start option
    6: Resync driver
    7: Migrate from application into DirXML
    8: Submit XDS command document to driver

    9: Submit XDS event document to driver

   10: Queue event for driver
   11: Check object password
   12: Initialize new driver object
   13: Passwords operations
   14: Cache operations
   99: Exit

   Enter choice:
   ```

5. Enter 13 for password operations.

   The following list of options appears.

   ```
   Select a password operation

    1: Set shim password
    2: Reset shim password

    3: Set Remote Loader password

    4: Clear Remote Loader password
    5: Set named password
    6: Clear named password(s)
    7: List named passwords

    8: Get passwords state
   99: Exit

   Enter choice:
   ```

6. Enter 5 to set a new Named Password.

   The following prompt appears:

```
Enter password name:
```

7. Enter the name by which you want to refer to the Named Password.
8. Enter the actual password that you want to secure, at the following prompt that appears:

   ```
   Enter password:
   ```

   The characters you type for the password are not displayed.
9. Confirm the password by entering it again, at the following prompt that appears:

   ```
   Confirm password:
   ```

10. After you enter and confirm the password, you are returned to the password operations menu.

After completing this procedure, you can use the 99 option twice to exit the menu and quit the DirXML Command Line Utility.

Removing a Named Password in the DirXML Command Line Utility

This option is useful if you no longer need Named Passwords you previously created.

1. Run the DirXML Command Line Utility.

   For information, see Appendix A, "DirXML Command Line Utility," on page 307.
2. Enter your user name and password.

   The following list of options appears.

   ```
   DirXML commands

    1: Start driver
    2: Stop driver
    3: Driver operations...
    4: Driver set operations...
    5: Log events operations...
    6: Get DirXML version

    7: Job operations...
   99: Quit

   Enter choice:
   ```

3. Enter 3 for driver operations.

   A numbered list of drivers appears.
4. Enter the number for the driver you want to remove Named Passwords from.

   The following list of options appears.

   ```
   Select a driver operation for:
   driver_name

    1: Start driver
    2: Stop driver
    3: Get driver state
    4: Get driver start option
    5: Set driver start option
    6: Resync driver
    7: Migrate from application into DirXML
    8: Submit XDS command document to driver

    9: Submit XDS event document to driver
   ```

```
10: Queue event for driver
11: Check object password
12: Initialize new driver object
13: Passwords operations
14: Cache operations
99: Exit

Enter choice:
```

5 Enter 13 for password operations.

The following list of options appears.

```
Select a password operation

 1: Set shim password
 2: Reset shim password

 3: Set Remote Loader password

 4: Clear Remote Loader password
 5: Set named password
 6: Clear named password(s)
 7: List named passwords

 8: Get passwords state
99: Exit

Enter choice:
```

6 (Optional) Enter 7 to see the list of existing Named Passwords.

The list of existing Named Passwords is displayed.

This step can help you make sure you are removing the correct password.

7 Enter 6 to remove one or more Named Passwords.

8 Enter No to remove a single Name Password, at the following prompt that appears:

```
Do you want to clear all named passwords? (yes/no):
```

9 Enter the name of the Named Password you want to remove, at the following prompt that appears:

```
Enter password name:
```

After you enter the name of the Named Password you want to remove, you are returned to the password operations menu:

```
Select a password operation

 1: Set shim password
 2: Reset shim password

 3: Set Remote Loader password

 4: Clear Remote Loader password
 5: Set named password
 6: Clear named password(s)
 7: List named passwords

 8: Get passwords state
99: Exit

Enter choice:
```

10 (Optional) Enter 7 to see the list of existing Named Passwords.

The list of existing Named Passwords is displayed.

This step lets you verify that you have removed the correct password.

After completing this procedure, you can use the 99 option twice to exit the menu and quit the DirXML Command Line Utility.

2.12 Reassociating a Driver Object with a Server

A driver object is associated with a server.

If the association becomes invalid for some reason, it is indicated by one of the following:

- When upgrading eDirectory on your Identity Manager server, you get the error "UniqueSPIException error -783."
- No server is listed next to the driver in the Identity Manager Overview screen.
- A server is listed next to the driver in the Identity Manager Overview screen, but the name is garbled text.

To resolve this issue, you must disassociate the driver object and the server, and then reassociate them.

Log into iManager and go to the driver in the Identity Manager Overview screen. Use the icons to remove and then add a server to the server name list next to the driver icon. Removing and then adding re-associates the server and the driver.

2.13 Adding Driver Heartbeat

The driver heartbeat is a feature of the Identity Manager drivers that ship with Identity Manager 2 and above. Its use is optional. Driver heartbeat is configured by using a driver parameter with a time interval specified. If a heartbeat parameter exists and has an interval value other than 0, the driver sends a heartbeat document to the Metadirectory engine if there is no communication on the Publisher channel for the specified interval of time.

The intent of the driver heartbeat is to give you a trigger to allow you to initiate an action at regular intervals, in case the driver does not communicate on the Publisher channel as often as you want the action to occur. You must customize your driver configuration or other tools if you want to take advantage of the heartbeat. The Metadirectory engine accepts the heartbeat document but does not take any action because of it.

For most drivers, a driver parameter for heartbeat is not used in the sample configurations, but you can add it.

A custom driver that is not provided with Identity Manager can also provide a heartbeat document, if the driver developer has written the driver to support it.

To configure the heartbeat, do the following:

1. In iManager, click *Identity Manager* > *Identity Manager Overview*.
2. Browse to and select your driver set, then click *Search*.
3. In Identity Manager Overview screen click the upper right corner of the driver icon, then click *Edit properties*.
4. On the Identity Manager tab, click *Driver Configuration*, scroll down to Driver Parameters, and look for Heart Beat or a similar display name.

   If a driver parameter already exists for heartbeat, you can change the interval and save the changes, and configuration is complete.

The value of the interval cannot be less than 1. A value of 0 means the feature is turned off.

The unit of time is usually minutes; however, some drivers might choose to implement it differently, such as using seconds.

5. If a driver parameter does not exist for heartbeat, click Edit XML.
6. Add a driver parameter entry like the following example, as a child of <publisher-options>. (For an AD driver, make it a child of <driver-options>.)

   ```
   <pub-heartbeat-interval display-name="Heart Beat">10</pub-heartbeat-
   interval>
   ```

   TIP: If the driver does not produce a heartbeat document after being restarted, check the placement of the driver parameter in the XML.

7. Save the changes, and make sure the driver is stopped and restarted.

After you have added the driver parameter, you can edit the time interval using the graphical view. Another option is to create a reference to a global configuration value (GCV) for the time interval. Like other global configuration values, the driver heartbeat can be set at the driver set level instead of on each individual driver object. If a driver does not have a particular global configuration value, and the driver set does have it, the driver inherits the value from the driver set.

The following is an example heartbeat status document sent by the Notes driver:

```
<nds dtdversion="2.0" ndsversion="8.x">
  <source>
    <product build="20031112_1037" instance="blackcap" version="2.0">DirXML
Driver for Lotus Notes</product>
    <contact>Novell, Inc.</contact>
  </source>
  <input>
    <status level="success" type="heartbeat"/>
  </input>
</nds>
```

2.14 Viewing Identity Manager Processes

To view Identity Manage processing events, use DSTRACE. You only use this during testing and troubleshooting Identity Manager. Running DSTRACE while the drivers are in production increases the utilization on the Identity Manager server and can cause events to process very slowly.

To see Identity Manager processes in DSTRACE, you add values to the driver set and the drivers. You can do this in Designer and iManager.

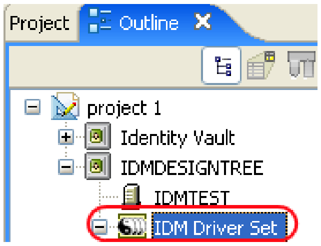
- Section 2.14.1, “Adding Trace Levels in Designer,” on page 45
- Section 2.14.2, “Adding Trace Levels in iManager,” on page 47
- Section 2.14.3, “Capturing Identity Manager Processes to a File,” on page 48

2.14.1 Adding Trace Levels in Designer

You can add trace levels to the driver set or to each driver.

Driver Set

1 In an open project in Designer, select the driver set in the *Outline* view.



2 Right-click and select *Properties*, then click *5. Trace*.

3 Set the parameters for tracing, then click *OK*. See Table 2-1 on page 46 for more information about the driver set trace parameters.

If you set the trace level on the driver set, all drivers appear in the DSTRACE logs.

Table 2-1 *Driver Set Trace Parameters*

Parameter	Description
Driver trace level	As the driver trace level increases, the amount of information displayed in DSTRACE increases. Trace level one shows errors, but not the cause of the errors. If you want to see password synchronization information, set the trace level to five.
XSL trace level	DSTRACE displays XSL events. Only set this trace level when troubleshooting XSL style sheets. If you do not want to see XSL information, set the level to zero.
Java debug port	Allows developers to attach a Java debugger.
Java trace file	When a value is set in this field, all Java information for the driver set is written to a file. The value for this field is the patch for that file. As long as the file is specified, Java information is written to this file. If you do not need to debug Java, leave this field blank.
Trace file size limit	Allows you to set a limit for the Java trace file. If you set the file size to unlimited, the file grows in size until there is no disk space left. **NOTE:** The trace file is created in multiple files. Identity Manager automatically divides the maximum file size by ten and creates ten separate files. The combined size of these files equals the maximum trace file size.

Driver

1. In an open project in Designer, select the driver in the Outline view.
2. Right-click and select *Properties*, then click *8. Trace*.
3. Set the parameters for tracing, then click *OK*. See Table 2-2 on page 47 for more information about these parameters.

   If you set the parameters on the driver only, only information for that driver appears in the DSTRACE log.

Table 2-2 *Driver Trace Parameters*

Parameter	Description
Trace level	As the driver trace level increases, the amount of information displayed in DSTRACE increases. Trace level one shows errors, but not the cause of the errors. If you want to see password synchronization information, set the trace level to five. If you select *Use setting from Driver Set*, the value is taken from the driver set.
Trace file	Specify a file name an location of where the Identity Manager information is written for the selected driver. If you select *Use setting from Driver Set*, the value is taken from the driver set.
Trace file size limit	Allows you to set a limit for the Java trace file. If you set the file size to unlimited, the file grows in size until there is no disk space left. **NOTE:** The trace file is created in multiple files. Identity Manager automatically divides the maximum file size by ten and creates ten separate files. The combined size of these files equals the maximum trace file size. If you select *Use setting from Driver Set*, the value is taken from the driver set.
Trace name	The driver trace messages are prepended with the value entered instead of the driver name. Use if the driver name is very long.

2.14.2 Adding Trace Levels in iManager

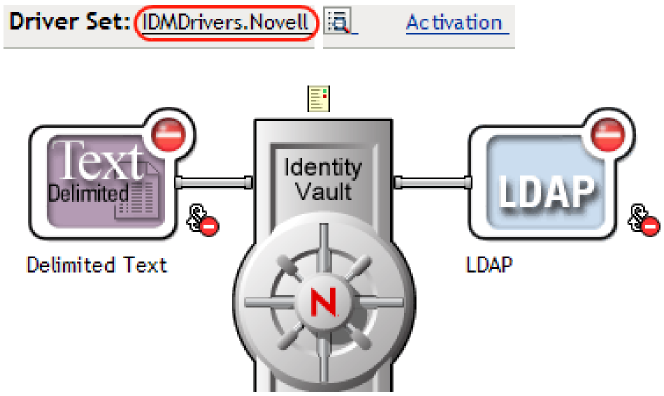
You can add trace levels to the driver set or to each driver.

Driver Set

1. In iManager select *Identity Manager > Identity Manager Overview*.

2 Browse to the driver set, then click *Search*.

3 Click on the driver set name.



4 Select the *Misc* tab for the driver set.

5 Set the parameters for tracing, then click *OK*. See Table 2-1 on page 46 for more information about these parameters.

Driver

1 In iManager select *Identity Manager* > *Identity Manager Overview*.

2 Browse to the driver set where the driver resides, then click *Search*.

3 Click the upper right corner of the driver, then click *Edit properties*.

4 Select the *Misc* tab for the driver.

5 Set the parameters for tracing, then click *OK*. See Table 2-2 on page 47 for more information.

NOTE: The option *Use setting from Driver Set* does not exist in iManager.

2.14.3 Capturing Identity Manager Processes to a File

Identity Manager processes are saved to a file by using a parameter on the driver or through DSTRACE. The parameter on the driver is the Trace file parameter.

The driver processed that are captured through DSTRACE are the processes that occur on the Identity Manager engine. If you use the Remote Loader, you need to capture a trace on the Remote Loader at the same time as you are capturing the trace on the Identity Manager engine.

The following methods helps you capture and save Identity Manager processes through DSTRACE on different OS platforms.

NetWare

Use `DSTRACE.NLM` to display trace messages on the system console or trace messages to a file (`SYS:\SYSTEM\DSTRACE.LOG`). `DSTRACE.NLM` displays the trace messages to a screen labeled DSTRACE Console.

1 Type `DSTRACE.NLM` at the server console.

This loads `DSTRACE.NLM` into memory.

2. Type `DSTRACE SCREEN ON` at the server console.

   Allows trace messages to appear on the DSTRACE Console screen.
3. `Type DSTRACE FILE ON` at the server console.

   Captures trace messages sent to the DSTRACE Console to the `DSTRACE.LOG`.
4. Type `DSTRACE -ALL` at the server console.

   Turns off all trace flags.
5. Type `DSTRACE +DXML DSTRACE +DVRS` at the server console.

   Displays the Identity Manager events.
6. Type `DSTRACE +TAGS DSTRACE +TIME` at the server console

   Displays the message tags and timestamps.
7. Toggle to the DSTRACE Console screen and watch for the event to pass.
8. Toggle back to the server console.
9. Type `DSTRACE FILE OFF` at the server console.

   Stops capture trace messages to the log file. It stops logging the information into the file as well.
10. Open the `DSTRACE.LOG` in a text editor and search for the event or the object you modified.

Windows

1. Open the Control Panel > NDS Services > dstrace.dlm, then click *Start*.

   A window opens named NDS Server Trace Utility.
2. Select *Edit* > *Options*, then click *Clear All*.

   This clears all of the default flags.
3. Select *DirXML* and *DirXML Drivers*.
4. Click OK.
5. Select *File* > *New*.
6. Specify the filename and location of where you want the DSTRACE information saved, then click Open.
7. Wait for the event to occur.
8. Select *File* > *Close*.

   This stops the information from being written to the log file.
9. Open the file in a text editor and search for the event or the object you modified.

UNIX

1. Type `ndstrace` to start the ndstrace utility.
2. Type `set ndstrace=nodebug`

   Turns off all trace flags currently set.
3. Type `set ndstrace on`

   Displays trace messages to the console.
4. Type `set ndstrace file on`

Captures trace messages to the file `ndstrace.log` the directory where eDirectory is installed. By default it is `/var/nds`.

5. Type `set ndstrace=+dxml`

   Displays the Identity Manager events.

6. Type `set ndstrace=+dvrs`

   Displays the Identity Manager driver events.

7. Wait for the event to occur.
8. Type `set ndstrace file off`

   This stops the logging of information to the file.

9. Type `exit` to quite the ndstrace utility.
10. Open the file in a text editor. Search for the event or the object that was modified.

iMonitor

iMonitor allows you to get DSTRACE information from a web browser. It does not matter where Identity Manager is running. These are the files that run iMonitor:

- `NDSIMON.NLM` Runs on NetWare.
- `NDSIMON.DLM` Runs on Windows*.
- `ndsimonitor` Runs on UNIX.

1. Access iMonitor from http://*server_ip*:8008/nds.

   The port of 8008 is the default port.

2. Enter in a user name and password with administrative rights, the click *Login*.
3. Select *Trace Configuration* on the left side.
4. Click *Clear All*.
5. Select DirXML and DirXML Drivers.
6. Click *Trace On*.
7. Select *Trace History* on the left side.
8. Click the document with the Modification Time of Current to see a live trace.
9. Change the *Refresh Interval* if you want to see information more often.
10. Select *Trace Configuration* on the left side, then click *Trace Off* to turn the tracing off.
11. You can view the trace history by selecting Trace History. The files are distinguished by their timestamp.

If you need a copy of the HTML file, the default location is:

- NetWare: `SYS:\SYSTEM\ndsimon\DSTRACE*.htm`
- Windows: `*Drive_letter*:\Novell\NDS\ndsimon\dstrace\*.htm`
- UNIX: `/var/nds/dstrace/*.htm`

Remote Loader

You can capture the events that occur on the machine running the Remote Loader service.

1. Launch the Remote Loader Console by clicking the icon.
2. Select the driver instance, then click *Edit*.
3. Set the *Trace Level* to 3 or above.
4. Specify a location and file for the trace file.
5. Specify the amount of disk space that the file is allowed.
6. Click *OK*, twice to save the changes.

You can also enable tracing from the command line by using the following switches. For more information, see Section 3.4, "Configuring the Remote Loader," on page 65.

Table 2-3 *Command Line Tracing Switches*

Option	Secondary Name	Parameter	Description
-trace	-t	integer	Specifies the trace level. This is only used when hosting an application shim. Trace levels correspond to those used on the Identity Manager server. Example: -trace 3 or -t3
-tracefile	-tf	filename	Specify a file to write trace messages to. Trace messages are written to the file if the trace level is greater than zero. Trace messages are written to the file even if the trace window is not open. Example: -tracefile c:\temp\trace.txt or -tf c:\temp\trace.txt
-tracefilemax	-tfm	size	Specifies the approximate maximum size that trace file data can occupy on disk. If you specify this option, there will be a trace file with the name specified using the tracefile option and up to 9 additional "roll-over" files. The roll-over files are named using the base of the main trace filename plus "_n", where n is 1 through 9. The size parameter is the number of bytes. Specify the size by using the suffixes K, M, or G for kilobytes, megabytes, or gigabytes. If the trace file data is larger than the specified maximum when the Remote Loader is started, the trace file data remains larger than the specified maximum until roll-over is completed through all 10 files. Example: -tracefilemax 1000M or -tfm 1000M

3 Deciding Whether to Use the Remote Loader

Identity Manager has an additional feature that extends Identity Manager functionality across applications. It is called the Remote Loader, and it allows the driver to access the application without having the Identity Vault and the Metadirectory engine installed on the same server as the application. As part of the planning process when installing Identity Manager, you need to decide if you are going to use the Remote Loader or not. This section defines what the Remote Loader is and contains instructions for installing and configuring the Remote Loader.

- Section 3.1, “Overview,” on page 53
- Section 3.2, “Providing for Secure Data Transfers,” on page 55
- Section 3.3, “Installing the Remote Loader,” on page 56
- Section 3.4, “Configuring the Remote Loader,” on page 65
- Section 3.5, “Setting Environment Variables on Solaris, Linux, or AIX,” on page 79
- Section 3.6, “Starting the Remote Loader,” on page 80
- Section 3.7, “Stopping the Remote Loader,” on page 83
- Section 3.8, “Configuring the Identity Manager Drivers for Use with the Remote Loader,” on page 84

3.1 Overview

There are two different ways to configure the installation of the Metadirectory engine. Figure 3-1 illustrates the first way. It shows that the Identity Vault, Metadirectory engine, and the driver shim all are installed and running on the same server. The driver shim is configured to communicate with the application and the Metadirectory engine.

Figure 3-1 *All Components Installed on the Same Server*

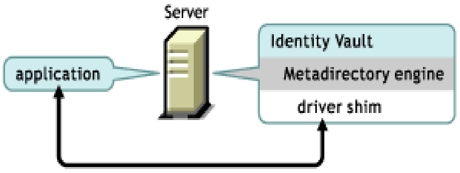
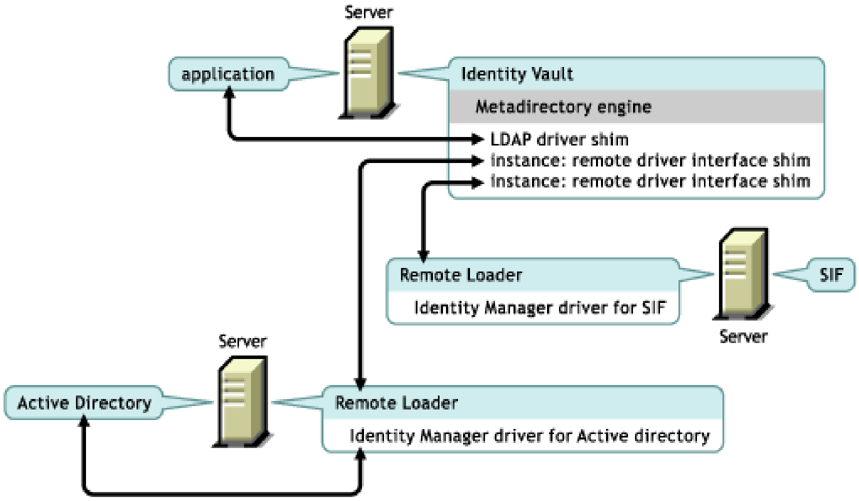


Figure 3-2 illustrates both configurations. The LDAP driver is installed on the same server as the Metadirectory engine and the Identity Vault. The SIF and Active Directory drivers are installed on different servers with the Remote Loader. The Remote Loader allows the driver to access the application without having the Identity Vault and Metadirectory engine installed on that same server.

Figure 3-2 *A System Using the Remote Loader*


Server
application
Identity Vault
Metadirectory engine
LDAP driver shim
instance: remote driver interface shim
instance: remote driver interface shim
Remote Loader
Identity Manager driver for SIF
SIF
Server
Server
Active Directory
Remote Loader
Identity Manager driver for Active directory

The Remote Loader enables the Metadirectory engine to exchange data with the Identity Vault as different processes and in different locations, including the following:

- **As a separate process on the server where the Metadirectory engine is running:** The Metadirectory engine runs as part of an eDirectory™ process. The Identity Manager drivers can run on the server where the Metadirectory engine is running. In fact, they can run as part of the same process as the Metadirectory engine.

  However, for strategic reasons and to simplifying troubleshooting, you might want the Identity Manager driver to run as a separate process on the server.

  If the driver is running as a separate process, the Remote Loader provides a communication channel between the Metadirectory engine and the driver.

- **On a server that is not running the Metadirectory engine:**

  Some of the Identity Manager drivers are unable to run where the Metadirectory engine is running. The Remote Loader enables you to run the Metadirectory engine in one environment while running an Identity Manager driver on a server in a different environment. For example, you cannot run the Active Directory driver on a NetWare® server. The Metadirectory engine can run on the NetWare server while the Remote Loader runs on an Active Directory server.

  - **Scenario: Separate Servers.** The Metadirectory engine is running on a NetWare server. You need to run the Identity Manager Driver for Active Directory. This driver is unable to run on a NetWare server because it must run in an Active Directory environment. You install and run the Remote Loader on a Windows 2003 server. The Remote Loader provides a communication channel between the Active Directory driver and the Metadirectory engine.

  - **Scenario: Non-Host.** The Metadirectory engine is running on Solaris*. You need to communicate with a NIS system where you want to provision user accounts. That system usually doesn't host the Metadirectory engine. You install the Remote Loader and the Identity Manager Driver for NIS on the NIS system. The Remote Loader on the NIS system runs the NIS driver and enables the Metadirectory engine and the NIS driver to exchange data.

Novell® recommends that you use the Remote Loader configuration for use with your drivers where possible. Use the Remote Loader even in cases where the connected system is on the same server as the Metadirectory engine. The following benefits occur by running the driver with the Remote Loader configuration:

- eDirectory is protected from any exceptions encountered by the driver shim.
- It improves the performance of the server running the Metadirectory engine, by offloading driver commands to the remote application or database.
- It allows you to run additional drivers on the server where the Metadirectory engine is not installed.

3.2 Providing for Secure Data Transfers

If you plan to use the Remote Loader, the first step is to provide secure data transfer between the Remote Loader and the Metadirectory engine. This requires that a connection between the Remote Loader and the Metadirectory engine is set up by using the Secure Socket Layer (SSL).

To accomplish this, complete the following tasks:

- Section 3.2.1, "Creating a Server Certificate," on page 55
- Section 3.2.2, "Exporting a Self-Signed Certificate," on page 56

If you are unfamiliar with certificates, it is easy to create a new one.

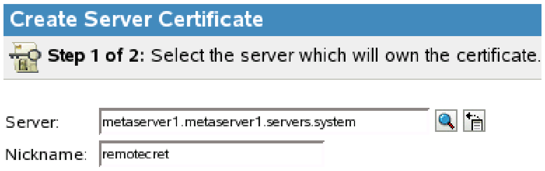
However, if an SSL server certificate already exists and you have experience with SSL certificates, you can use the existing certificate instead of creating and using a new one.

When a server joins a tree, eDirectory creates the following default certificates:

- SSL CertificateIP
- SSL CertificateDNS

3.2.1 Creating a Server Certificate

1 In Novell iManager, click *Novell Certificate Server* > *Create Server Certificate*.



Create Server Certificate

Step 1 of 2: Select the server which will own the certificate.

Server: metaserver1.metaserver1.servers.system

Nickname: remotecret

2 Select the server to own the certificate, and give the certificate a nickname (for example, remotecert).

IMPORTANT: We recommend that you don't use spaces in the certificate nickname. For example, use remotecert instead of remote cert.

Also, make a note of the certificate nickname. You will use this nickname for the KMO name in the driver's remote connection parameters.

3 Leave the Creation method set to *Standard*, then click *Next*.

4 Review the Summary, click *Finish*, then click *Close*.

You have created a server certificate. Continue with Section 3.2.2, “Exporting a Self-Signed Certificate,” on page 56.

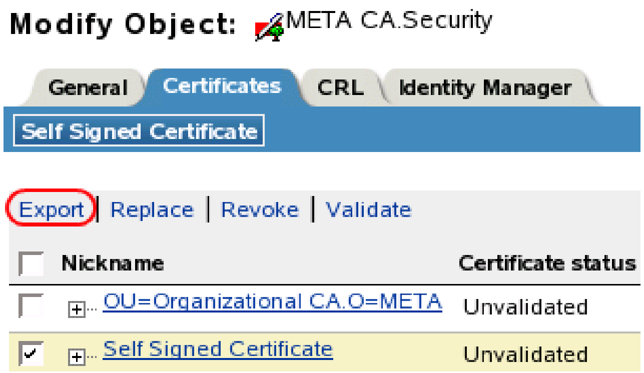
3.2.2 Exporting a Self-Signed Certificate

1 In iManager, click *eDirectory Administration > Modify Object*.

2 Browse to and select the Certificate Authority in the Security container, then click *OK*.

The Certificate Authority (CA) is named after the tree name (Treename-CA.Security).

3 Click the *Certificates* tab, click *Self-Signed Certificate*, then click *Export*.



	Nickname	Certificate status
□	OU=Organizational CA.O=META	Unvalidated
☑	Self Signed Certificate	Unvalidated

4 In the Export Certificate Wizard, select *No*, then click *Next*.

You don’t want to export the private key with the certificate.

5 Select *File in Base64 format* (for example, IDMDESIGNTREE CA.b64), then click *Next*.

Export format: BASE64

IMPORTANT: When the Remote Loader is running on a Windows 2003 R2 SP1 32-bit server, the certificate must be in Base64 format. If you use the DER format, the Remote Loader fails to connect to the Identity Manager engine.

---

6 Click the link to *Save the exported certificate to a file*, specify a filename, specify a location, then click *Save*.

7 In the Save As dialog box, copy this file to a local directory.

8 Click *Close*.

3.3 Installing the Remote Loader

This section provides information on the following:

- Section 3.3.1, “Requirements,” on page 57
- Section 3.3.2, “Supported Drivers,” on page 57

- Section 3.3.3, “Installing a Remote Loader on a Windows Server,” on page 58
- Section 3.3.4, “Installing the Remote Loader on Linux,” on page 60
- Section 3.3.5, “Installing the Remote Loader on UNIX,” on page 62
- Section 3.3.6, “Installing the Java Remote Loader on UNIX, Linux, or AIX,” on page 64
- Section 3.3.7, “Installing a Remote Loader on HP-UX, AS/400, OS/390, or z/OS,” on page 65

3.3.1 Requirements

Each driver requires that the connected system be available and the relevant APIs are provided. Refer to the Identity Manager Driver documentation (http://www.novell.com/documentation/idm35drivers) for operating system and connected system requirements that are specific to each system.

3.3.2 Supported Drivers

Here is a list of the drivers that support the Remote Loader:

- Active Directory*
- Avaya* PBX
- Delimited Text
- Exchange 5.5
- GroupWise®
- JDBC*
- JMS
- LDAP
- Drivers for Linux* and UNIX*
- Lotus Notes*
- NT Domain
- PeopleSoft* 3.7
- PeopleSoft 5.2
- Remedy* ARS
- SAP* HR
- SAP User Management
- Scripting
- SIF*
- SOAP
- WorkOrder
- Manual Task Services
- GenericNull Services
- GenericLoopBack

For additional information about each driver, see the Identity Manager 3.5 Drivers Documentation Web site (http://www.novell.com/documentation/idm35drivers/).

The Remote Loader cannot be used for drivers that are not remote capable. For example

- eDirectory
- Role-Based Entitlement
- User Application

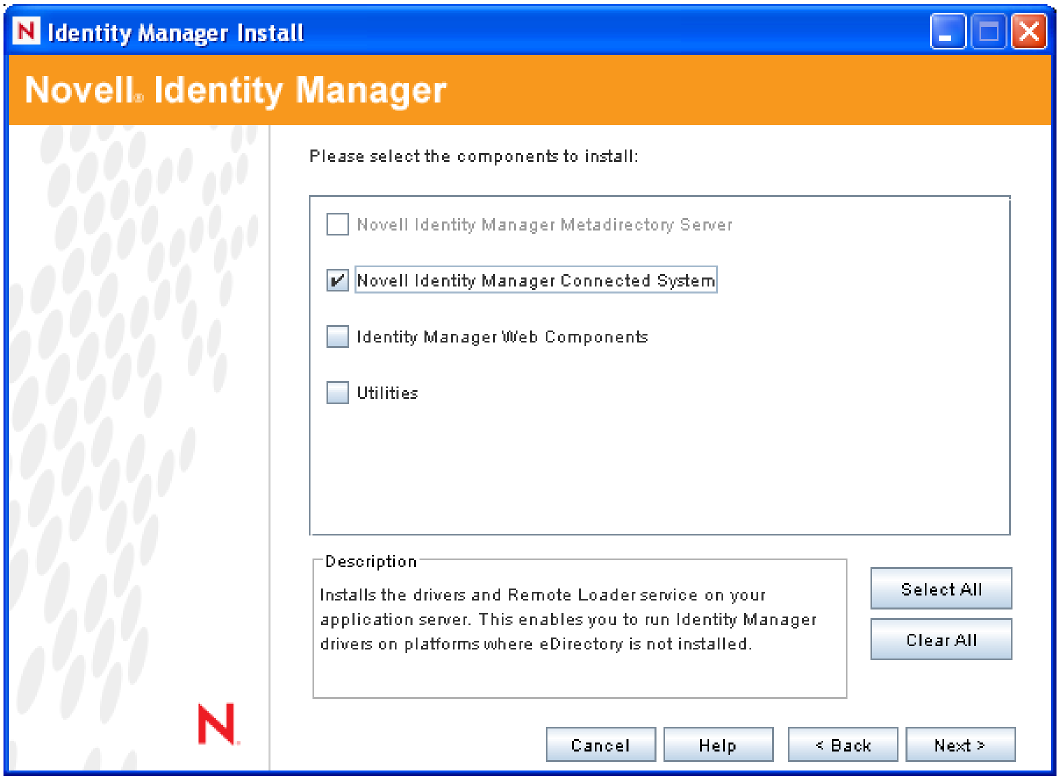
3.3.3 Installing a Remote Loader on a Windows Server

The Remote Loader Console uses `rlconsole.exe` to interface with `dirxml_remote.exe`. `dirxml_remote.exe` is an executable that enables the Metadirectory engine to communicate with the Identity Manager drivers running on Windows.

WARNING: The security of the file system is critical to ensuring the security of the system as a whole. If you are installing the Remote Loader on a Windows 2000 server, verify that directories containing the Remote Loader are accessible only to appropriate administrators. For more information, see TID# 3243550, Securing a Remote Loader Install on a Microsoft Windows 2000 Server (http://www.novell.com/support/search.do?cmd=displayKC&docType=kc&externalId=3243550&sliceId=SAL_Public&dialogID=47824778&stateId=0%200%2047832401).

---

1. Download `Identity_Manager_3_5_1_NW_Win.iso` from the Novell download Web site (http://download.novell.com).
2. Run the Identity Manager installation program for Windows, located at `IDM3.5.1_NW_Win:\nt\install.exe`.
3. Select the language to run the installation program in, then click *OK*.
4. View the Welcome page, then click *Next*.
5. Accept the license agreement, and view the two Overview pages.
6. In the Identity Manager Install dialog box, deselect all components except *Novell Identity Manager Connected System*, then click *Next*.



Please select the components to install:

☐ Novell Identity Manager Metadirectory Server

☑ Novell Identity Manager Connected System

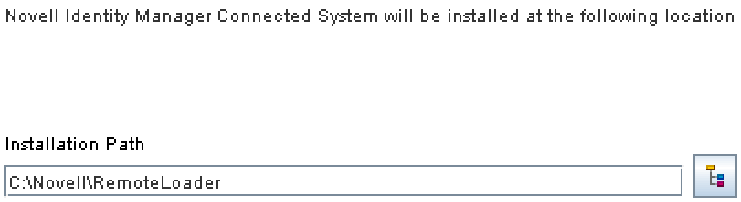
☐ Identity Manager Web Components

☐ Utilities

Description

Installs the drivers and Remote Loader service on your application server. This enables you to run Identity Manager drivers on platforms where eDirectory is not installed.

7 Select a location for the connected system (the Remote Loader and remote driver shims), then click *Next*.

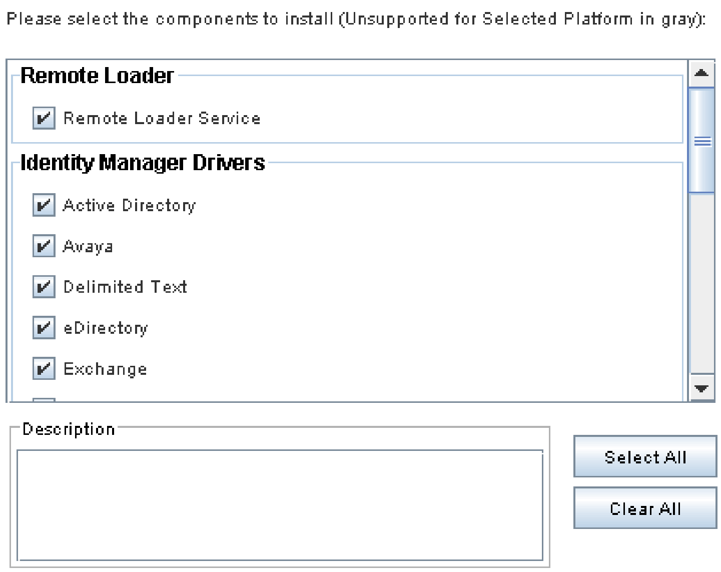


Novell Identity Manager Connected System will be installed at the following location

Installation Path

C:\Novell\RemoteLoader

8 Select the *Remote Loader Service* and remote driver shims (drivers), then click *Next*.



Please select the components to install (Unsupported for Selected Platform in gray):

Remote Loader

☑ Remote Loader Service

Identity Manager Drivers

☑ Active Directory

☑ Avaya

☑ Delimited Text

☑ eDirectory

☑ Exchange

9 Acknowledge the activation requirement, view products to be installed, then click *Finish*.

10 Select whether to place the Remote Loader Console icon on your desktop.

11 Click *Finish* to run the install.

3.3.4 Installing the Remote Loader on Linux

You can install the Remote Loader through a GUI interface or through a command line.

- "GUI Installation of the Remote Loader on Linux" on page 60
- "Command Line Installation of the Remote Loader on LInux" on page 61

GUI Installation of the Remote Loader on Linux

1 Download `Identity_Manager_3_5_1_Linux.iso` from the Novell download Web site (http://download.novell.com).

2 On the host computer, log in as `root`.

3 To run the GUI install on Linux, click the `install.bin` file in the root directory. You are asked if you want to run the install file in terminal mode or in display mode. Select *Terminal*.

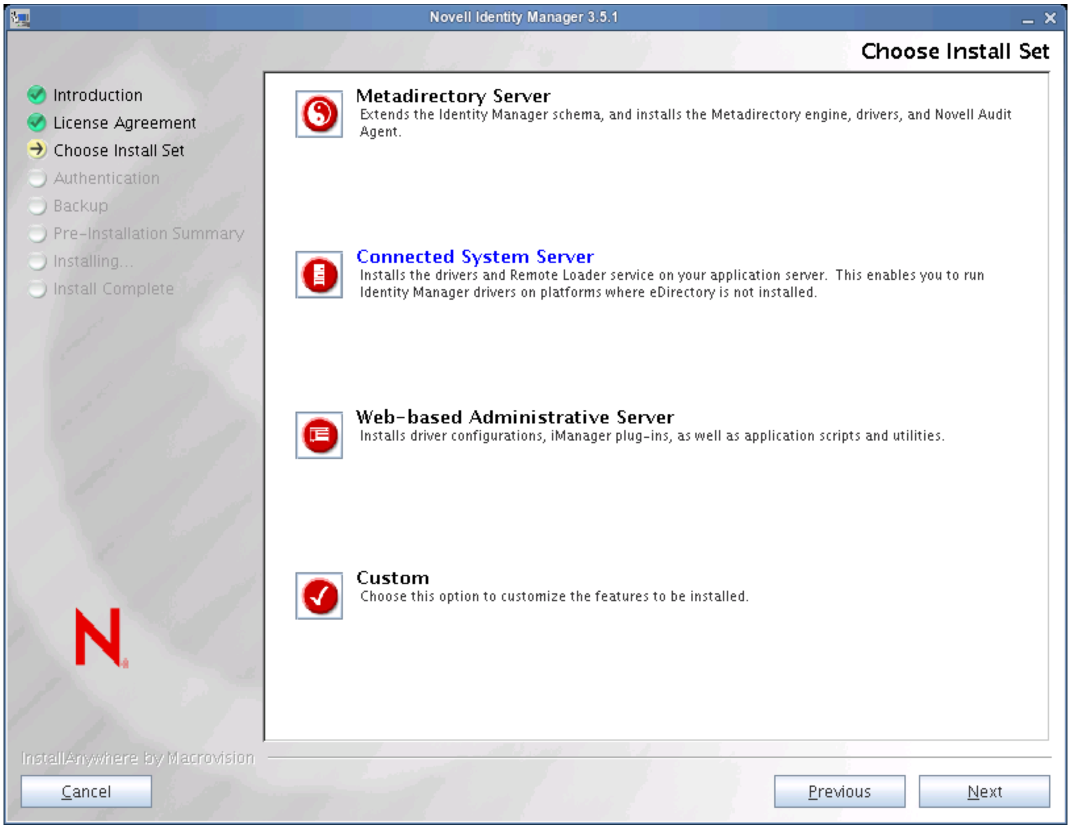
The `install.bin` file checks to see if X windows is present, and if it is, it brings up Identity Manager's GUI install program for Linux.

4 Select the language to run the installation program in, then click *OK*.

5 View the welcome page, then click *Next*.

6 Accept the license agreement, then click *Next*.

7 Click *Connected System Server*, then click *Next*.



Metadirectory Server
Extends the Identity Manager schema, and installs the Metadirectory engine, drivers, and Novell Audit Agent.

Connected System Server
Installs the drivers and Remote Loader service on your application server. This enables you to run Identity Manager drivers on platforms where eDirectory is not installed.

Web-based Administrative Server
Installs driver configurations, iManager plug-ins, as well as application scripts and utilities.

Custom
Choose this option to customize the features to be installed.

8 Acknowledge the activation requirement, view products to be installed, then click *Install*.

9 Read the summary screen, then click *Done*.

Command Line Installation of the Remote Loader on LInux

1 Download `Identity_Manager_3_5_1_Linux.iso` from the Novell download Web site (http://download.novell.com).

2 On the host computer, log in as `root`.

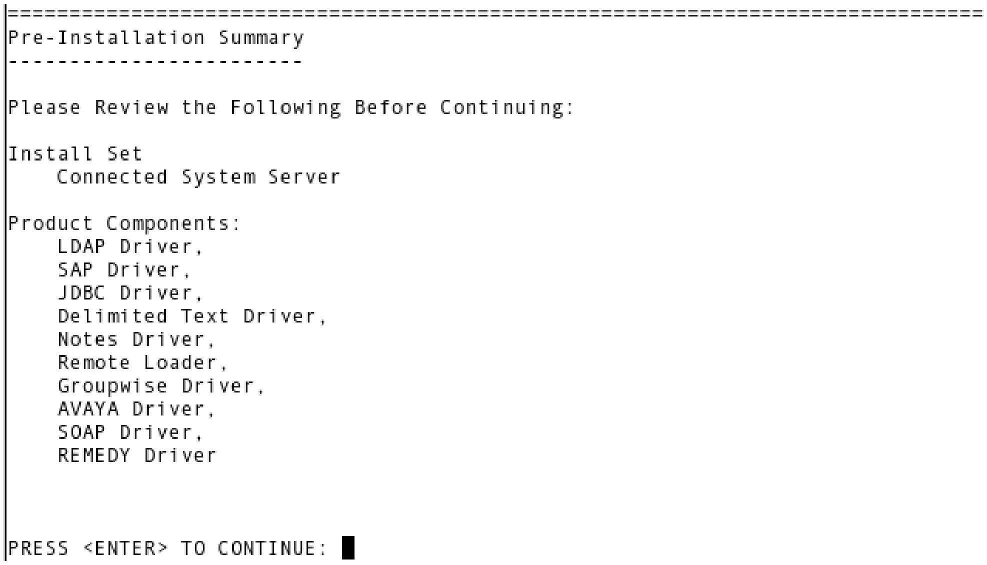
3 Execute the installation file located at `IDM3.5.1_Lin:\linux\setup\idm_linux.bin`.

4 After accepting the license agreement, press Enter to arrive at the Choose Install Set page:

```
===============================================================================
Choose Install Set
------------------

Please choose the Install Set to be installed by this installer.

  ->1- Metadirectory Server
    2- Connected System Server
    3- Web-based Administrative Server

    4- Customize...

ENTER THE NUMBER FOR THE INSTALL SET, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
   :
```

5 Select Connected System Server by typing 2, then press Enter.

6 On the Pre-Installation Summary screen, review components that you have selected to install, then press Enter.



```
===============================================================================
Pre-Installation Summary
------------------------

Please Review the Following Before Continuing:

Install Set
    Connected System Server

Product Components:
    LDAP Driver,
    SAP Driver,
    JDBC Driver,
    Delimited Text Driver,
    Notes Driver,
    Remote Loader,
    Groupwise Driver,
    AVAYA Driver,
    SOAP Driver,
    REMEDY Driver



PRESS <ENTER> TO CONTINUE:
```

3.3.5 Installing the Remote Loader on UNIX

`rdxml` is an executable that enables the Metadirectory engine to communicate with the Identity Manager drivers running in Solaris, Linux, or AIX* environments.

You can install the Remote Loader through a GUI interface or command line.

- "GUI Installation of the Remote Loader on UNIX" on page 62
- "Command Line Installation of the Remote Loader on UNIX" on page 63

GUI Installation of the Remote Loader on UNIX

1 Download `Identity_Manager_3_5_1_Unix.iso` from the Novell download Web site (http://download.novell.com).

2 Run one of the following installation files, depending on your platform:

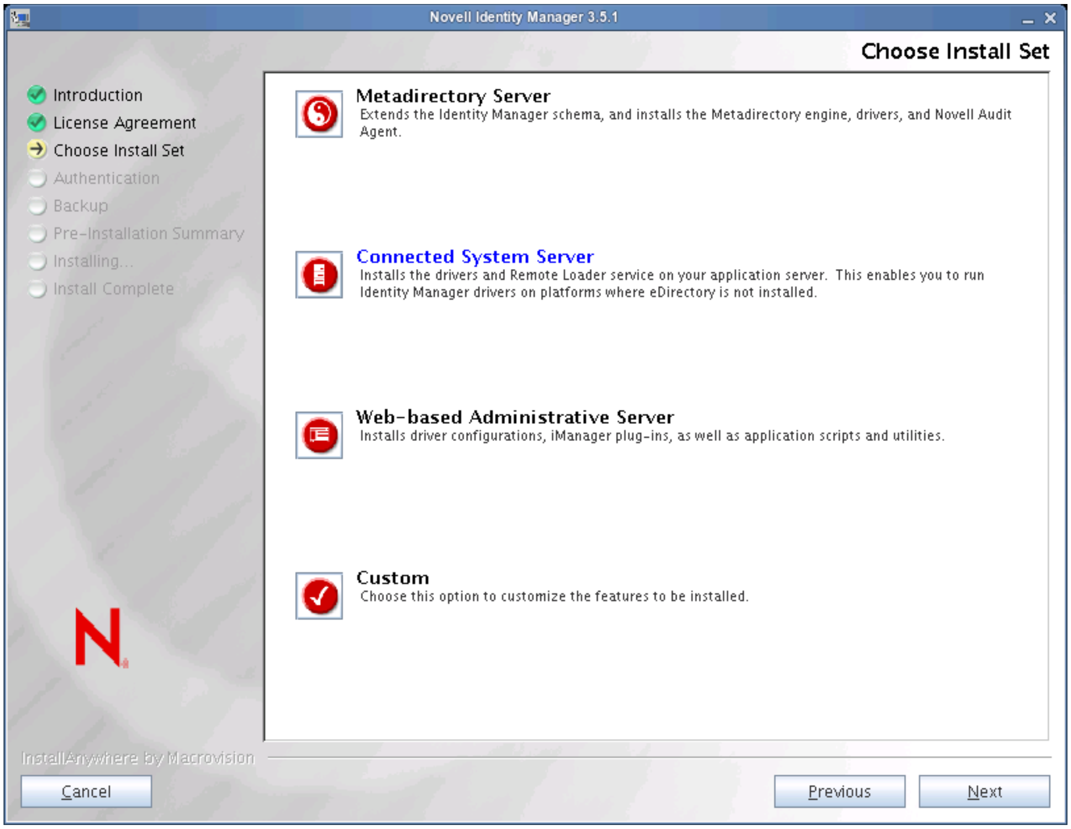
- `IDM3.5.1_Unix:\aix\setup\idm_aix.bin -i gui`
- `IDM3.5.1_Unix:\solaris\setup\idm_solaris.bin -i gui`

3 Select the language to run the installation program in, then click *OK*.

4 View the welcome page, then click *Next*.

5 Accept the license agreement, then click *Next*.

6 Click *Connected System Server*, then click *Next*.



Metadirectory Server
Extends the Identity Manager schema, and installs the Metadirectory engine, drivers, and Novell Audit Agent.

Connected System Server
Installs the drivers and Remote Loader service on your application server. This enables you to run Identity Manager drivers on platforms where eDirectory is not installed.

Web-based Administrative Server
Installs driver configurations, iManager plug-ins, as well as application scripts and utilities.

Custom
Choose this option to customize the features to be installed.

7 Acknowledge the activation requirement, view products to be installed, then click *Install*.

8 Read the summary screen, then click *Done*.

Command Line Installation of the Remote Loader on UNIX

1 Download `Identity_Manager_3_5_1_Unix.iso` from the Novell download Web site (http://download.novell.com).

2 Run one of the following installation files, depending on your platform:

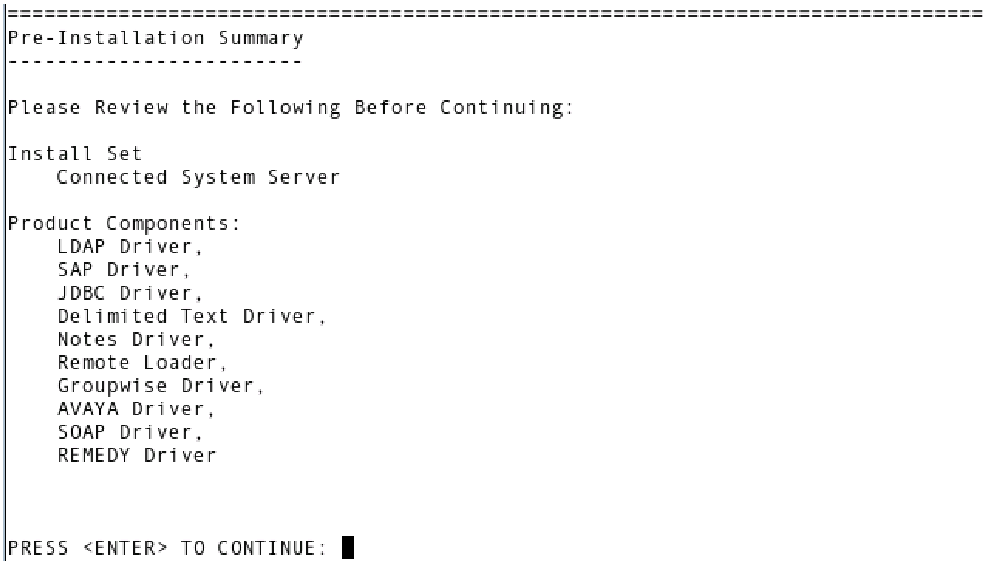
- `IDM3.5.1_Unix:\aix\setup\idm_aix.bin`
- `IDM3.5.1_Unix:\solaris\setup\idm_solaris.bin`

3 After accepting the license agreement, press Enter to arrive at the Choose Install Set page:

```
===============================================================================
Choose Install Set
------------------

Please choose the Install Set to be installed by this installer.

  ->1- Metadirectory Server
    2- Connected System Server
    3- Web-based Administrative Server

    4- Customize...

ENTER THE NUMBER FOR THE INSTALL SET, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
   :
```

4 Select *Connected System Server* by typing 2, then press Enter.

5 On the Pre-Installation Summary screen, review components that you have selected to install, then press Enter.



```
===============================================================================
Pre-Installation Summary
------------------------

Please Review the Following Before Continuing:

Install Set
    Connected System Server

Product Components:
    LDAP Driver,
    SAP Driver,
    JDBC Driver,
    Delimited Text Driver,
    Notes Driver,
    Remote Loader,
    Groupwise Driver,
    AVAYA Driver,
    SOAP Driver,
    REMEDY Driver



PRESS <ENTER> TO CONTINUE:
```

3.3.6 Installing the Java Remote Loader on UNIX, Linux, or AIX

dirxml_jremote is a pure Java Remote Loader. It is used to exchange data between the Metadirectory engine running on one server and the Identity Manager drivers running in another location, where rdxml or dirxml_jremote doesn't run. It should be able to run on any system with a compatible JRE* (1.4.0 minimum, 1.4.2 or higher recommended) and Java Sockets, but is only officially supported on the following:

- HP-UX*
- AS/400*
- OS/390
- z/OS
- plus all supported Metadirectory engine platforms

This section assumes that you have downloaded Identity Manager. If you need to download Identity Manager, go to the Novell download Web site (http://download.novell.com).

1 Verify that Java 1.4.x JDK*/JRE is available on the host system.

2 Copy the dirxml_jremote.tar.gz file to the desired location on the server running the Remote Loader. It is located at IDM_3.5.1_Linux:\java_remoteloader\dirxml_jremote.tar.gz.

For example: /usr/dirxml

3 Unzip and extract dirxml_jremote.tar.gz.

For example: gunzip dirxml_jremote.tar.gz or tar xvf dirxml_jremote.tar

4 Copy the application shim .jar files to the lib subdirectory that was created when the dirxml_jremote.tar file was extracted.

5 Customize the `dirxml_jremote` script by doing either of the following:

- Verify that the Java executable is reachable through the PATH environment variable. For more information, see Section 3.5, "Setting Environment Variables on Solaris, Linux, or AIX," on page 79.
- Edit the `dirxml_jremote` script and prepend the path to the Java executable on the script line that executes Java.

6 Configure the sample `config8000.txt` file for use with your application shim. For more information, see Section 3.4.2, "Configuring the Remote Loader by Creating a Configuration File," on page 69.

3.3.7 Installing a Remote Loader on HP-UX, AS/400, OS/390, or z/OS

The HP-UX, AS/400, OS/390,and z/OS platforms require the Java Remote Loader.

1 Create a directory on the target system where you want to run the Java Remote Loader.

2 From the Identity Manager CD or download image, copy the appropriate file in the `/java_remoteloader` directory to the directory that you created in Step 1:

Platform	File
HP-UX	`dirxml_jremote.tar.gz`
AS/400	`dirxml_jremote.tar.gz`
z/OS	`dirxml_jremote_mvs.tar`
OS/390	`dirxml_jremote_mvs.tar`

3 For HP-UX, AS/400, or z/OS, unzip the `dirxml_jremote` file.

4 Untar the file that you just copied.

The Java Remote Loader is now ready for configuration. Because the tar file doesn't contain drivers, you must manually copy the drivers into the `lib` directory. The `lib` directory is under the directory where the untarring occurred.

For information on MVS, untar the `dirxml_jremote_mvs.tar` file. Then refer to the `usage.html` document.

3.4 Configuring the Remote Loader

The Remote Loader can host the Identity Manager application shims contained in `.dll, .so`, or `.jar` files. The Java Remote Loader hosts only Java driver shims. It won't load or host a native (C++) driver shim.

You can configure the driver on Windows through a graphical utility called the Remote Loader Console utility or from the command line.

- Section 3.4.1, "Configuring the Remote Loader on Windows," on page 66
- Section 3.4.2, "Configuring the Remote Loader by Creating a Configuration File," on page 69

3.4.1 Configuring the Remote Loader on Windows

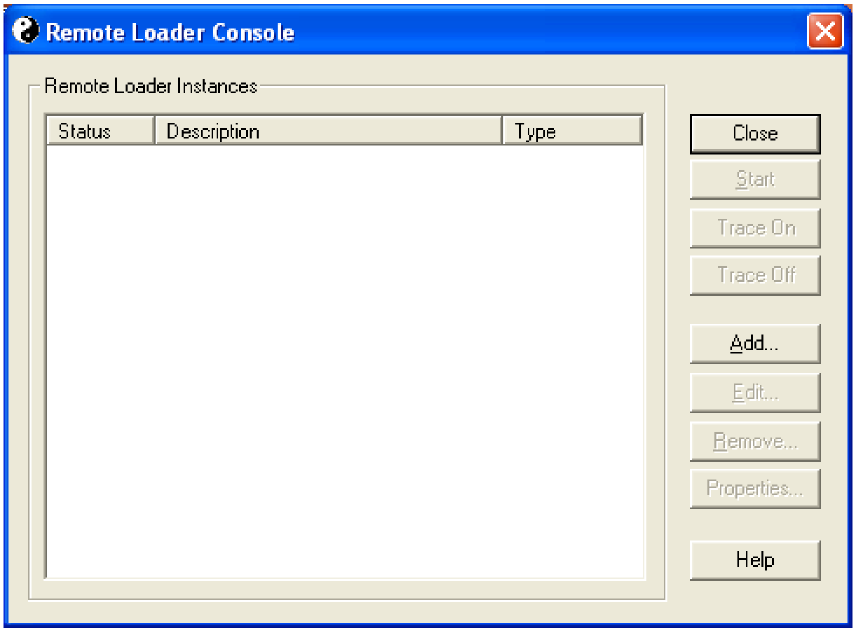
The Remote Loader Console utility enables you to manage all Identity Manager drivers running as a Remote Loader on the Windows server. The utility is installed during the installation of Identity Manager.

If you are upgrading to Identity Manager, the Console detects and imports existing instances of the Remote Loader. (To be automatically imported, driver configurations must be stored in the remote loader directory, typically `c:\novell\remoteloader`.) You can then use the Console to manage the remote drivers.

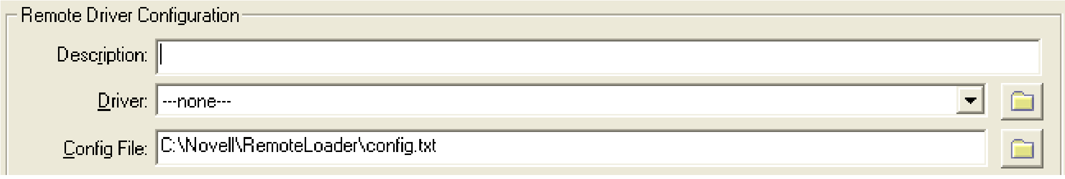
1 Double-click the *Remote Loader Console* icon on the desktop to launch the Remote Loader Console.

The Remote Loader Console allows you to start, stop, add, remove, and edit each instance of a Remote Loader.

2 Click *Add* to add a Remote Loader instance of your driver on this server.



3 Specify the *Remote Driver Configuration* parameters.



Description:

Driver: ---none---

Config File: C:\Novell\RemoteLoader\config.txt

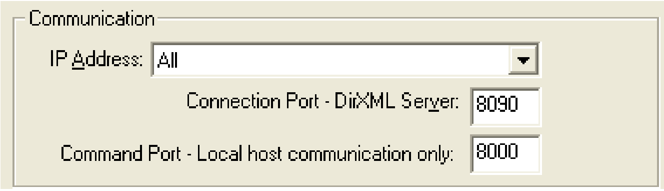
3a Specify a description to identify the Remote Loader instance.

3b Browse to and select the appropriate shim for your driver.

3c Specify a name for the configuration file.

The Remote Loader Console places configuration parameters into this text file and uses those parameters when it runs.

4 Specify the *Communication* parameters.



IP Address: All

Connection Port - DirXML Server: 8090

Command Port - Local host communication only: 8000

4a Specify the IP address where the Remote Loader listens for connections from the Metadirectory server.

4b Specify the TCP port on which the Remote Loader listens for connections from the Metadirectory server.

The default TCP/IP port for this connection is 8090. With each new instance you create, the default port number automatically increases by one.

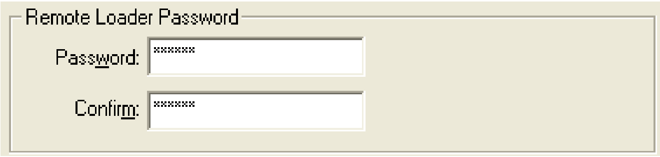
4c Specify the TCP port number where a Remote Loader listens for commands such as `Stop` and `Change Trace Level`.

Each instance of the Remote Loader that runs on a particular computer must have a different command port number. The default command port is 8000. With each new instance you create, the default port number automatically increases by one.

NOTE: By specifying different connection ports and command ports, you can run multiple instances of the Remote Loader on the same server hosting different driver instances.

---

5 Specify the *Remote Loader Password*.

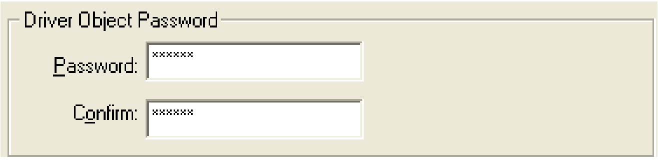


Password: *******

Confirm: *******

This password is used to control access to a Remote Loader instance for a driver. It must be the same case-sensitive password specified in the *Enter the Remote Loader Password* field on the Identity Manager driver configuration page. It is important that this password be difficult to guess and be different from the driver object password.

6 Specify the *Driver Object Password*.



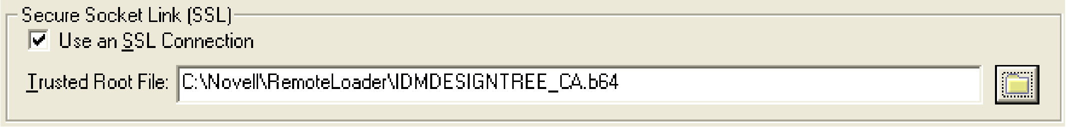
Driver Object Password

Password: ******

Confirm: ******

The Remote Loader uses this password to authenticate to the Metadirectory server. It must be the same case-sensitive password specified in the *Driver Object Password* field on the Identity Manager driver configuration page. It is important that this password be difficult to guess and be different from the Remote Loader password.

7 Specify the *Secure Socket Link* parameters.



Secure Socket Link (SSL)

☑ Use an SSL Connection

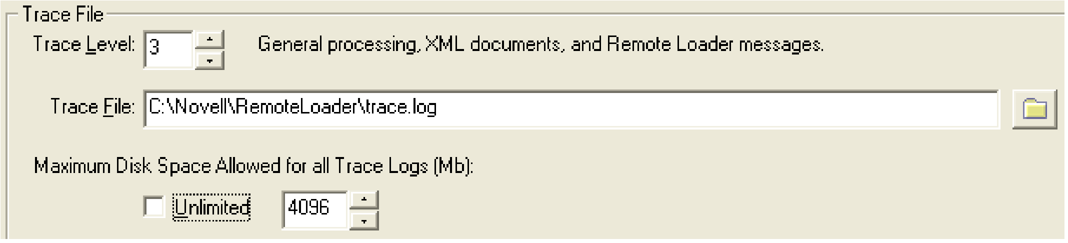
Trusted Root File: C:\Novell\RemoteLoader\IDMDESIGNTREE_CA.b64

7a Select *Use an SSL connect,* if you are encrypting the transfer of data between the Remote Loader and the Metadirectory server.

7b Browse to and select a trusted root file.

This is the exported self-signed certificate from the eDirectory tree's Organization Certificate Authority. See Section 3.2.2, "Exporting a Self-Signed Certificate," on page 56.

8 Specify *Trace File* parameters.



Trace File

Trace Level: 3 General processing, XML documents, and Remote Loader messages.

Trace File: C:\Novell\RemoteLoader\trace.log

Maximum Disk Space Allowed for all Trace Logs (Mb):

☐ Unlimited 4096

8a Specify a trace level greater than zero to display a trace window that contains informational messages from both the Remote Loader and the driver.

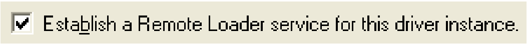
The most common setting is trace level 3. If the trace level is set to 0, the trace window is not displayed.

8b Specify a trace filename where trace messages are written.

Each Remote Loader instance running on a particular machine must use a different trace file. Trace messages are written to the trace file only if the trace level is greater than zero.

8c Specify the approximate maximum size that the trace file for this instance can occupy on disk.

9 Select *Establish a Remote Loader service for this driver instance* if you want the Remote Loader as a service.



☑ Establish a Remote Loader service for this driver instance.

When this option is enabled, the operating system automatically starts the Remote Loader when the computer starts.

10 Click *OK*, to save the configuration information.

If you need to change any of the parameters:

1. In the Remote Loader Console, select the Remote Loader instance from the *Description* column.
2. Click *Stop*, type the Remote Loader password, then click *OK*.
3. Click *Edit*, then modify the configuration information. These are the same fields you filled in when you add a Remote Loader instance.
4. Click *OK*, to save the changes.

3.4.2 Configuring the Remote Loader by Creating a Configuration File

For the Remote Loader to run, it requires a configuration file (for example, `LDAPShim.txt`). Windows is the only platform that provides a GUI interface to create this file. You can create or edit a configuration file by using command line options. The following steps provide information on basic parameters for the configuration file. Appendix B, "Options for Configuring a Remote Loader," on page 321.

1. To create a configuration file, open a text editor.
2. (Optional) Specify a description by using the `-description` option.

Option	Secondary Name	Parameter	Description

-description	-desc	short description	Specify a short description string (for example, SAP) to be used for the trace window title and for Novell Audit logging.
			Example:
			`-description SAP`
			`-desc SAP`
			The Remote Loader Console places long forms in the configuration files. You can use either a long form (for example, -description) or a short form (for example, -desc).

3. Specify a TCP/IP port that the Remote Loader instance will use by using the `-commandport` option.

Option	Secondary Name	Parameter	Description

-commandport	-cp	port number	Specifies the TCP/IP port that the Remote Loader instance uses for control purposes. If the Remote Loader instance is hosting an application shim, the command port is the port on which another Remote Loader instance communicates with the instance that is hosting the shim. If the Remote Loader instance is sending a command to an instance that is hosting an application shim, the command port is the port on which the hosting instance is listening. If not specified, the default command port is 8000. Multiple instances of the Remote Loader can run on the same server hosting different driver instances by specifying different connection ports and command ports.
			Example:
			`-commandport 8001`
			`-cp 8001`

4 Specify the parameters for connecting to the Metadirectory server running the Identity Manager remote interface shim by using the `-connection` option.

The following table describes the `-connection` option:

Option	Secondary Name	Parameter	Description

- connection	- conn	connection configuration string	Specifies the connection parameters for the connection to the Metadirectory server running the Identity Manager remote interface shim. The default connection method for the Remote Loader is TCP/IP using SSL. The default TCP/IP port for this connection is 8090. Multiple instances of the Remote Loader can run on the same server. Each instance of the Remote Loader hosts a separate Identity Manager application shim instance. Differentiate multiple instances of the Remote Loader by specifying different connection ports and command ports for each Remote Loader instance.
			Example:
			`-connection “port=8091 rootfile=server1.pem”`
			`-conn “port=8091 rootfile=server1.pem”`

Type -connection “*parameter* [parameter]”.

The following table describes the parameters for the `-connection` option and their supported platforms:

Parameter	Example	Description	Platform
port	`port=8090`	A decimal port number. A required parameter. It specifies the TCP/IP port on which the Remote Loader listens for connections from the remote interface shim.	Windows and Unix

Parameter	Example	Description	Platform

address	`address=137.65.134.83`	The IP address. An optional parameter. Specifies that the Remote Loader listens on a particular local IP address. This is useful if the server hosting the Remote Loader has multiple IP addresses and the Remote Loader must listen on only one of the addresses.
		You have two options:
		`address='ip address'`
		`address='localhost'`
		If you don't use the address, the Remote Loader listens on all local IP addresses.
			Windows and Unix

fromaddress	`-connect "port=8094 fromaddress=10.0.0.2"`	The Remote Loader only accepts connections from the specified IP address. Any other connections are not allowed.	Windows and Unix
handshaketimeout	`-connection "port=8091 handshaketimeout=1000"`	Increases the time out period of the handshake between the Remote Loader and the Metadirectory engine. The value is some integer greater than or equal to zero. Zero means never time out. The non-zero number is the number of milliseconds for the time out to occur. The default value is 1000 milliseconds.	Windows and Unix

Parameter	Example	Description	Platform
rootfile	`rootfile='`*`trusted certname`*`'`	A conditional parameter. If you are running SSL and need the Remote Loader to communicate with a native driver, type. This parameter is valid only for native drivers. You have the option of specifying the absolute path. If the absolute path is not specified,use the following guidelines to choose the directory: ◆ On a Windows platform, use the directory in which `dirxml_remote.exe` is found. ◆ On a Unix platform, specify the directory with the `-datadir` parameter or with the `RDXMLPATH` environment variable. If you do not specify a directory, the default directory `/var/opt/novell/dirxml/rdxml` is used.	Windows and Unix

Parameter	Example	Description	Platform
keystore	`keystore='`*`keystorename`*`'`	A conditional parameter. Used only for the Identity Manager application shims contained in `.jar` files. Specifies the filename of the Java keystore that contains the trusted root certificate of the issuer of the certificate used by the remote interface shim. This is typically the Certificate Authority of the eDirectory tree that is hosting the remote interface shim. If you are running SSL and need the Remote Loader to communicate with a Java driver, type a key-value pair. You have the option of specifying the absolute path. If the absolute path is not specified,use the following guidelines to choose the directory: ◆ On a Windows platform, use the directory in which `dirxml_remote.exe` is found. ◆ On a Unix platform, specify the directory with the `-datadir` parameter or with the `RDXMLPATH` environment variable. If you do not specify a directory, the default directory /var/opt/novell/dirxml/rdxml is used. **NOTE:** For more information on creating the keystore option, see Section 3.8.3, “Creating a Keystore,” on page 87	Windows and Unix

Parameter	Example	Description	Platform
storepass	`storepass='dirxml'`	Used only for the Identity Manager application shims contained in `.jar` files. Specifies the password for the Java keystore specified by the keystore parameter. This option applies only to the Java Remote Loader.	Unix
keypass	`keypass='mypassword'`	Used only for the Identiy Manager application shims contained in `.jar` files.Specifies the password for the Java keystore specified by the keystore parameter.	Windows
fromaddress	*137.65.134.84*	Specifies that the remote loader will only accept connections from the specified IP address.	Windows and Unix
handshaketimeout	*1500*	Specifies the 'handshake timeout' value for connections to the remote loader. If the SSL handshake and password exchange handshake do not complete within this period after the initial TCP connection, the remote loader closes the connection. **NOTE:** The default value of 1000 (1 second) should only be changed when handshake timeouts occur with otherwise valid connections from the Identity Manager engine.	Windows and Unix

5 (Optional) Specify a trace parameter by using the `-trace` option.

Option	Secondary Name	Parameter	Description

-trace	-t	integer	Specifies the trace level. This is only used when hosting an application shim. Trace levels correspond to those used on the Metadirectory server.
			Example:
			`-trace 3`
			`-t 3`

6 (Optional) Specify a tracefile by using the `-tracefile` option.

Option	Secondary Name	Parameter	Description

-tracefile	-tf	filename	Specify a file to write trace messages to. Trace messages are written to the file if the trace level is greater than zero. Trace messages are written to the file even if the trace window is not open.
			Example:
			`-tracefile c:\temp\trace.txt`
			`-tf c:\temp\trace.txt`

7 (Optional) Limit the size of the tracefile by using the -tracefilemax option.

Option	Secondary Name	Parameter	Description

-tracefilemax	-tfm	size	Specifies the approximate maximum size that trace file data can occupy on disk. If you specify this option, there will be a trace file with the name specified using the tracefile option and up to 9 additional “roll-over” files. The roll-over files are named using the base of the main trace filename plus “_*n*”, where n is 1 through 9.
			The size parameter is the number of bytes. Specify the size by using the suffixes K, M, or G for kilobytes, megabytes, or gigabytes.
			If the trace file data is larger than the specified maximum when the Remote Loader is started, the trace file data remains larger than the specified maximum until roll-over is completed through all 10 files
			Example:
			`-tracefilemax 1000M`
			`-tfm 1000M`
			In this example, the trace file can be only 1 GB.

8 Specify the class by using the -class option, or the module by using the `-module` option.

Option	Secondary Name	Parameter	Description

-class	-cl	Java class name	Specifies the Java class name of the Identity Manager application shim that is to be hosted.
			For example, for a Java driver, type one of the following:
			`-class com.novell.nds.dirxml.driver.ldap.LDAPDriverShim`
			`-cl com.novell.nds.dirxml.driver.ldap.LDAPDriverShim`
			Java uses a keystore to read certificates. The `-class` option and the `-module` option are mutually exclusive.
			To see a list of the Java class name see Table B-2 on page 328 in Appendix B, “Options for Configuring a Remote Loader,” on page 321.

-module	-m	modulename	Specifies the module containing the Identity Manager application shim that is to be hosted.
			For example, for a native driver, type one of the following:
			`-module "c:\Novell\RemoteLoader\Exchange5Shim.dll"`
			`-m "c:\Novell\RemoteLoader\Exchange5Shim.dll"`
			or
			`-module "/usr/lib/dirxml/NISDriverShim.so"`
			`-m "/usr/lib/dirxml/NISDriverShim.so"`
			The `-module` option uses a rootfile certificate. The `-module` option and the `-class` option are mutually exclusive.

9 Name and save the file.

You can change some settings while the Remote Loader is running, see Table 3-1 for a list of some of these settings. For a complete list of these settings, see Appendix B, “Options for Configuring a Remote Loader,” on page 321.

***Table 3-1** Selected Remote Loader Parameters*

Parameter	Description
-commandport	Specifies an instance of the Remote Loader.

Parameter	Description
-config	Specifies a configuration file.
-javadebugport	Specifies that the Remote Loader instance is to enable Java debugging on the specified port.
-password	Specifies the password for authentication.
-service	Installs an instance as a service. Windows only.
-tracechange	Changes the trace level.
-tracefilechange	Changes the name of the trace file being written to.
-unload	Unloads the Remote Loader instance.
-window	Turns the trace window on or off in a Remote Loader instance. Windows only.

The following sample configuration files show various options that can be configured.

On a Windows platform:

```
-description "Sample1"
-commandport 8015
-connection "port=8111"
-trace 4
-tracefile "C:\Novell\RemoteLoader\Sample1-Trace.log"
-tracefilemax 25M
-class"com.novell.nds.dirxml.driver.ldap.LDAPDriverShim"
```

On a Unix platform:

```
-description "Sample1"
-commandport 8015
-connection "port=8111"
-trace 4
-tracefile "/tmp/trace.txt"
-tracefilemax 25M
-class  "com.novell.nds.dirxml.driver.ldap.LDAPDriverShim"
-module "$DXML_HOME/dirxmlremote/libcskeldrv.so.0.0.0"
```

3.5 Setting Environment Variables on Solaris, Linux, or AIX

After installing the Remote Loader, you can set the environment variable `RDXML_PATH`, which changes the current directory for rdxml. This directory is then taken as the base path for files that are subsequently created. To set the value of the `RDXML_PATH` variable, specify the following commands:

- `set RDXML_PATH=path`
- `export RDXML_PATH`

3.6 Starting the Remote Loader

Each platform has a different way to start the Remote Loader.

- Section 3.6.1, "Starting the Remote Loader on Windows," on page 80
- Section 3.6.2, "Auto-starting the Remote Loader," on page 82
- Section 3.6.3, "Starting the Remote Loader on Solaris, Linux, or AIX," on page 82

3.6.1 Starting the Remote Loader on Windows

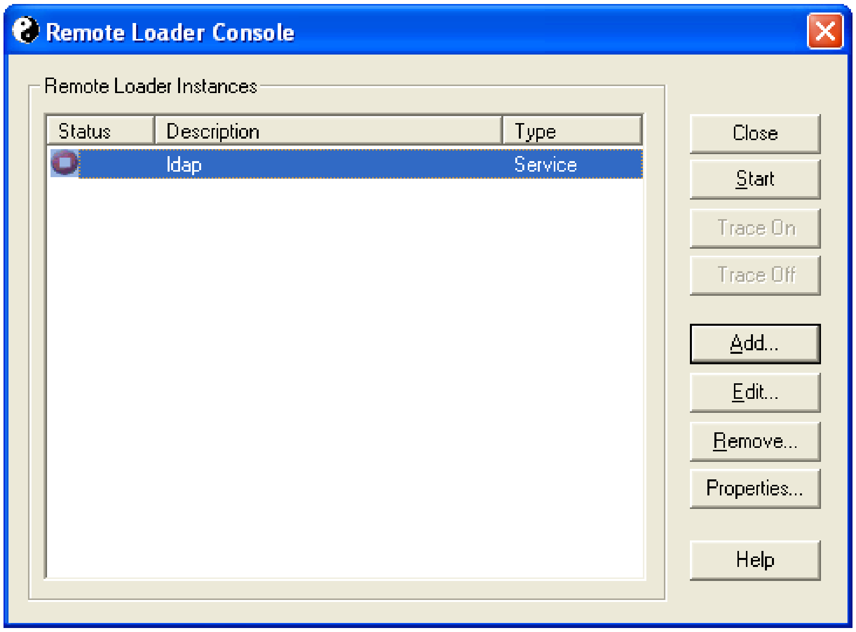
You can start the Remote Loader from the Remote Loader Console icon or from the command line.

- "Starting from the Remote Loader Console" on page 80
- "Starting from the Command Line in Windows" on page 81

Starting from the Remote Loader Console

1 Click the *Remote Loader Console* icon on the desktop.

2 Select a driver instance, then click *Start*.



Status	Description	Type
●	ldap	Service

Starting from the Command Line in Windows

The command line functionality is provided by `dirxml_remote.exe`. By default, it is located in `c:\novell\RemoteLoader\dirxml_remote.exe`. At a command prompt:

1. Set the password for the Remote Loader. For password command options, see Table 3-2 on page 81.

   ```
   dirxml_remote -config path_to_config_file -sp password password
   ```

2. Start the Remote Loader.

   ```
   dirxml_remote -config path_to_config_file
   ```

3. Use iManager to start the driver.
4. Confirm that the Remote Loader is working properly.

   The Remote Loader loads the Identity Manager application shim only when the Remote Loader is in communication with the remote interface shim on the Metadirectory server. This means, for example, that the application shim shuts down if the Remote Loader loses communication with the Metadirectory server.

Table 3-2 *Password Command Line Options*

Option	Secondary Name	Parameter	Description

-password	-p	password	Specifies the password for command authentication. This password must be the same as the first password specified with `setpasswords` for the loader instance being commanded. If a command option (for example, unload or tracechange) is specified and the `password` option isn't specified, the user is prompted to enter the password for the loader that is the target of the command.
			Example:
			`-password novell4`
			`-p novell4`

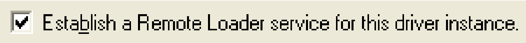
Option	Secondary Name	Parameter	Description

-setpasswords	-sp	password
		password
			Specifies the password for the Remote Loader instance and the password of the Identity Manager Driver object of the remote interface shim that the Remote Loader communicates with. The first password in the argument is the password for the Remote Loader. The second password in the optional arguments is the password for the Identity Manager Driver object associated with the remote interface shim on the Metadirectory server. Either no password or both passwords must be specified. If no password is specified, the Remote Loader prompts for the passwords. This is a configuration option. Using this option configures the Remote Loader instance with the passwords specified but doesn't load an Identity Manager application shim or communicate with another loader instance.
			Example:
			`-setpasswords novell4 staccato3`
			`-sp novell4 staccato3`

3.6.2 Auto-starting the Remote Loader

To auto-start the Remote Loader on a Windows platform, see Step 9 in Section 3.4.1, "Configuring the Remote Loader on Windows," on page 66

Select *Establish a Remote Loader service for this driver instance* if you want the Remote Loader as a service.



☑ Establish a Remote Loader service for this driver instance.

When this option is enabled, the operating system automatically starts the Remote Loader when the computer starts.

To auto-start the Remote Loader on a Linux platform, place your configuration file in the following location: `etc/opt/novell/dirxml/rdxml`. Your Remote Loader instance will start automatically, when the computer starts.

3.6.3 Starting the Remote Loader on Solaris, Linux, or AIX

On Solaris, Linux, or AIX, the binary component rdxml provides the Remote Loader functionality. The default location of this component is in the `/usr/bin/` directory.

1 Set the password for the Remote Loader. For command password options, see Table 3-2 on page 81.

Platform	Command
Solaris LInux AIX	`rdxml` -config *path_to_config_file* -sp *password password*
HP-UX AS/400 OS/390 z/OS	`dirxml_jremote` -config *path_to_config_file* -sp *password password*

2 Start the Remote Loader.

Platform	Command
Solaris LInux AIX	`rdxml` -config *path_to_config_file*
HP-UX AS/400 OS/390 z/OS	`dirxml_jremote` -config *path_to_config_file*

3 Use iManager to start the driver.

4 Confirm that the Remote Loader is operating properly.

The Remote Loader loads the Identity Manager application shim only when the Remote Loader is in communication with the remote interface shim on the Metadirectory server. This means, for example, that the application shim shuts down if the Remote Loader loses communication with the Metadirectory server.

For Linux, Solaris, or AIX, use the `ps` command or a trace file to find out whether the command and connection ports are listening.

For HP-UX and similar platforms, monitor the Java Remote Loader by using the `tail` command on the tracefile:

```
tail -f trace filename
```

If the last line of the log shows the following, the loader is successfully running and awaiting connection from the Identity Manager remote interface shim:

```
TRACE: Remote Loader: Entering listener accept()
```

To configure the Remote Loader (rdxml) to start automatically on UNIX, see TID 10097249 (http://support.novell.com/cgi-bin/search/searchtid.cgi?/10097249.htm).

3.7 Stopping the Remote Loader

Each platform has a different way to stop the Remote Loader. Table 3-3 contains the instructions for each platform.

Table 3-3 *How to Stop the Remote Loader*

Platform	Command
Windows	Use the Remote Loader Console to stop a driver instance.
Solaris LInux AIX	`rdxml -config` *`path_to_config_file`* `-u`
HP-UX AS/400 OS/390 z/OS	`dirxml_jremote -config` *`path_to_config_file`* `-u`

If multiple instances of the Remote Loader are running on the computer, pass the `-cp` *command port* option so that the Remote Loader can stop the appropriate instance.

When you stop the Remote Loader, you must have sufficient rights or specify the Remote Loader password. For example, the Remote Loader is running as a Windows service. You have sufficient rights to stop it. You enter a password, but realize that it is incorrect. The Remote Loader stops anyway.

The Remote Loader isn't "accepting" the password. Instead, it is ignoring the password because the password is redundant in this case. If you run the Remote Loader as an application rather than as a service, the password is used.

3.8 Configuring the Identity Manager Drivers for Use with the Remote Loader

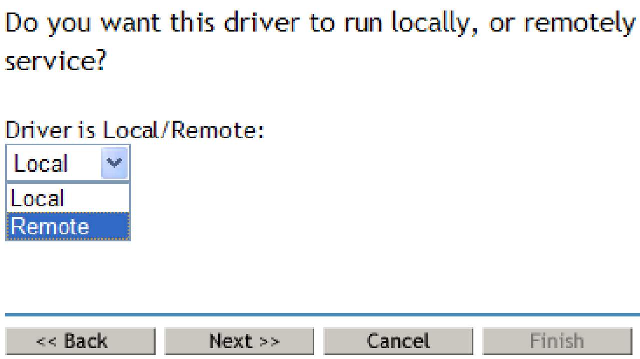
You can configure a new driver or enable an existing driver to communicate with the Remote Loader. This section provides general information on configuring drivers so that they communicate with the Remote Loader. For driver-specific information, refer to the relevant driver implementation guide.

- Section 3.8.1, "Importing and Configuring a New Driver," on page 84
- Section 3.8.2, "Configuring an Existing Driver," on page 86
- Section 3.8.3, "Creating a Keystore," on page 87

3.8.1 Importing and Configuring a New Driver

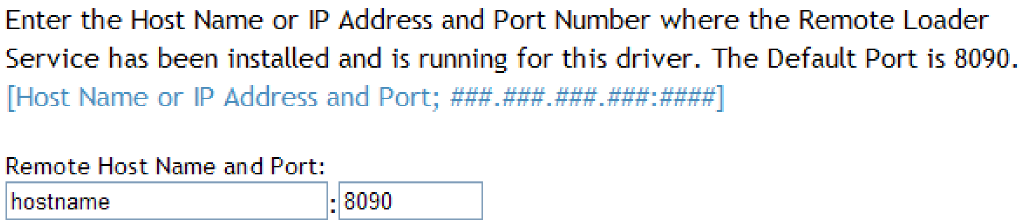
1 In iManager, import or create and configure a new driver.

2 Scroll to the bottom of the configuration options, select Remote from the drop-down list, then click *Next*.


Do you want this driver to run locally, or remotely with the Remote Loader service?

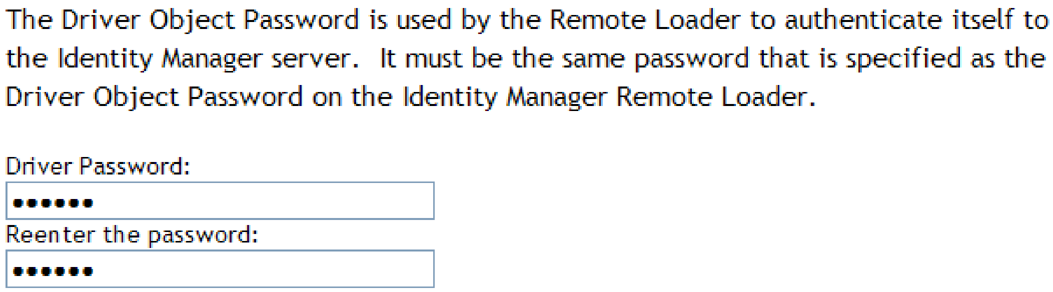
Driver is Local/Remote: Local

3 Specify a remote hostname and port.


Enter the Host Name or IP Address and Port Number where the Remote Loader Service has been installed and is running for this driver. The Default Port is 8090. [Host Name or IP Address and Port; ###.###.###.###:####]

Remote Host Name and Port: hostname : 8090

4 Type and re-enter a password for the driver password.

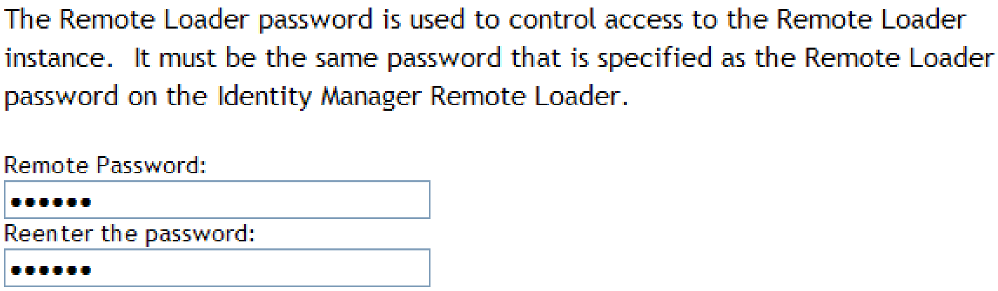

The Driver Object Password is used by the Remote Loader to authenticate itself to the Identity Manager server. It must be the same password that is specified as the Driver Object Password on the Identity Manager Remote Loader.

Driver Password: ••••••

Reenter the password: ••••••

Make sure that this password is difficult to guess and is different from the Remote Loader password.

5 Type and re-enter the Remote Loader password, then click *Next*.


The Remote Loader password is used to control access to the Remote Loader instance. It must be the same password that is specified as the Remote Loader password on the Identity Manager Remote Loader.

Remote Password: ••••••

Reenter the password: ••••••

Make sure that this password is difficult to guess and is different from the driver object password.

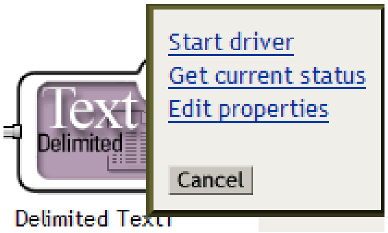
6 Define a security-equivalent user, click *Next*, then click *Finish*.

3.8.2 Configuring an Existing Driver

Specify parameters on the Driver object for connecting to the Remote Loader.

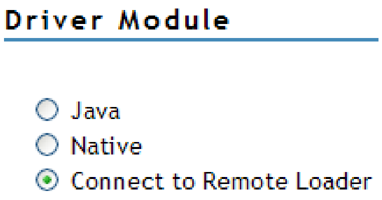
1 In iManager, click *Identity Manager > Identity Manager Overview*.

2 Browse to and select the driver that you want to modify.



3 Click the driver status icon, then click *Edit Properties*.

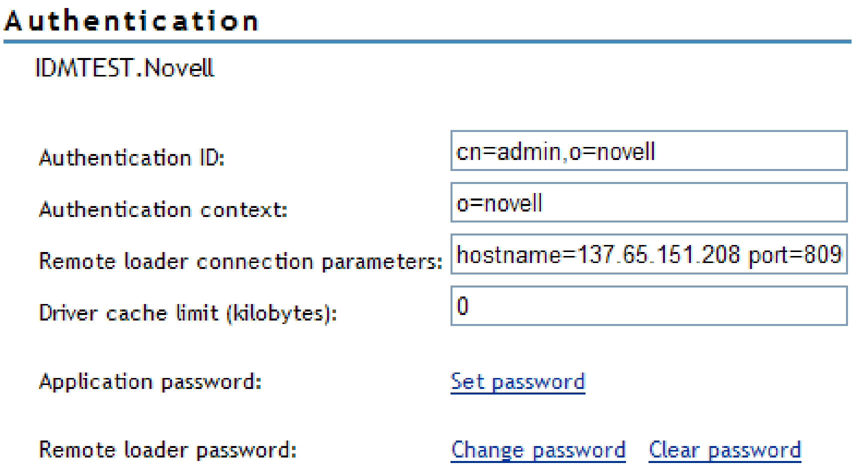
4 In the *Driver Module* section, select *Connect to Remote Loader*.



Driver Module

○ Java
○ Native
● Connect to Remote Loader

5 In the *Authentication* section, specify parameters for the Remote Loader.



Authentication

IDMTEST.Novell

Authentication ID:	cn=admin,o=novell
Authentication context:	o=novell
Remote loader connection parameters:	hostname=137.65.151.208 port=809
Driver cache limit (kilobytes):	0
Application password:	Set password
Remote loader password:	Change password Clear password

- Remote Loader Connection Parameters

  Earlier, you exported the self-signed certificate. (See Section 3.2.2, "Exporting a Self-Signed Certificate," on page 56.) For SSL, you need the nickname of the self-signed certificate.

In the *Remote Loader Connection Parameters* edit box, type parameters in key-value pairs. For example, type

```
hostname=192.168.0.1 port=8090 kmo=remotecert
hostname=192.168.0.1 port=8090 kmo='remote cert'
```

- **hostname:** The host name or IP address (for example, 190.162.0.1). Specifies the address or name of the computer that the Remote Loader runs on. If you don't specify the IP address or server name, this value defaults to localhost.
- **port:** Where the Remote Loader accepts connections from the remote interface shim. If you don't specify this communication parameter, this value defaults to 8090.
- **kmo:** Specifies the Key Name (for example, kmo=remotecert) of the Key Material Object (KMO) containing the keys and certificate used for SSL.

  If you used spaces in the certificate name, you need to enclose the KMO object nickname in single quotation marks.

  The KMO object name is the nickname value you specified in Step 2 of Section 3.2.1, "Creating a Server Certificate," on page 55.

- **Enter the Application Password:** Specify the password of the application user ID. Typically, the driver shim needs this password so that the driver can connect to the application. Make sure that this password is difficult to guess and is different from the Remote Loader password.
- **Enter the Remote Loader Password:** Specify the password for the Remote Loader. The remote interface shim uses this password to authenticate itself to the Remote Loader. Make sure that this password is difficult to guess and is different from the driver object password.

  Set or reset both the application password and the Remote Loader password at the same time.

6 Click *OK*.

3.8.3 Creating a Keystore

A keystore is a Java file that contains encryption keys and, optionally, certificates. If you want to use SSL between the Remote Loader and the Metadirectory engine, and you are using a Java shim, you need to create a keystore file.

- "Keystore on Windows" on page 87
- "Keystore on Solaris, Linux, or AIX" on page 87
- "Keystore on All Platforms" on page 88

Keystore on Windows

On Windows, run the Keytool utility, typically found in the `c:\novell\remoteloader\jre\bin` directory.

Keystore on Solaris, Linux, or AIX

On Solaris, Linux, or AIX environments, use the `create_keystore` file. `Create_keystore` is installed with rdxml and is also included in the `dirxml_jremote.tar.gz` file, found in the `\dirxml\java_remoteloader` directory. The `create_keystore` file is a shell script that calls the Keytool utility.

On UNIX, when the self-signed certificate is used to create the keystore, the certificate can be exported in Base64 or binary DER format.

Enter the following at the command line:

```
create_keystore self-signed_certificate_name keystorename
```

For example, type one of the following

```
create_keystore tree-root.b64 mystore
create_keystore tree-root.der mystore
```

The `create_keystore` script specifies a hard-coded password of “dirxml” for the keystore password. This is not a security risk because only a public certificate and public key are stored in the keystore.

Keystore on All Platforms

To create a keystore on any platform, you can enter the following at the command line:

```
keytool -import -alias trustedroot -file self-signed_certificate_name -
keystore filename -storepass
```

The filename can be any name (for example, rdev_keystore).

Creating Policies

4

Policies enable you to customize the flow of information into and out of the Identity Vault, for a particular environment.

For example, one company might use the inetorgperson as the main user class, and another company might use User. To handle this, a policy is created that tells the Metadirectory engine what a user is called in each system. Whenever operations affecting users are passed between connected systems, Identity Manager applies the policy that makes this change.

Policies also create new objects, update attribute values, make schema transformations, define matching criteria, maintain Identity Manager associations, and many other things.

A detailed guide to Policies is contained in the *Understanding Policies for Identity Manager 3.5.1* (http://www.novell.com/documentation/idm35/policy/data/bookinfo.html). This guide contains:

- A detailed description of each available policy
- A discussion on creating policies using XSLT style sheets.

An in-depth Policy Builder user guide and reference, including examples and syntax for each condition, action, noun, and verb at *Policies in Designer 2.1* (http://www.novell.com/documentation/idm35/policy_designer/data/bookinfo.html) and at *Policies in iManager for Identity Manager 3.5.1* (http://www.novell.com/documentation/idm35/policy_imanager/data/bookinfo.html).

Password Synchronization across Connected Systems

5

5.1 Overview

Identity Manager provides bidirectional password synchronization, by taking advantage of Universal Password and connected system support for publishing or subscribing to passwords.

As with other attributes for a user account, you can choose your authoritative data sources.

5.1.1 Overview of Passwords

NDS® passwords, Simple passwords, Distribution passwords, and Universal passwords are used for different purposes. In previous versions of eDirectory™ and Identity Manager, connected systems could update only the NDS password, in a one-way synchronization.

Identity Manager uses Universal Password, which is a reversible password that can be synchronized with the other Identity Vault passwords. Universal Password was introduced in eDirectory 8.7.1, and is protected by three layers of encryption.

NMAS™ controls the relationship between Universal Password and the other Identity Vault passwords. For example, NMAS controls whether Universal Password is kept synchronized with NDS Password, Simple Password, or Distribution Password. NMAS intercepts incoming requests to change passwords and handles them according to settings in NMAS password policies.

Identity Manager uses the Distribution Password to control password synchronization between the Identity Vault and connected systems. Identity Manager implements certain password synchronization features using the Distribution Password, including bidirectional password synchronization policies between Identity Vault and connected systems; password tunneling; and password check status on connected systems.

Like Universal Password, Distribution Password is protected by three layers of encryption, and is reversible.

In the NMAS password policy, you can specify whether the Distribution Password should be the same as the Universal Password. (The setting is *Synchronize Distribution Password when setting Universal Password*). If the Distribution Password is the same as the Universal Password, and you choose to use bidirectional Password Synchronization with connected systems, keep in mind that you are using Identity Manager to extract the Universal Password from eDirectory and send it to other connected systems. You need to secure the transport of the password, as well as the connected systems it will be stored on. (See Chapter 9, “Security: Best Practices,” on page 259.)

If the Distribution Password is not the same as the Universal Password (because you disable the setting in the NMAS password policy), you can “tunnel” passwords among connected systems that use the Distribution Password, without using or affecting the Universal Password or NDS Password. Keep in mind that tunnelling synchronizes passwords among connected systems only. If enabled, tunneling does not set the Identity Vault/Universal password.

For more information on the various eDirectory passwords, see the *Novell Modular Authentication Services (NMAS) 2.3 Administration Guide* (http://www.novell.com/documentation/nmas23/index.html). For examples of different ways of using password synchronization with Identity Manager, see Section 5.8, “Implementing Password Synchronization,” on page 124.

5.1.2 What Is Bidirectional Password Synchronization?

Bidirectional password synchronization is the combination of Identity Manager accepting passwords from the connected systems you specify, and distributing passwords to the connected systems you specify.

The ability to have bidirectional password synchronization with a particular connected system depends on what the connected system supports.

Some connected systems can accept new and modified passwords from Identity Manager, and can also provide the user's actual password to Identity Manager. These connected systems are the ones that support bidirectional password synchronization with Identity Manager:

- Active Directory
- Novell® eDirectory
- Network Information Services (NIS)
- NT Domain

For these connected systems, the user can change a password in one of the systems and have that password synchronized to the other systems through Identity Manager. However, if you are using Advanced Password Rules in your NMAS password policies, it's best to have users make password changes in the User Application self-service console. This is the best place for password changes because it lists all the rules that the user's password must comply with.

Because other connected systems can't provide the user's actual password, they can't support full bidirectional password synchronization. However, they can provide data that can be used to create passwords and send them to Identity Manager, by defining policies within the driver configuration.

Several other systems can accept passwords from Identity Manager, including setting an initial password for a new user, modifying a password, or both. See Section 5.2, "Connected System Support for Password Synchronization," on page 100.

5.1.3 Comparison of Password Synchronization 1.0 and Identity Manager Password Synchronization

Table 5-1 *Password Synchronization 1.0 and Identity Manager Password Synchronization*

	Password Synchronization 1.0	**Password Synchronization with Identity Manager 2 and 3**
Product delivery	A product separate from Identity Manager.	Included with Identity Manager, not sold separately.
Platforms	◆ Active Directory ◆ NT Domain ◆ eDirectory	Full bidirectional password synchronization is supported on these platforms: ◆ Active Directory ◆ eDirectory ◆ NIS ◆ NT Domain These connected systems support publishing user passwords to Identity Manager. Because Universal Password and Distribution Password are reversible, Identity Manager can distribute passwords to connected systems. Any connected system that supports the Subscriber password element can subscribe to passwords from Identity Manager. See Section 5.2, "Connected System Support for Password Synchronization," on page 100.
Password used in an Identity Vault	NDS Password (non-reversible)	Universal Password (reversible), or Distribution Password (also reversible). The NDS password can also be kept synchronized, if desired. For example scenarios, see Section 5.8, "Implementing Password Synchronization," on page 124.

	Password Synchronization 1.0	Password Synchronization with Identity Manager 2 and 3
Main functionality for Windows connected systems	To send passwords to Identity Manager so the Identity Vault password is synchronized with the Windows password. Because the NDS password is not reversible, passwords were not sent back to NT or AD.	To provide bidirectional password synchronization. Because Universal Password and Distribution Password are reversible, passwords can be synchronized in both directions.
LDAP changes	Not supported.	Supported
Novell Client™	Required.	Not required.
nadLoginName attribute	Used for keeping passwords updated.	Not used.
The component that contains the password synchronization functionality	The Identity Manager driver contained the functionality for updating nadLoginName.	Identity Manager policies in the driver configuration provide the password synchronization functionality. The driver simply carries out the tasks given by the Metadirectory engine, which come from logic in the policies. The driver manifest, global configuration values, and driver filter settings must also support password synchronization. These are included in the sample driver configurations, or can be added to an existing driver. See Section 5.7, “Upgrading Existing Driver Configurations to Support Password Synchronization,” on page 116.
Agents	A separate piece of software.	No agents are installed; instead, the functionality is now part of the driver.

5.1.4 Features of Identity Manager Password Synchronization

Identity Manager Password Synchronization is bidirectional. Passwords can be sent from connected systems and accepted by Identity Manager, and passwords can be distributed by Identity Manager and accepted by connected systems.

- “Accepting Passwords from Connected Systems” on page 94
- “Distributing Passwords to Connected Systems” on page 95
- “Enforcing Password Policies in the Data Store and on Connected Systems” on page 95
- “Scenarios for Synchronizing Passwords” on page 96
- “Notifying Users of Password Synchronization Failures” on page 97
- “Checking the Password Synchronization Status for a User” on page 97

Accepting Passwords from Connected Systems

As in previous versions of DirXML® and Identity Manager, any connected system can publish a password to the Identity Vault.

You can specify which connected system applications Identity Manager accepts passwords from. You can even choose whether Identity Manager updates the password for users in the same Identity Vault where Identity Manager is running, or whether Identity Manager simply acts as a conduit or "tunnel," synchronizing passwords only between connected systems. This means that it is possible to keep the Identity Vault password separate from the password that Identity Manager distributes to connected systems, if desired.

Some connected systems (AD, other Identity Vaults, NT, and NIS) can provide the user's actual password, which means that when a user changes a password on a connected system, the change can be synchronized to Identity Manager and back out to other connected systems.

Other connected systems don't support providing the user's actual password, but you can configure them to provide to Identity Manager a password that is manufactured in a style sheet, such as an initial password based on last name or employee ID.

Distributing Passwords to Connected Systems

Identity Manager Password Synchronization can distribute a common password to connected systems.

In previous versions of Identity Manager, a driver could send passwords to Identity Manager from a user account on a connected system, and the password could be used to update the corresponding user in eDirectory. But because the NDS password in eDirectory is non-reversible, you couldn't push a password out from the central Identity Manager Identity Vault to multiple connected systems. You could obtain the eDirectory password only by capturing the password before it was stored in eDirectory, such as through the Novell Client.

The Universal Password provided by eDirectory 8.7.3 is reversible. Because of this, Identity Manager can accept a password from a connected system, then distribute the password from the Identity Vault to connected systems that support setting initial password for new accounts and modifying a password.

Regardless of where the password comes from, Identity Manager uses the Distribution Password as the repository from which it distributes passwords to connected systems. The Distribution Password, like the Universal Password, lets you enforce password policies.

For information about using Universal Password and Distribution Password when synchronizing passwords, see "Implementing Password Synchronization" on page 124.

As with other attributes of a user, you can decide which systems are authoritative sources for passwords. Identity Manager distributes the passwords from the authoritative source to the other connected systems.

You can set up bidirectional password synchronization among connected systems that support it.

Enforcing Password Policies in the Data Store and on Connected Systems

By making calls to NMAS, Identity Manager can enforce password policies on incoming passwords. If the password being published from a connected system to Identity Manager does not comply, you can specify that Identity Manager not accept the password into the Identity Vault. This also means that passwords that don't comply with your policies are not distributed to other connected systems.

In addition, Identity Manager can enforce password policies on connected systems. If the password being published to Identity Manager does not comply with rules in a policy, you can specify that Identity Manager not only does not accept the password for distribution, but actually resets the noncompliant password on the connected system by using the current Distribution Password in the Identity Vault.

For example, you want to require passwords to include at least one numeric character. However, the connected system does not have the ability to enforce such a policy. You specify that Identity Manager resets passwords that flow from the connected system but do not comply with rules in the policy.

If you are using Advanced Password Rules and Identity Manager Password Synchronization, we recommend that you research the password policies for all the connected systems to make sure that the Advanced Password Rules in the eDirectory password policy are compatible. This research helps ensure that passwords are synchronized successfully.

Keep in mind that you must make sure that the users who are assigned NMAS password policies match with the users you want to participate in Password Synchronization for connected systems.

NMAS password policies are assigned with a tree-centric perspective. In contrast, Password Synchronization is set up per driver. Also, drivers are installed on a per-server basis and can manage only those users who are in a master or read/write replica. To get the results you expect from Password Synchronization, make sure the containers that are in a master or read/write replica on the server running the drivers for Password Synchronization match the containers where you have assigned password policies with Universal Password enabled. Assigning a password policy to a partition root container ensures that all users in that container and subcontainers are assigned the password policy.

For information about how NMAS password policies are assigned to users, see “Assigning Password Policies to Users” in the *Password Management Administration Guide* (http://www.novell.com/documentation/password_management/index.html).

Scenarios for Synchronizing Passwords

Identity Manager enables you to specify which systems should be authoritative sources for passwords. Also, you decide how you want passwords to flow.

Much of the functionality of Identity Manager Password Synchronization relies on Universal Password, the reversible password functionality provided by the Identity Vault. However, some scenarios don't require you to deploy Universal Password.

Identity Manager Password Synchronization also relies on the Distribution Password. As with Universal Password, a policy can be enforced on the Distribution Password.

For basic ways that you can implement password synchronization, see “Implementing Password Synchronization” on page 124. You can combine these scenarios to meet the needs of your environment.

Synchronizing Passwords on Windows without the Novell Client

A Novell Client is no longer required for password synchronization with Active Directory and NT Domain.

Notifying Users of Password Synchronization Failures

The “Enforcing Password Policies in the Data Store and on Connected Systems” on page 95 explains that Identity Manager can enforce password policies by not accepting (from connected systems) passwords that don't comply.

Using the e-mail notification feature, you can specify that Identity Manager notify the user when a password change that the user made was not successful.

Scenario. You have set Identity Manager to not accept an incoming password from NT Domain if it doesn't comply with your password policy. You have enabled e-mail notification. One rule in your NMAS password policy specifies that the company name can't be used as a password. A user changes the password on the NT Domain connected system to be the company name. NMAS does not accept the password, and Identity Manager sends an e-mail message to the user stating that the password change was not synchronized.

Before you can use this feature, you must set up the e-mail server and templates. You can customize the following:

- The text of the messages that Identity Manager sends
- The notification that sends a copy to the administrator

For more information, see “Configuring E-Mail Notification” on page 161.

Checking the Password Synchronization Status for a User

Identity Manager enables you to query connected systems to check a user’s password synchronization status. If the connected system supports the check password feature, you can find out whether passwords are synchronizing successfully.

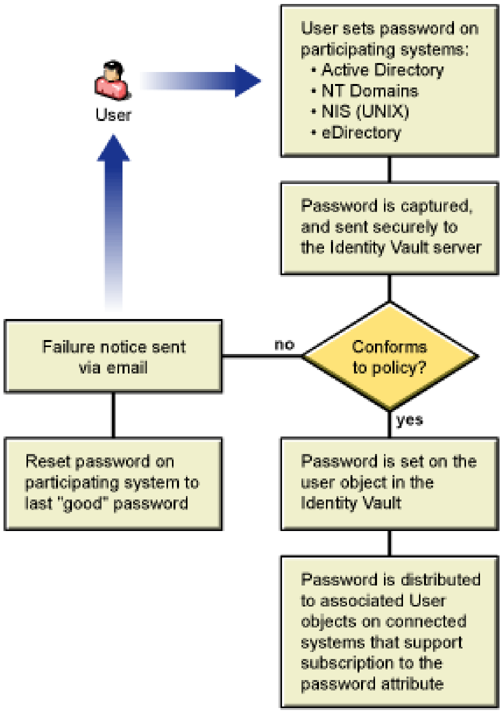
For information on how to check passwords, see “Checking the Password Synchronization Status for a User” on page 160.

For a list of which systems support checking passwords, see “Connected System Support for Password Synchronization” on page 100.

5.1.5 Overview Illustrations of Password Synchronization Flow

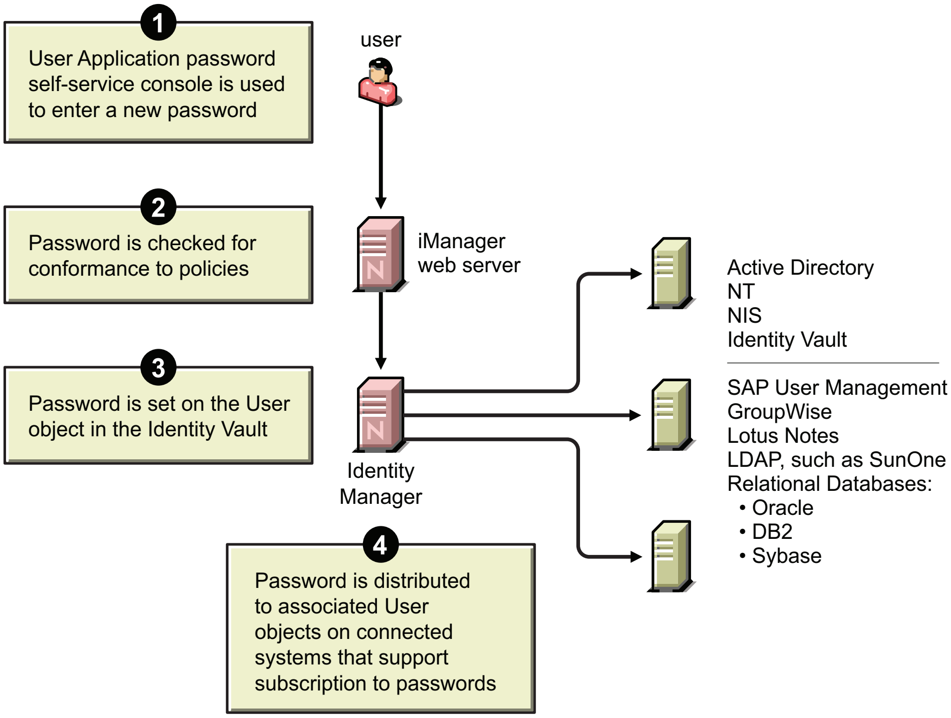
The following figure describes how connected systems publish passwords to Identity Manager.

Figure 5-1 How Connected Systems Publish Passwords to Identity Manager.


User
User sets password on participating systems:
• Active Directory
• NT Domains
• NIS (UNIX)
• eDirectory
Password is captured, and sent securely to the Identity Vault server
Conforms to policy?
no
yes
Failure notice sent via email
Reset password on participating system to last "good" password
Password is set on the user object in the Identity Vault
Password is distributed to associated User objects on connected systems that support subscription to the password attribute

The following figure describes how Identity Manager distributes passwords to connected systems.

Figure 5-2 How Identity Manager Distributes Passwords to Connected Systems

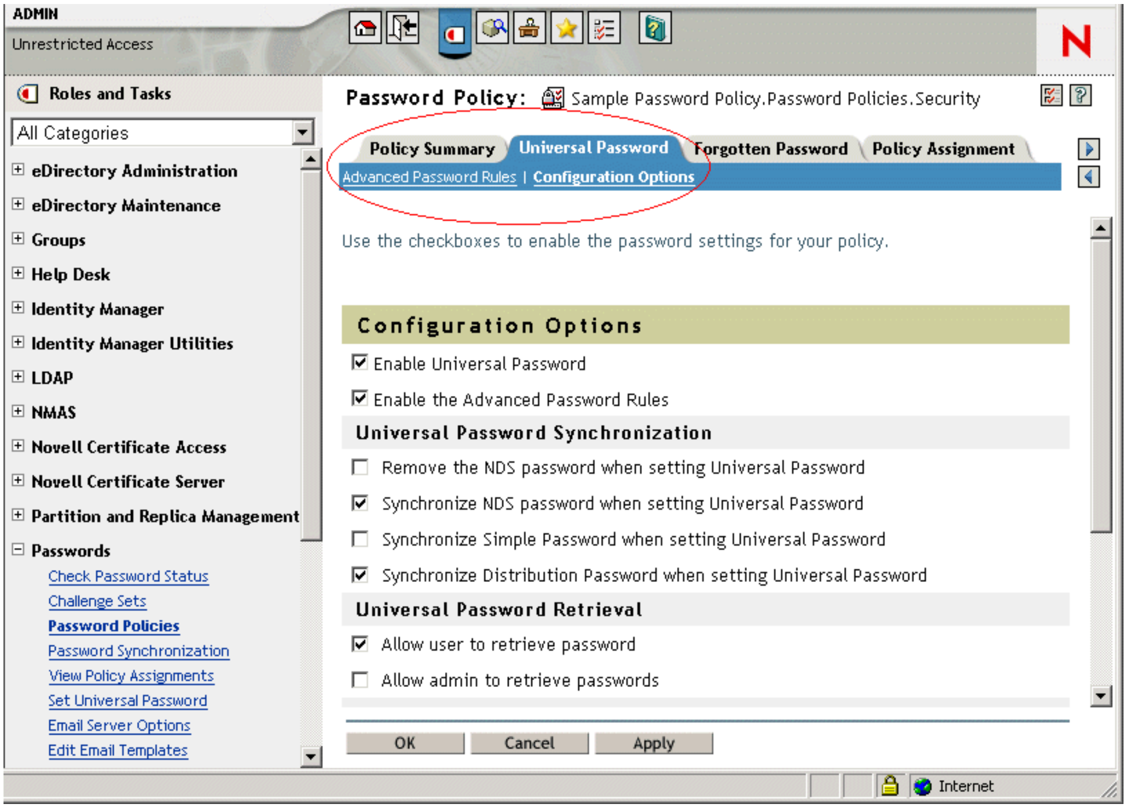

1
User Application password self-service console is used to enter a new password
2
Password is checked for conformance to policies
3
Password is set on the User object in the Identity Vault
4
Password is distributed to associated User objects on connected systems that support subscription to passwords
user
iManager web server
Identity Manager
Active Directory
NT
NIS
Identity Vault
SAP User Management
GroupWise
Lotus Notes
LDAP, such as SunOne
Relational Databases:
• Oracle
• DB2
• Sybase

5.1.6 Browser Display Variations

This documentation frequently uses figures in procedures to illustrate options in iManager. How the options actually display on your desktop depends on your browser.

For example, Internet Explorer displays iManager options by using tabs.

Figure 5-3 *Tabs in iManager*



Password Policy: Sample Password Policy.Password Policies.Security

Use the checkboxes to enable the password settings for your policy.

Configuration Options

☑ Enable Universal Password

☑ Enable the Advanced Password Rules

Universal Password Synchronization

☐ Remove the NDS password when setting Universal Password

☑ Synchronize NDS password when setting Universal Password

☐ Synchronize Simple Password when setting Universal Password

☑ Synchronize Distribution Password when setting Universal Password

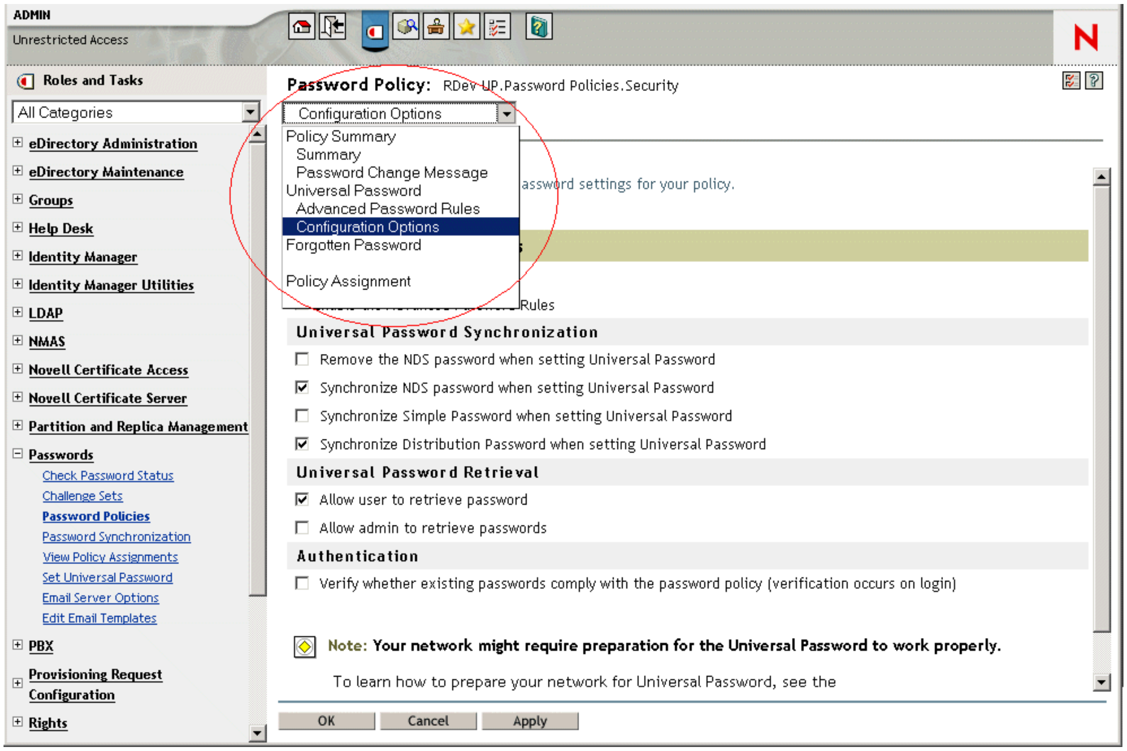
Universal Password Retrieval

☑ Allow user to retrieve password

☐ Allow admin to retrieve passwords

However, the Firefox browser displays iManager options by using a drop-down list.

Figure 5-4 *A Drop-Down List in iManager*



Password Policy: RDev-UP.Password Policies.Security

asswqrd settings for your policy.

Rules

Universal Password Synchronization

□ Remove the NDS password when setting Universal Password

☑ Synchronize NDS password when setting Universal Password

□ Synchronize Simple Password when setting Universal Password

☑ Synchronize Distribution Password when setting Universal Password

Universal Password Retrieval

☑ Allow user to retrieve password

□ Allow admin to retrieve passwords

Authentication

□ Verify whether existing passwords comply with the password policy (verification occurs on login)

Note: Your network might require preparation for the Universal Password to work properly.

To learn how to prepare your network for Universal Password, see the

In this documentation, figures are displayed as they appear in the Firefox browser.

5.2 Connected System Support for Password Synchronization

When a User object is created, Identity Manager is always capable of accepting a password from a connected system, even if the connected system does not support providing the user's actual password from that system.

AD, NT, eDirectory, and NIS can accept a password from Identity Manager and also support sending the user's actual password to Identity Manager. This means they offer full support for bidirectional password synchronization.

When you define a policy within the driver configuration on the Publisher channel, other systems can provide data that can be used to create passwords. The example driver configurations for most of the drivers include an example policy that provides a default password based on Surname.

Connected systems have varying abilities to accept a password from Identity Manager. Some connected systems support setting an initial password for new accounts, but not Password Modify events.

The capabilities of the sample driver configurations are noted in the driver manifest. The following tables provide additional information that is not in the driver manifest. The tables indicate whether an application accepts initial password for a new account, versus whether it can accept a modification to an existing password. The manifest indicates only that the connected system is capable of accepting a password, and doesn't show this distinction.

Drivers are in groups so that you can see sample driver configurations that have similar abilities.

5.2.1 Systems That Support Bidirectional Password Synchronization

The following connected systems support bidirectional password synchronization. They can provide the user's actual password on the connected system, and accept passwords from Identity Manager.

Table 5-2 *Systems that Support Bidirectional Password Synchronization*

Connected System Driver	Subscriber Channel	Subscriber Channel	Subscriber Channel	Publisher Channel
	Application Can Accept Setting of Initial Password	**Application Can Accept Modification of Password**	**Application Supports Check Password**	**Application Can Provide (sync) Password**
Active Directory	Yes	Yes	Yes	Yes
eDirectory[1]	Yes	Yes	Yes	Yes
NT Domain	Yes	Yes	No	Yes
NIS	Yes	Yes	Yes	Yes
SIF	Yes	Yes	No	Yes

[1] Between Identity Vault trees, you can have bidirectional password synchronization for users even if Universal Password is not enabled for those users. See Section 5.8.2, “Scenario 1: Using NDS Password to Synchronize between Two Identity Vaults,” on page 126.

5.2.2 Systems That Accept Passwords from Identity Manager

The following connected systems can accept passwords from Identity Manager to some degree. They can't provide a user's actual password on the connected system to Identity Manager.

Although they can't provide the user's actual password, they can be configured to create a password by using a policy on the Publisher channel, based on other user data in the connected system. (The sample driver configurations demonstrate a default password based on the surname.)

Table 5-3 *Systems That Accept Passwords from Identity Manager*

Connected System Driver	Subscriber Channel	Subscriber Channel	Subscriber Channel	Publisher Channel
	Application Can Accept Setting of Initial Password	**Application Can Accept Modification of Password**	**Application Supports Check Password**	**Application Can Provide (Sync) Password**
Groupwise®	Yes	Yes	No	No[2]
JDBC	Yes[3]	No4	No	No[5]
LDAP	Yes[6]	Yes[6]	Yes	No

Connected System Driver	Subscriber Channel	Subscriber Channel	Subscriber Channel	Publisher Channel
	Application Can Accept Setting of Initial Password	**Application Can Accept Modification of Password**	**Application Supports Check Password**	**Application Can Provide (Sync) Password**
Notes	Yes	Yes[7]	Yes[7]	No
SAP User Management	Yes	Yes	No	No

[2] GroupWise supports two authentication methods:

- GroupWise provides its own authentication and maintains user passwords.
- GroupWise authenticates against eDirectory using LDAP and does not maintain passwords.

  When you use this option, GroupWise ignores driver-synchronized passwords.

[3] The ability to set an initial password is available on all databases where the OS user account is distinct from the database user account, such as Oracle*, MS SQL, MySQL*, and Sybase*.

[4] The Identity Manager Driver for JDBC can be used to modify a password on the connected system, but that feature is not demonstrated in the sample driver configuration.

[5] Passwords can be synchronized as data when stored in a table.

[6] If the target LDAP server allows setting the userpassword attribute.

[7] The Notes driver can accept a password modification and check passwords only for the HTTPPassword field in Lotus Notes.

5.2.3 Systems That Don’t Accept or Provide Passwords

The following connected systems can't accept passwords or provide a user's password on the connected system using the sample driver configuration.

Although they can't provide the user's password to Identity Manager, they can be configured to create a password by using a policy on the Publisher channel, based on other user data in the connected system. (The sample driver configurations demonstrate a default password based on surname.)

Table 5-4 *Systems That Don’t Accept or Provide Passwords*

Connected System Driver	Subscriber Channel	Subscriber Channel	Subscriber Channel	Publisher Channel
	Application Can Accept Setting of Initial Password	**Application Can Accept Modification of Password**	**Application Supports Check Password**	**Application Can Provide (Sync) Password**
Delimited Text	No[8]	No[8]	No[8]	No[8]
Exchange 5.5	No	No	No	No
PeopleSoft 3.6	No	No	No	No

Connected System Driver	Subscriber Channel	Subscriber Channel	Subscriber Channel	Publisher Channel
	Application Can Accept Setting of Initial Password	**Application Can Accept Modification of Password**	**Application Supports Check Password**	**Application Can Provide (Sync) Password**
PeopleSoft 4.0	No	No	No	No
SAP HR	No	No	No	No

[8] The Identity Manager Driver for Delimited Text does not have features in the driver shim that directly support Password Synchronization. However, the driver can be configured to handle passwords, depending on the connected system you are synchronizing with.

5.2.4 Systems That Don't Support Password Synchronization

The following connected systems are not intended to be used with password synchronization.

Table 5-5 *Systems That Don't Support Password Synchronization*

Connected System Driver	Subscriber Channel	Subscriber Channel	Subscriber Channel	Publisher Channel
	Application Can Accept Setting of Initial Password	**Application Can Accept Modification of Password**	**Application Supports Check Password**	**Application Can Provide (sync) Password**
Avaya* PBX	No	No	No	No
Entitlements Service Driver	No	No	No	No
GenericLoopBack Service Driver	No	No	No	No
Manual Task Service Driver	No	No	No	No

5.3 Prerequisites for Password Synchronization

Password Synchronization depends on the following elements being in place:

- "Support for Universal Password" on page 104
- "Password Synchronization Capabilities in the Driver Manifest" on page 104
- "Using Global Configuration Values to Control Password Synchronization" on page 104
- "Policies Required in the Driver Configuration" on page 107
- "Filters You Install on the Connected System to Capture Passwords" on page 111
- "NMAS Password Policies You Create for Users" on page 111
- "NMAS Login Methods" on page 111

5.3.1 Support for Universal Password

To accommodate password synchronization across connected systems, Identity Manager requires Universal Password. See the following:

- "Deploying Universal Password" in the *Password Management Administration Guide* (http://www.novell.com/documentation/password_management/index.html)
- Section 5.4.3, "Preparing to Use Universal Password," on page 113

5.3.2 Password Synchronization Capabilities in the Driver Manifest

The driver manifest declares whether a connected system supports the following password synchronization functions:

- Publishing the user's actual password to Identity Manager
- Accepting a password from Identity Manager

  The manifest does not distinguish between accepting the creation of an initial password versus accepting password modifications.
- Letting Identity Manager check the password on the connected system, to determine the password synchronization status of a user

NOTE: The driver manifest is written by the driver developer or the Identity Manager expert who creates the driver configuration. It is not meant to be edited by a network administrator. The driver manifest represents the true capabilities of the driver shim and configuration. Changing the manifest alone does not change functionality. To add functionality, the driver shim, connected system, or driver configuration might be enhanced.

The sample driver configurations delivered with Identity Manager contain driver manifest entries. To add them to an existing driver, see Section 5.7, "Upgrading Existing Driver Configurations to Support Password Synchronization," on page 116.

5.3.3 Using Global Configuration Values to Control Password Synchronization

Global configuration values enable you to set a constant value that you can reference in a policy. Global configuration values are sometimes called server variables, because they are held in an attribute that is per replica.

For Password Synchronization, global configuration values enable you to create settings for the flow of passwords to and from Identity Manager. Because the Identity Manager password synchronization policies in the driver configuration are written to behave differently based on your settings in the global configuration value, it's easy to change the flow of passwords without having to edit policies.

By using global configuration values, you control the following settings separately for each connected system:

Table 5-6 *Settings for Connected Systems*

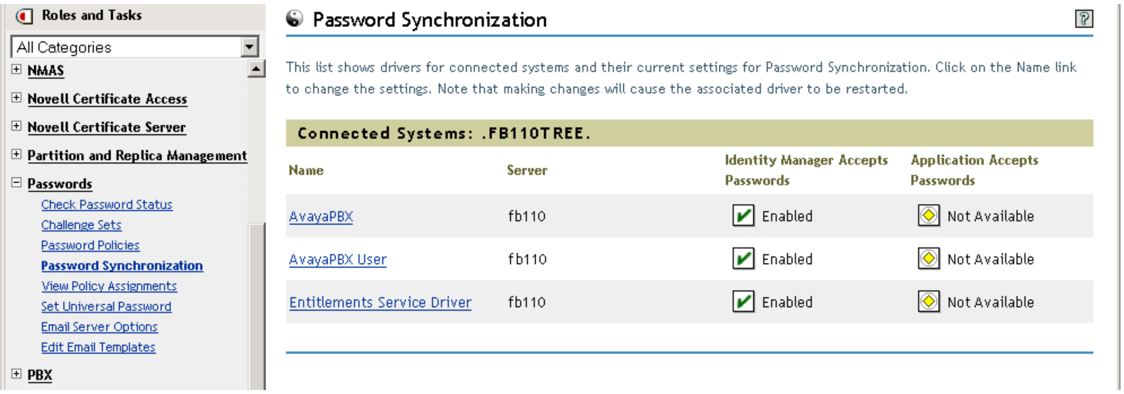
Setting	Description
Whether Identity Manager accepts passwords from the connected system	This setting applies to a password provided by the connected system, as well as a password that could be created by Identity Manager policies in the driver configuration on the Publisher channel. If you disable this setting, both kinds of passwords are stripped out so that they don't reach Identity Manager.
Whether Identity Manager updates Universal Password directly, or updates Distribution Password directly	Identity Manager controls the entry point (that password Identity Manager updates). NMAS controls the flow of passwords between each different kind of password, based on what you have set in the NMAS password policy. To view an NMAS password policy: 1. In iManager, select *Passwords* > *Password Policies*. 2. Select a policy in the *Password Policy List*. 3. Click *Edit*. 4. Select an option from the drop-down list or tab (depending on which version of iManager you are using). See Section 5.8, “Implementing Password Synchronization” for scenarios that use these methods.
Whether NMAS password policies are enforced on passwords coming in to Identity Manager from a connected system	If these policies are enforced, noncompliant passwords coming in are not written to the Identity Manager data store.
Whether Identity Manager uses the Identity Manager password to enforce NMAS password policies on a connected system, by resetting passwords that don't comply with the policy rules	This option is dimmed in the NMAS interface if the connected system doesn't support it (as declared in the driver manifest). The password is reset only after a password operation fails on the Publisher channel.
Whether the connected system accepts passwords	This setting applies to both a password distributed by Identity Manager and a password that could be created by Identity Manager policies in the driver configuration on the Subscriber channel. If you disable this setting, both kinds of passwords are stripped out so that they don't reach the connected system. This option is dimmed in the interface if the connected system doesn't support it (as declared in the driver manifest).
Whether users are notified by e-mail when a password is not synchronized	Automatically sends e-mails to affected users.

The driver configurations delivered with Identity Manager contain driver manifest entries. To add them to an existing driver, see Section 5.7, “Upgrading Existing Driver Configurations to Support Password Synchronization,” on page 116.

To edit global configuration values:

1. In iManager, select *Passwords* > *Password Synchronization*.
2. Search for a driver.

After you specify where you want to search for connected system drivers, iManager displays an overview of the password flow settings for all the connected system drivers it finds.



Password Synchronization

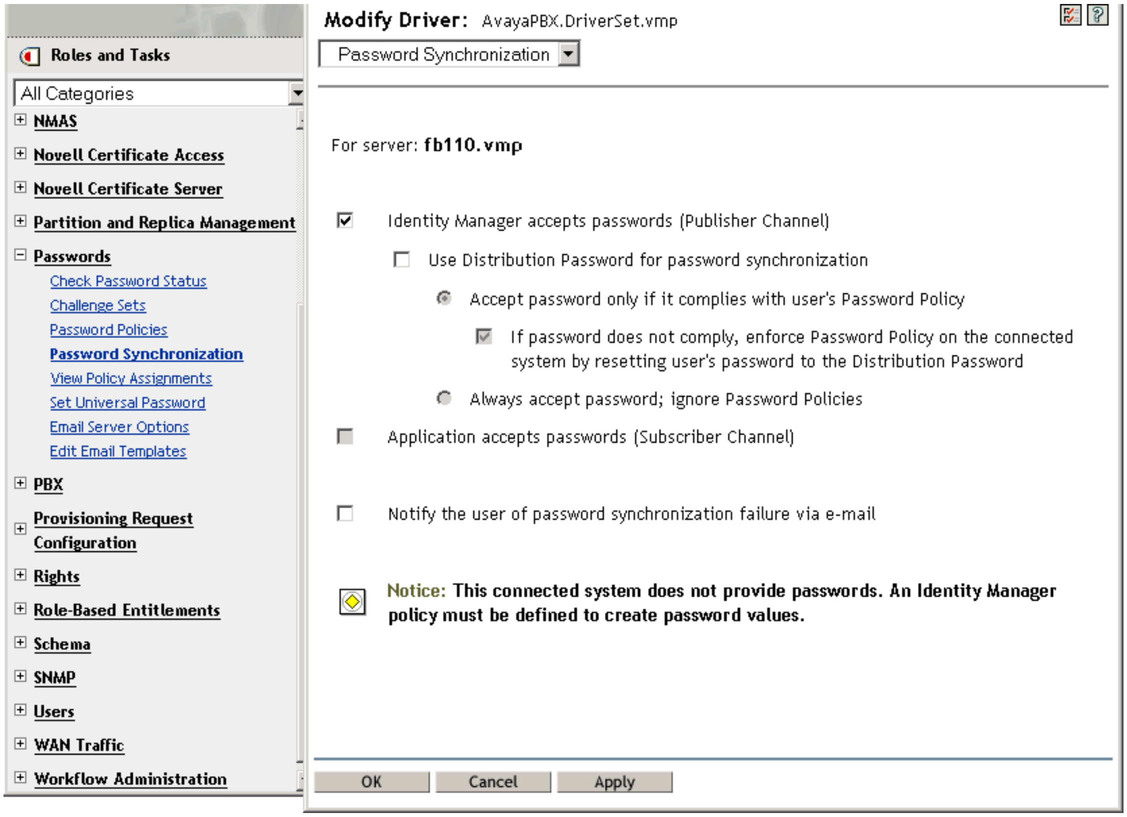
This list shows drivers for connected systems and their current settings for Password Synchronization. Click on the Name link to change the settings. Note that making changes will cause the associated driver to be restarted.

Connected Systems: .FB110TREE.

Name	Server	Identity Manager Accepts Passwords	Application Accepts Passwords
AvayaPBX	fb110	☑ Enabled	◇ Not Available
AvayaPBX User	fb110	☑ Enabled	◇ Not Available
Entitlements Service Driver	fb110	☑ Enabled	◇ Not Available

3 To view settings, click a driver name.

The Modify Driver page displays the global configuration values for Password Synchronization.



Modify Driver: AvayaPBX.DriverSet.vmp

Password Synchronization

For server: **fb110.vmp**

☑ Identity Manager accepts passwords (Publisher Channel)

☐ Use Distribution Password for password synchronization

● Accept password only if it complies with user's Password Policy

☑ If password does not comply, enforce Password Policy on the connected system by resetting user's password to the Distribution Password

○ Always accept password; ignore Password Policies

☐ Application accepts passwords (Subscriber Channel)

☐ Notify the user of password synchronization failure via e-mail

◇ Notice: **This connected system does not provide passwords. An Identity Manager policy must be defined to create password values.**

If an option on this page is dimmed, the driver manifest shows that the connected system does not support that option.

4 Make changes, then click *OK*.

NOTE: You can set global configuration values on each driver separately. Global configuration values on a driver override those on the driver set. Setting the values on a specific driver gives you more granular control. This page displays only the global configuration values that are present on the individual driver.

If you set global configuration values on the Driver Set object, those values are inherited by a driver in that driver set if the driver does not have values of its own. If a driver has no settings of its own and inherits the global configuration values from the driver set, iManager does not display them. Although iManager does not display inherited global configuration values, they are still honored by the password synchronization policies.

5.3.4 Policies Required in the Driver Configuration

Identity Manager policies on the Publisher and Subscriber channels for each driver govern the password flow, based on your settings in the global configuration variables explained above. These policies are included in the driver configurations in Identity Manager.

If you are upgrading an existing driver configuration instead of replacing it, you must add certain policies to the configuration. (See Section 5.7, "Upgrading Existing Driver Configurations to Support Password Synchronization," on page 116.) These policies must be in your driver configuration in the correct location for password synchronization to work.

- "Policies Required in the Publisher Command Transformation Set" on page 107
- "Policies Required in the Publisher Input Transformation Policy Set" on page 109
- "Policies Required in the Subscriber Command Transformation Policy Set" on page 109
- "Policies Required in the Subscriber Output Transformation Policy Set" on page 110

Policies Required in the Publisher Command Transformation Set

The policies listed in the Password Synchronization Policy Name column must be present in the order listed. Also, they must be the last policies in the Publisher Command Transformation policy set.

Table 5-7 Policies Required in the Publisher Command Transformation Set

Location in the Driver Configuration	Password Synchronization Policy Name	What the Policy Does
Publisher Command Transformation	Password(Pub)-Default Password Policy	Adds a default password to an Add object if the Add object does not already contain a password. This policy and the Password(Sub)-Default Password Policy are the only policies that you can modify or remove. For password synchronization functionality to work properly, the other policies should be used without changes.
	Password(Pub)-Check Password GCV	Checks the GCV to determine whether you have specified that Identity Manager accepts passwords from this connected system. If not, it strips out all password elements. The name of the GCV is enable-password-publish, and the display name is I*dentity Manager accepts passwords from application*.
	Password(Pub)-Publish Distribution Password	Transforms the <`password`> element to the form that allows it to update Universal Password. This policy references the following GCVs: ◆ publish-password-to-dp ◆ enforce-password-policy
	Password(Pub)-Publish NDS Password	Allows the <`password`> element to go through if you have specified that the NDS password should be updated. If not, it strips out the <`password`> element. This policy references the GCV named publish-password-to-nds.
	Password(Pub)-Add Password Payload	Puts in payload data that is passed around in the engine for purposes of e-mail notification.
	Password(Sub)-Add Password Payload	Puts in payload data that is passed around in the engine for purposes of e-mail notification.

Policies Required in the Publisher Input Transformation Policy Set

We recommend that the Password(Pub)-Sub Email Notifications policy be listed last if there are multiple policies in the Input Transformation.

Table 5-8 Policies Required in the Publisher Input Transformation Policy Set

Location in the Driver Configuration	Password Synchronization Policy Name	What the Policy Does
Publisher Input Transformation	Password(Pub)-Sub Email Notifications	If the password payload information comes through, and the status shows a problem, it sends e-mail to the user. It uses the e-mail address indicated in the Internet EMail Address attribute in eDirectory. This policy references the GCV named notify-user-on-password-dist-failure to determine whether to send notification e-mails.

Policies Required in the Subscriber Command Transformation Policy Set

The policies listed in the Password Synchronization Policy Name column must be present in the order listed. Also, they must be the last policies in the Subscriber Command Transformation policy set.

Table 5-9 Policies Required in the Subscriber Command Transformation Policy Set

Location in the Driver Configuration	Password Synchronization Policy Name	What the Policy Does
Subscriber Command Transformation	Password(Sub)-Transform Distribution Password	Transforms the Universal Password to a <`password`> element.
	Password(Sub)-Default Password Policy	Adds a default password to an Add object if the Add object does not already contain a password. This policy and the Password(Pub)-Default Password Policy are the only policies that you can modify or remove. For password synchronization functionality to work properly, the other policies should be used without changes,.
	Password(Sub)-Check Password GCV	Checks the GCV to determine whether you have specified that the connected system accepts passwords. If not, it strips out all password elements. The name of the GCV is enable-password-subscribe, and the display name is *Application accepts passwords from Identity Manager data store*.
	Password(Sub)-Add Password Payload	Puts in password payload data that is passed around in the engine for purposes of e-mail notification.

Policies Required in the Subscriber Output Transformation Policy Set

We recommend that the Password(Sub)-Pub Email Notifications policy be listed last if there are multiple policies in the Output Transformation.

Table 5-10 Policies Required in the Subscriber Output Transformation Policy Set

Location in the Driver Configuration	Password Synchronization Policy Name	What the Policy Does
Subscriber Output Transformation	Password(Sub)-Pub Email Notifications	If the password payload information comes through, and the status shows a problem, it sends e-mail to the user. This policy references the GCV named notify-user-on-password-dist-failure to determine whether to send notification e-mails.

5.3.5 Filters You Install on the Connected System to Capture Passwords

For AD, NT Domain, and NIS, filters must be installed to capture the user's password.

See Section 5.9, "Setting Up Password Filters," on page 157.

5.3.6 NMAS Password Policies You Create for Users

Although you can use some features of Password Synchronization without Universal Password, NMAS password policies must be used to enable Universal Password for your users. The password policy also lets you specify Advanced Password Rules, and specify whether users' existing passwords are checked for compliance with the rules.

To use Identity Manager Password Synchronization, you must understand password policies. Password policies are explained in "Managing Passwords by Using Password Policies" in the *Password Management Administration Guide* (http://www.novell.com/documentation/password_management/index.html).

5.3.7 NMAS Login Methods

For some situations, you must have the NMAS Simple Password Login Method in place to be able to do password functions. For example, LDAP requires it.

For information about login methods, see the *Novell Modular Authentication Services (NMAS) 3.0 Administration Guide* (http://www.novell.com/documentation/nmas30/index.html).

5.4 Preparing to Use Identity Manager Password Synchronization and Universal Password

- "Switching Users from NDS Password to Universal Password" on page 112
- "Helping Users Change Passwords" on page 112
- "Preparing to Use Universal Password" on page 113
- "Matching Containers" on page 114
- "Setting Up E-Mail Notification" on page 114

5.4.1 Switching Users from NDS Password to Universal Password

When you turn on Universal Password for a group of users by using a password policy, the Universal Password must be populated.

If you have previously been using Password Synchronization to update the NDS password, you need to plan for the transition of users' passwords. To switch to using Universal Password, you can do one of the following to have your users create a Universal Password:

- If you use the Novell Client, roll out the Novell Client that supports Universal Password.

  The Novell Client is not required for Identity Manager Password Synchronization.

  After you roll out the Novell Client, the next time users log in by using the Novell Client, it captures the NDS password before it is hashed, and uses it to populate the Universal Password. (See "Planning Login and Change Password Methods for your Users" in the Password Management guide.)
- If you are not using the Novell Client, have users log in to the iManager self-service console. That login method populates Universal Password. To access the iManager self-service console, go to /nps on your iManager server. For example, https://www.myiManager.com/nps.
- Have users log in by using any service that authenticates by using a Universal-Password-enabled LDAP server. For example, log in through a company portal.

5.4.2 Helping Users Change Passwords

When a user changes a password in iManager, the iManager self-service console, or the Novell Client, the Advanced Password Rules from the NMAS password policy are displayed. Viewing the rules enables the user to create a compliant password without needing to guess at the rules.

Depending on how your password flow is set up, a user could change a password on a connected system, and the password would be synchronized to Identity Manager and other connected systems. However, the connected systems don't display the Advanced Password Rules when the user changes a password.

If you want to enforce Advanced Password Rules and avoid noncompliant passwords, it's best to require users to change the password only in the iManager self-service console or Novell Client, or at least make sure that the Advanced Password Rules are well publicized for users.

On a connected system, the user is allowed to change the password without viewing the password policy rules. Therefore, the user might not remember the rules correctly. Only the policies of the connected system itself are enforced when users first make the change. The following issues might occur for the user when creating a noncompliant password on a connected system, depending on your Identity Manager settings:

- If you have enabled the setting that enforces the policy on passwords coming in to Identity Manager from connected systems, the user's new password won't be synchronized to the Identity Vault. If you have set Identity Manager to notify users of failure, they find out by e-mail that their password didn't synchronize.

- If you also have set Identity Manager to replace noncompliant passwords on connected systems, the user cannot log in on the connected system by using the new password that he or she chose.

  Identity Manager resets the password on the connected system to the Distribution Password, which is probably the last compliant password that the user created.

5.4.3 Preparing to Use Universal Password

To prepare to use Universal Password, refer to “Deploying Universal Password” in the *Password Management Administration Guide* (http://www.novell.com/documentation/password_management/index.html). Most of the information that you need is in that chapter.

In addition, keep in mind the following:

- eDirectory 8.7.1 or later is required for using Universal Password. NetWare® 6.5 is not required.
- Identity Manager Password Synchronization relies on both Universal Password and the Distribution Password. The Distribution Password is the repository from which Identity Manager distributes passwords to connected systems. As with Universal Password, NMAS policies can be enforced on the Distribution Password.
- The Identity Manager iManager plug-ins, which ship with Identity Manager, include the Password Management plug-ins. These plug-ins enable you to create password policies and determine how you want Universal Password to be synchronized with NDS Password, Simple Password, and Distribution Password.

  These plug-ins replace the plug-ins for Universal Password that shipped with NetWare 6.5. They are described in “Managing Passwords by Using Password Policies” in the *Password Management Administration Guide* (http://www.novell.com/documentation/password_management/index.html).
- eDirectory 8.6.2 can't be used for the tree that Identity Manager is using. However, eDirectory 8.6.2 is supported for a subset of password synchronization features. Therefore, you can use eDirectory 8.6.2 for other trees if you are not yet ready to upgrade your entire environment.
- One way to reduce the impact when you are upgrading software for deploying Universal Password is to create a separate tree for Identity Manager as an Identity Vault. Many environments already use an Identity Vault for Identity Manager and the drivers.
- Universal Password gives you capabilities, such as enforcement of password policies and the ability to use special characters, that were not supported with previous password management tools.
- It's very important to update the Novell Client and other utilities, to avoid having the NDS Password get out of sync with the Universal Password (sometimes referred to as “password drift”). See “Planning Login and Change Password Methods for Your Users” in the *Password Management Administration Guide* (http://www.novell.com/documentation/password_management/index.html).
- The latest version of the Novell Client supports Universal Password, can populate Universal Password for a user when you first enable Universal Password for that user, and can display and enforce NMAS password policies when users are changing passwords.
- A connected system does not display the Advanced Password Rules that you create in a password policy. At this time, neither does the Novell Client, although it enforces them.

  It's best to require users to change the password only in the iManager self-service console.

If you allow users to change their passwords on a connected system or by using the latest version of the Novell Client, help users be successful in creating a compliant password by making sure your password policy rules are well publicized for your users.

- Make sure that administrators and help desk understand that ConsoleOne® supports Universal Password only if it is used on a NetWare 6.5 server or later, or is used on a machine that has the latest Novell Client.
- Make sure administrators and help desk users understand the implications of using utilities that support only NDS Password. These utilities can be used to log in, but they should not be used to change passwords. This measure avoids password drift.

  The *Novell Modular Authentication Services (NMAS) 3.0 Administration Guide* (http://www.novell.com/documentation/nmas30/index.html) references a TID that lists utilities and their support for Universal Password.

5.4.4 Matching Containers

NMAS password policies are assigned with a tree-centric perspective. In contrast, Password Synchronization is set up per driver. Drivers are installed on a per-server basis and can manage only those users who are in a master or read/write replica.

To get the results you expect from Password Synchronization, make sure that the containers that are in a master or read/write replica on the server running the drivers for Password Synchronization match the containers where you have assigned password policies with Universal Password enabled. Assigning a password policy to a partition root container ensures that all users in that container and subcontainers are assigned the password policy.

5.4.5 Setting Up E-Mail Notification

To use the e-mail notification feature, you must do the following:

- Use the Notification Configuration task in iManager to set up the e-mail server.
- Use the Notification Configuration task in iManager to customize the e-mail templates if desired.
- Make sure that the Identity Vault users have the Internet EMail Address attribute populated.

Follow the instructions in Section 5.12, "Configuring E-Mail Notification," on page 161.

5.5 Configuring and Synchronizing a New Driver

If you have not used Password Synchronization 1.0 in your environment, and you are creating a driver or replacing an existing configuration with a new Identity Manager configuration, set up Identity Manager Password Synchronization functionality:

1. Make sure your environment is ready to use Universal Password.

   See Section 5.4, "Preparing to Use Identity Manager Password Synchronization and Universal Password," on page 111.
2. Create a driver, or replace an existing driver's configuration with the Identity Manager configuration.

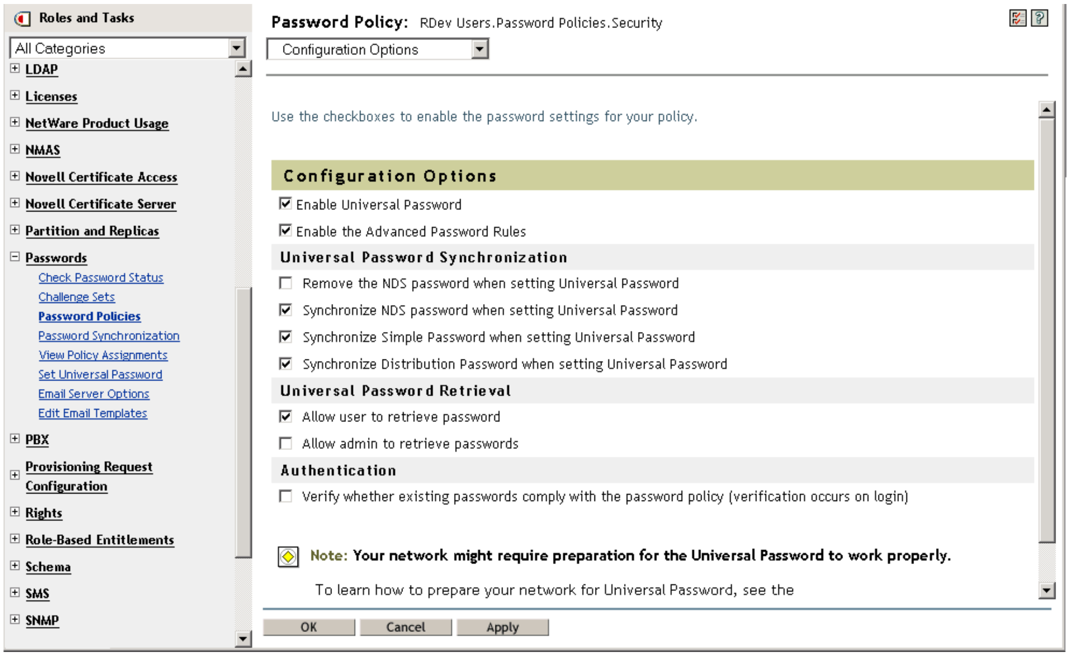
The Identity Manager configurations contain the Identity Manager policies and other items necessary for Identity Manager Password Synchronization. See the individual Identity Manager Driver Guides (http://www.novell.com/documentation/idm35drivers/) for information on importing the new sample driver configurations.

3 Turn on Universal Password for users by creating NMAS password policies with Universal Password enabled.

See “Creating Password Policies” in the *Password Management Administration Guide* (http://www.novell.com/documentation/password_management/index.html). If you previously used Universal Password with NetWare 6.5, some extra steps are described in “(NetWare 6.5 Only) Re-Creating Universal Password Assignments” in the *Password Management Administration Guide*.

We recommend that you assign password policies as high in the tree as possible.

The Configuration Options page enables you to select how you want NMAS to keep the different kinds of passwords synchronized.



Password Policy: RDev Users.Password Policies.Security

Configuration Options

Use the checkboxes to enable the password settings for your policy.

Configuration Options

☑ Enable Universal Password

☑ Enable the Advanced Password Rules

Universal Password Synchronization

□ Remove the NDS password when setting Universal Password

☑ Synchronize NDS password when setting Universal Password

☑ Synchronize Simple Password when setting Universal Password

☑ Synchronize Distribution Password when setting Universal Password

Universal Password Retrieval

☑ Allow user to retrieve password

□ Allow admin to retrieve passwords

Authentication

□ Verify whether existing passwords comply with the password policy (verification occurs on login)

Note: Your network might require preparation for the Universal Password to work properly.

To learn how to prepare your network for Universal Password, see the

For scenarios on using Password Synchronization, and how Identity Manager password policies fit in, see Section 5.8, “Implementing Password Synchronization,” on page 124. Also see the online help.

4 (Active Directory, NIS, or NT Domain only) If you want the connected systems to provide user passwords to Identity Manager, install new Password Synchronization filters and configure them.

For instructions, see the driver implementation guide for each of these drivers, at Identity Manager Drivers (http://www.novell.com/documentation/idm35drivers/).

5 For each connected system, make sure that password flow is set the way you want.

5a In iManager, click *Passwords* > *Password Synchronization*, and search for the drivers for connected systems that you want to manage.

5b View the current settings for password flow.

This is a graphical interface for the global configuration values (GCVs). Edit them by clicking the name of a driver. You can edit settings for the following:

- Whether Identity Manager accepts passwords from this system.
- Which password you want Identity Manager to update: Universal Password directly, or Distribution Password directly.

  Identity Manager controls the entry point, meaning the password that Identity Manager updates. NMAS controls the flow of passwords between each different kind of password, based on what you have set in the Configuration Options for a password policy. See the figure in Step 3 on page 115.
- Whether the password policy for the user is enforced on password changes coming in to Identity Manager.
- Whether the password policy for the user is enforced on the connected system by resetting passwords that don't comply.
- Whether passwords are accepted by this connected system.
- Whether e-mail notifications are sent when password synchronization fails.

6 Test password synchronization.

- Confirm that the Identity Manager password is distributed to the systems you specified.
- Confirm that the connected systems you specified are publishing passwords to Identity Manager.

For troubleshooting tips, see Section 5.8, “Implementing Password Synchronization,” on page 124.

5.6 Upgrading Password Synchronization 1.0

This task applies only to existing Identity Manager Drivers for Active Directory and NT Domain that are being used with Password Synchronization 1.0.

It's very important that you follow the correct procedure when upgrading from Password Synchronization 1.0.

For instructions, see the driver implementation guides for the Identity Manager Drivers for Active Directory and NT Domain, at Identity Manager Drivers (http://www.novell.com/documentation/dirxmldrivers/index.html).

5.7 Upgrading Existing Driver Configurations to Support Password Synchronization

This section explains how to add support for Identity Manager Password Synchronization to existing driver configurations, instead of replacing your existing driver configurations with the Identity Manager sample configurations.

You add support to each driver that you want to participate in password synchronization. You do this by importing an “overlay” configuration file to add the policies, driver manifest, and the GCVs, all at once.

After adding the policies, driver manifest, and GCVs, you must also add the nspmDistributionPassword attribute to the driver filter.

IMPORTANT: If you are upgrading an Identity Manager Driver for AD or NT Domain, and that driver is being used with Password Synchronization 1.0, follow the upgrade instructions in the driver implementation guides for the Identity Manager Drivers for Active Directory and NT Domain, at Identity Manager Drivers (http://www.novell.com/documentation/dirxmldrivers/index.html).

The policies added in this procedure are for supporting Password Synchronization by using Universal Password and Distribution Password. If you are using the Identity Manager driver to synchronize only the NDS Password, you should not use the policies in the Identity Manager driver configuration. NDS Password is synchronized by using Public Key and Private Key attributes instead of these policies, as described in Section 5.8.2, "Scenario 1: Using NDS Password to Synchronize between Two Identity Vaults," on page 126.

- "Step 1: Converting the Driver to Identity Manager 3.5 Format" on page 117
- "Step 2: Adding to the Driver Configuration" on page 119
- "Step 3: Changing Filter Settings" on page 121
- "Step 4: Setting Up Password Synchronization Flow" on page 123

Prerequisites

- ❑ Create a backup of your existing driver by using the Export Drivers Wizard.
- ❑ Make sure you have installed the new driver shim.

  Some password synchronization features (for example, Check Password Status) won't work without the new Identity Manager driver shim

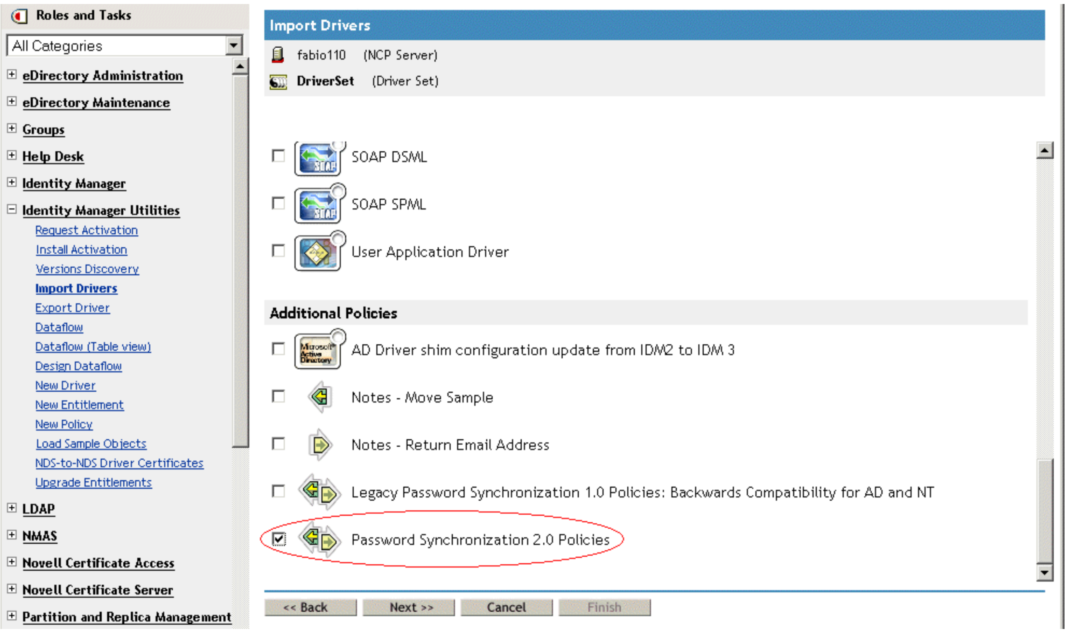
  IMPORTANT: If you are upgrading an Identity Manager Driver for AD or NT Domain, and that driver is being used with Password Synchronization 1.0, don't install the driver shim until you have reviewed the upgrade instructions. Follow the upgrade instructions in the driver implementation guides for the Identity Manager Drivers for Active Directory and NT Domain, at Identity Manager Drivers (http://www.novell.com/documentation/dirxmldrivers/index.html).

5.7.1 Step 1: Converting the Driver to Identity Manager 3.5 Format

1. Make sure that your environment is ready to use Universal Password.

   See Section 5.4, "Preparing to Use Identity Manager Password Synchronization and Universal Password," on page 111.

   If you are using DirXML® 1.1a, see Section 2.3, "Upgrading a Driver Configuration from DirXML 1.1a to Identity Manager 3.5.1 Format," on page 25.
2. In iManager, click *Identity Manager Utilities* > *Import Drivers*.
3. Select the driver set where your existing driver resides, then click *Next*.
4. In the list of driver configurations that appears, scroll to *Additional Policies*, then select only *Password Synchronization 2.0 Policies*.



Import Drivers

fabio110 (NCP Server)
DriverSet (Driver Set)

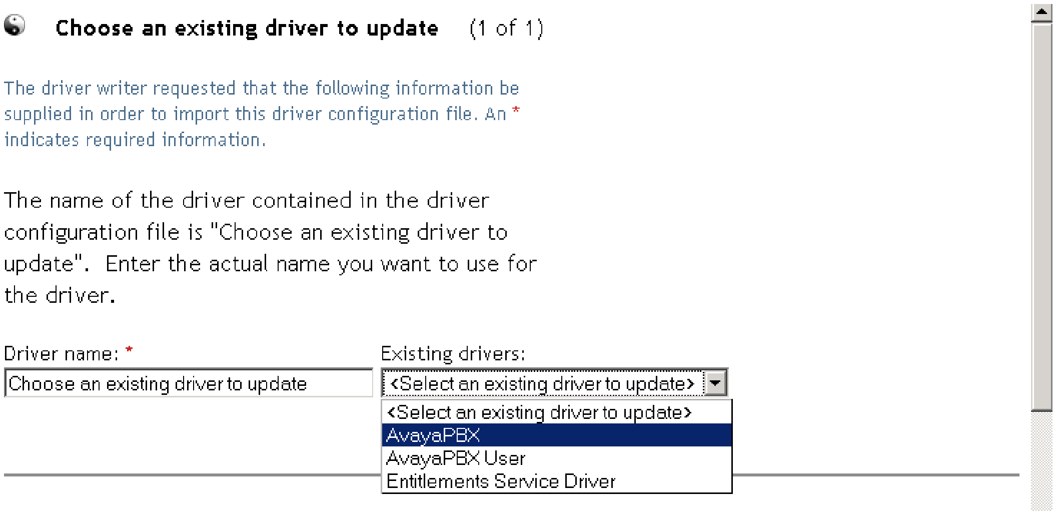
□ SOAP DSML
□ SOAP SPML
□ User Application Driver

Additional Policies

□ AD Driver shim configuration update from IDM2 to IDM 3
□ Notes - Move Sample
□ Notes - Return Email Address
□ Legacy Password Synchronization 1.0 Policies: Backwards Compatibility for AD and NT
☑ Password Synchronization 2.0 Policies

5 Click *Next*.

6 In the *Existing drivers* drop-down list, select your existing driver to update.


Choose an existing driver to update (1 of 1)

The driver writer requested that the following information be supplied in order to import this driver configuration file. An * indicates required information.

The name of the driver contained in the driver configuration file is "Choose an existing driver to update". Enter the actual name you want to use for the driver.

Driver name: * Choose an existing driver to update

Existing drivers: <Select an existing driver to update>

<Select an existing driver to update>
AvayaPBX
AvayaPBX User
Entitlements Service Driver

7 In the *Connected System* drop-down list, select the connected system type.

If the driver name doesn’t appear in the drop-down list, select *Other Systems*.

Based on the type of driver, the Import Driver Wizard makes entries in the driver manifest that indicate the capabilities of the driver configuration and the connected system:

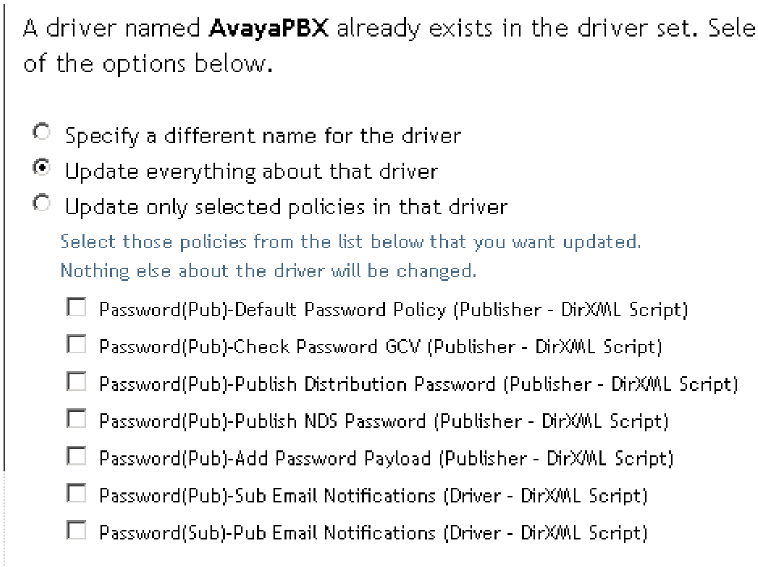
- Whether the connected system can provide passwords to Identity Manager.

  This refers to the users's actual password on the connected system, not to a password that can be created using a style sheet. Only AD, eDirectory, and NIS can do this.

- Whether the connected system can accept passwords from Identity Manager
- Whether the connected system can check a password to see if it matches the password in Identity Manager.

Correct entries in the driver manifest are required for Password Synchronization policies to work. The driver manifest indicates the combined ability of the connected system, the Identity Manager driver shim, and the driver configuration policies, and should not usually be edited by the network administrator.

8 Click *Next*.



A driver named **AvayaPBX** already exists in the driver set. Select one of the options below.

○ Specify a different name for the driver
● Update everything about that driver
○ Update only selected policies in that driver

Select those policies from the list below that you want updated.
Nothing else about the driver will be changed.

□ Password(Pub)-Default Password Policy (Publisher - DirXML Script)
□ Password(Pub)-Check Password GCV (Publisher - DirXML Script)
□ Password(Pub)-Publish Distribution Password (Publisher - DirXML Script)
□ Password(Pub)-Publish NDS Password (Publisher - DirXML Script)
□ Password(Pub)-Add Password Payload (Publisher - DirXML Script)
□ Password(Pub)-Sub Email Notifications (Driver - DirXML Script)
□ Password(Sub)-Pub Email Notifications (Driver - DirXML Script)

9 If you don’t have driver manifest or GCV values that you want to save, select *Update everything about that driver*.

This option gives you the driver manifest, global configuration values (GCVs), and Identity Manager policies necessary for password synchronization.

The driver manifest and GCVs overwrite any values that already exist. Because these kinds of driver parameters were new in Identity Manager 2, a DirXML 1.*x* driver should have no existing values to be overwritten.

The password synchronization policies don't overwrite any existing policy objects. They are simply added to the Driver object.

NOTE: If you have driver manifest or GCV values that you want to save, select *Update only selected policies in that driver*, and select the check boxes for all the policies. This option imports the password policies but does not change the driver manifest or GCVs. You need to manually paste in any additional values.

10 Click *Next*, then click *Finish* to complete the wizard.

At this point, the new policies have been created as policy objects under the Driver object, but are not yet part of the driver configuration. To link them in, you must manually insert each of them at the right point in the driver configuration on the Subscriber and Publisher channels.

5.7.2 Step 2: Adding to the Driver Configuration

For a list of the policies you add, and where to insert them, see Section 5.3.4, “Policies Required in the Driver Configuration,” on page 107.

Insert each of the new policies into the correct place on your existing driver configuration.

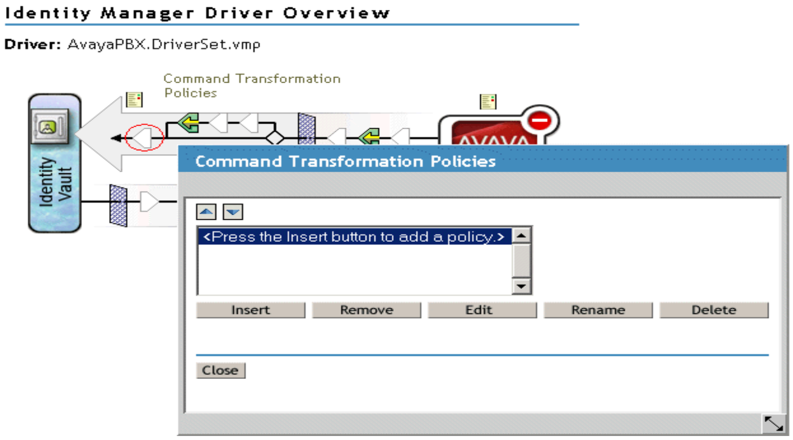
If the policy set has multiple policies, make sure these Identity Manager password synchronization policies are listed last.

Repeat the following steps for each policy.

1 Select *Identity Manager* > I*dentity Manager Overview*, then search for the driver set that contains the driver you are updating.

2 Click the driver that you just updated (for example, AvayaPBX).

3 Click the icon (for example, Command Transformation Policies on the Publisher channel) for the place where you need to add one of the new policies.

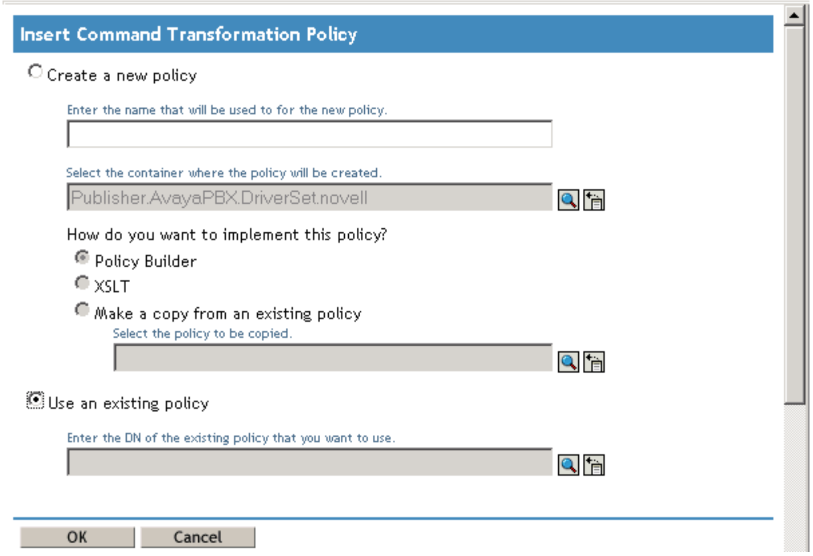


Identity Manager Driver Overview

Driver: AvayaPBX.DriverSet.vmp

Command Transformation Policies
Identity Vault
AVAYA
Command Transformation Policies
<Press the Insert button to add a policy.>
Insert
Remove
Edit
Rename
Delete
Close

4 Click *Insert* to add the new policy.



Insert Command Transformation Policy

○ Create a new policy

Enter the name that will be used to for the new policy.

Select the container where the policy will be created.

Publisher.AvayaPBX.DriverSet.novell

How do you want to implement this policy?

● Policy Builder

○ XSLT

○ Make a copy from an existing policy

Select the policy to be copied.

● Use an existing policy

Enter the DN of the existing policy that you want to use.

5 Click *Use an existing policy*, browse for the new policy object, then click *OK*.

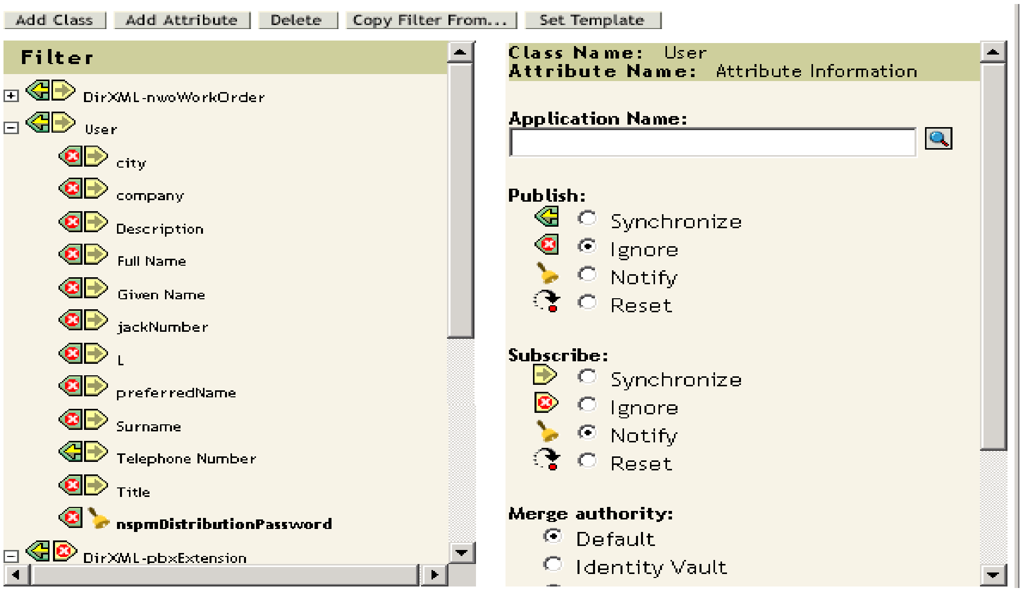
6 If you have more than one policy in the list for any of the new policies, use the arrow buttons ▲ ▼ to move the new policies to the correct location in the list.

Make sure that the policies are in the order listed in Section 5.3.4, “Policies Required in the Driver Configuration,” on page 107.

5.7.3 Step 3: Changing Filter Settings

1 For the object classes that you want to synchronize passwords for (such as User), make sure that nspmDistributionPassword attribute is in the filter and has the following settings:

- For the Publisher channel, set the filter to *Ignore* for the *nspmDistributionPassword* attribute.
- For the Subscriber channel, set the filter to *Notify* for the *nspmDistribution Password* attribute.



Class Name: User
Attribute Name: Attribute Information

Application Name:

Publish:
○ Synchronize
● Ignore
○ Notify
○ Reset

Subscribe:
○ Synchronize
○ Ignore
● Notify
○ Reset

Merge authority:
● Default
○ Identity Vault

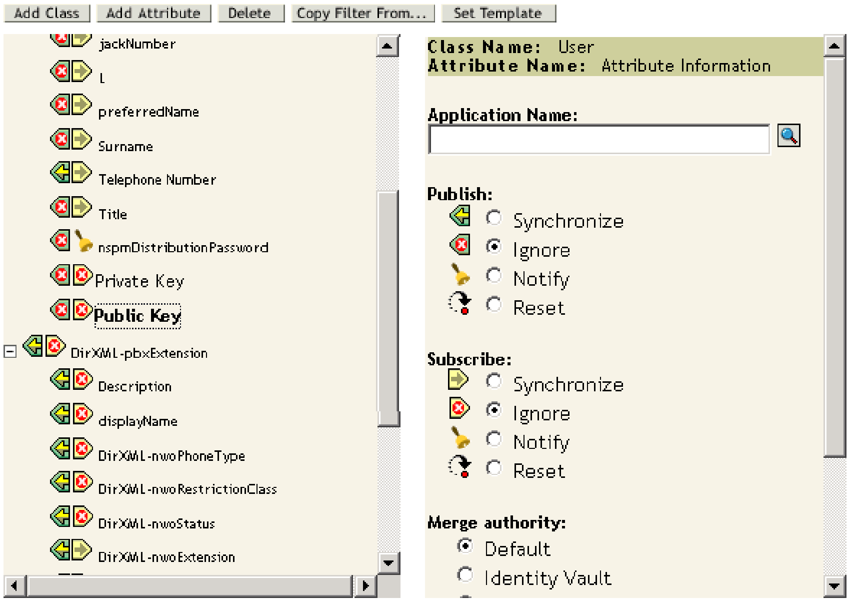
To view the attribute, you might need to scroll to and select the class (for example, User), then scroll through the attributes.

If the nspmDistributionPassword isn't listed:

1a Make sure that the class is selected, then click *Add Attribute*.

1b Scroll to and select nspmDistributionPassword, then click OK.

2 For all objects that have *Notify* set for the *nspmDistributionPassword* attribute, set both the Public Key and Private Key attributes to Ignore.



Class Name: User
Attribute Name: Attribute Information

Application Name:

Publish:
○ Synchronize
● Ignore
○ Notify
○ Reset

Subscribe:
○ Synchronize
● Ignore
○ Notify
○ Reset

Merge authority:
● Default
○ Identity Vault

3 For each driver that you want to upgrade to participate in password synchronization, repeat Step 2 on page 117 (in “Converting the Driver to Identity Manager 3.5 Format”) through Step 2 in this section (“Change Filter Settings”).

At this point, the driver has the new driver shim, Identity Manager format, and the other elements that are necessary in the driver configuration to support password synchronization: driver manifest, GCVs, password synchronization policies, and filter settings.

4 Check the individual driver implementation guides for any additional steps or information on setting up Identity Manager Password Synchronization. See Identity Manager Drivers (http://www.novell.com/documentation/lg/dirxmldrivers/index.html).

5 Turn on Universal Password for users by creating password policies with Universal Password enabled.

See “Creating Password Policies” in the *Password Management Administration Guide* (http://www.novell.com/documentation/password_management/index.html). If you previously used Universal Password with NetWare 6.5, some extra steps are described in “(NetWare 6.5 only) Re-Creating Universal Password Assignments” in the *Password Management Administration Guide.*

We recommend that you assign password policies as high in the tree as possible.

The Configuration Options page has options for how you want NMAS to keep the different kinds of passwords synchronized. The default settings should work for most implementations. For more information, see the online help for that page.

For scenarios on using Password Synchronization, and how password policies fit in, see Section 5.8, “Implementing Password Synchronization,” on page 124.

NMAS password policies are assigned with a tree-centric perspective. In contrast, Password Synchronization is set up per driver. Drivers are installed on a per-server basis and can manage only those users who are in a master or read/write replica.

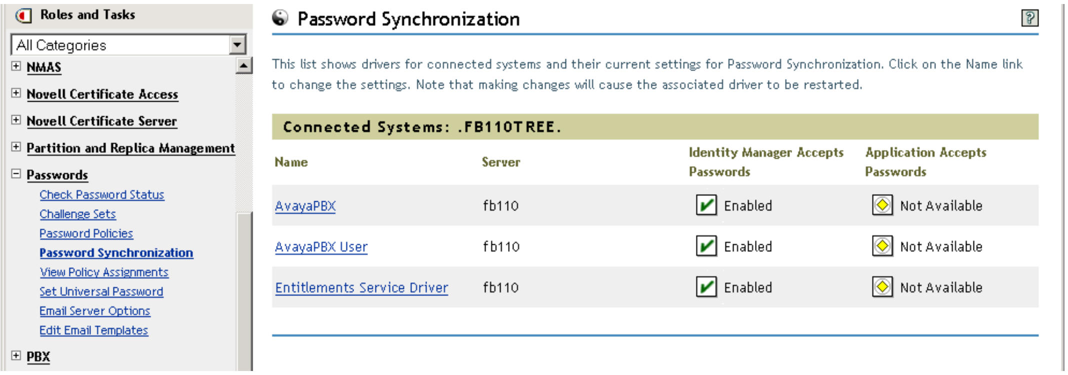
To get the results you expect from Password Synchronization, make sure that the containers in a master or read/write replica on the server running the drivers for Password Synchronization match the containers where you have assigned password policies with Universal Password enabled. Assigning a password policy to a partition root container ensures that all users in that container and subcontainers are assigned the password policy.

5.7.4 Step 4: Setting Up Password Synchronization Flow

Make sure that your password flow is set the way you want it for each connected system.

1 In iManager, select *Passwords* > *Password Synchronization*.

2 Search a tree or container for the drivers for connected systems that you want to manage.



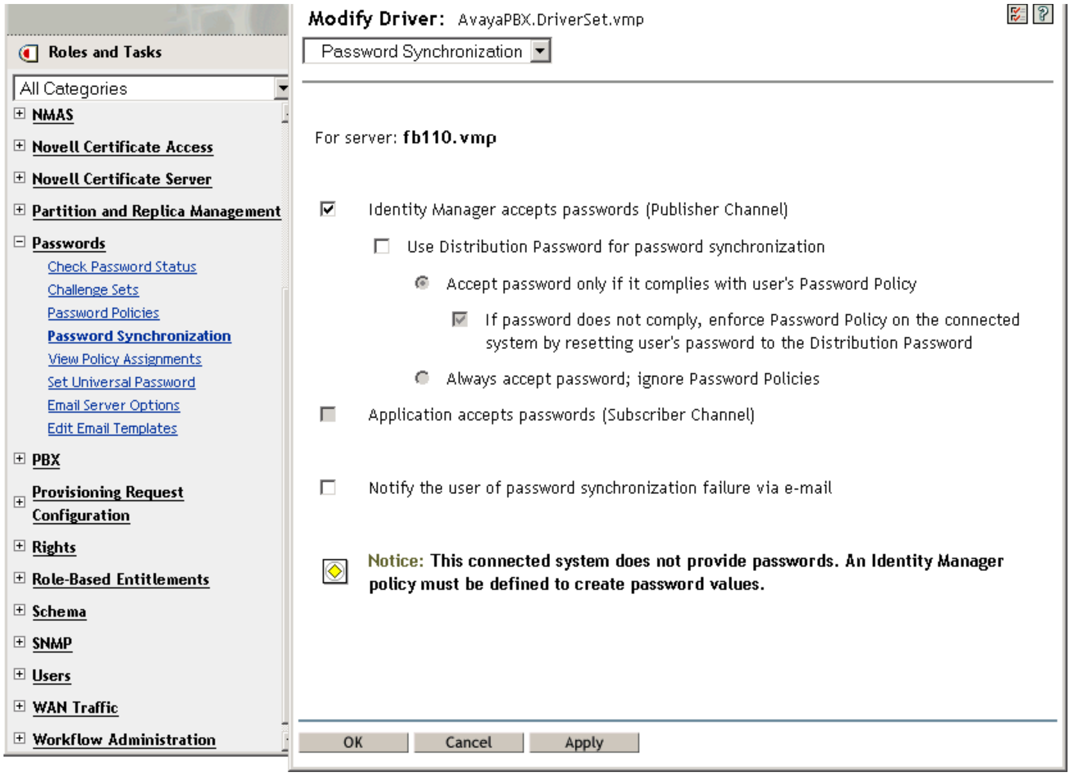
Password Synchronization

This list shows drivers for connected systems and their current settings for Password Synchronization. Click on the Name link to change the settings. Note that making changes will cause the associated driver to be restarted.

Connected Systems: .FB110TREE.

Name	Server	Identity Manager Accepts Passwords	Application Accepts Passwords
AvayaPBX	fb110	☑ Enabled	◇ Not Available
AvayaPBX User	fb110	☑ Enabled	◇ Not Available
Entitlements Service Driver	fb110	☑ Enabled	◇ Not Available

3 View the current settings for password flow by selecting a driver



Modify Driver: AvayaPBX.DriverSet.vmp

Password Synchronization

For server: **fb110.vmp**

☑ Identity Manager accepts passwords (Publisher Channel)

□ Use Distribution Password for password synchronization

● Accept password only if it complies with user's Password Policy

☑ If password does not comply, enforce Password Policy on the connected system by resetting user's password to the Distribution Password

○ Always accept password; ignore Password Policies

□ Application accepts passwords (Subscriber Channel)

□ Notify the user of password synchronization failure via e-mail

Notice: **This connected system does not provide passwords. An Identity Manager policy must be defined to create password values.**

This page lists the global configuration values (GCVs). Change them by selecting options.

Identity Manager controls the entry point (which password Identity Manager updates). NMAS controls the flow of passwords between each different kind of password, based on the options you set in Configuration Options. (Step 3 on page 106 displays the Configuration Options page.) If you select *Use Distribution Password for password synchronization*, Identity Manager uses Distribution Password directly. If you deselect this option, Identity Manager uses Universal Password directly.

For information (including illustrations) on these options, see Section 5.8, “Implementing Password Synchronization,” on page 124. Also see the online help.

4 Test password synchronization.

Confirm that the Identity Manager password is distributed to the systems you specified.

Confirm that the connected systems you specified are publishing passwords to Identity Manager.

For troubleshooting tips, see Section 5.8, “Implementing Password Synchronization,” on page 124.

5.8 Implementing Password Synchronization

The Password Synchronization functionality provided in Identity Manager enables you to implement several different scenarios. This section outlines basic scenarios, to help you understand how the settings in Identity Manager Password Synchronization and NMAS password policies affect the way passwords are synchronized. You can use one or more of the scenarios to meet the needs of your environment.

- Section 5.8.1, “Overview of Identity Manager’s Relationship to NMAS,” on page 125

- Section 5.8.2, “Scenario 1: Using NDS Password to Synchronize between Two Identity Vaults,” on page 126
- Section 5.8.3, “Scenario 2: Using Universal Password to Synchronize Passwords,” on page 129
- Section 5.8.4, “Scenario 3: Synchronizing an Identity Vault and Connected Systems, with Identity Manager Updating the Distribution Password,” on page 140
- Section 5.8.5, “Scenario 4: Tunneling,” on page 149
- “Scenario 5: Synchronizing Application Passwords to the Simple Password” on page 153

5.8.1 Overview of Identity Manager’s Relationship to NMAS

- “Utilities and NMAS” on page 125
- “Identity Manager and NMAS” on page 126

Utilities and NMAS

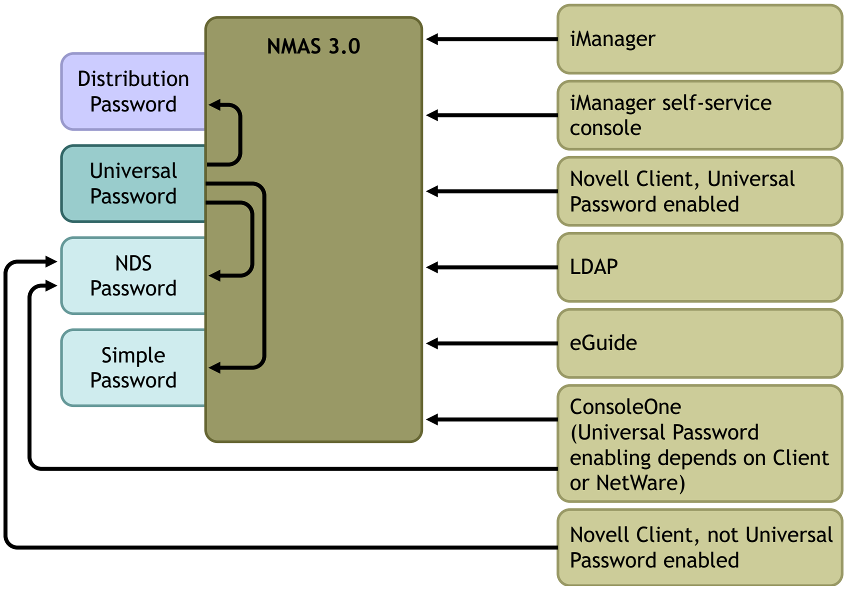
Utilities such as iManager and the Novell Client communicate with NMAS rather than directly updating a specific password. NMAS is the entity that determines which passwords are updated.

NMAS synchronizes passwords within an Identity Vault, based on your settings in NMAS password policies.

Legacy utilities that are not Universal Password-enabled update the NDS password directly, instead of communicating with NMAS and letting NMAS determine which passwords are updated. Be aware of how users and help desk administrators use legacy utilities in your environment. Because legacy utilities update the NDS password directly instead of going through NMAS, password drift (Universal Password and NDS password get out of sync) can occur if you are using Universal Password and NMAS 2.3.

For example, to ensure support of Universal Password, make sure that users upgrade to the Novell Client, and make sure that help desk users use ConsoleOne only with the latest Novell Client or NetWare release.

Figure 5-5 Using NMAS to Synchronize Passwords


NMAS 3.0
Distribution Password
Universal Password
NDS Password
Simple Password
iManager
iManager self-service console
Novell Client, Universal Password enabled
LDAP
eGuide
ConsoleOne (Universal Password enabling depends on Client or NetWare)
Novell Client, not Universal Password enabled

Identity Manager and NMAS

Identity Manager controls the "entry point" (updating either Universal or Distribution Password directly). NMAS controls the flow of synchronizing passwords inside the Identity Vault.

In Scenario 1, the Identity Manager Driver for eDirectory can be used to update the NDS password directly. This scenario is basically the same as the one provided in DirXML 1.*x*.

In Scenario 2, Scenario 3, and Scenario 4, Identity Manager is used to update either Universal Password or Distribution Password. Identity Manager goes through NMAS to make password changes. This allows NMAS to update other Identity Vault passwords as determined by NMAS password policy settings, and allows NMAS to enforce Advanced Password Rules from NMAS password policies for passwords being synchronized with connected systems. In these scenarios, the password that Identity Manager distributes to connected systems is always the Distribution Password.

The difference between Scenario 2, Scenario 3, and Scenario 4 lies in the different combinations of NMAS password policy settings and Identity Manager Password Synchronization settings for each connected system driver.

5.8.2 Scenario 1: Using NDS Password to Synchronize between Two Identity Vaults

As in Password Synchronization 1.0, you can synchronize NDS Password between two Identity Vaults by using the eDirectory driver. This scenario does not require Universal Password to be implemented, and can be used with eDirectory 8.6.2 or later. Another name for this kind of password synchronization is synchronizing the public/private key pair.

This method should be used only to synchronize passwords from Identity Vault to Identity Vault. It does not use NMAS and therefore cannot be used to synchronize passwords to connected applications.

- “Advantages and Disadvantages of Scenario 1” on page 127
- “Setting Up Scenario 1” on page 128
- “Troubleshooting Scenario 1” on page 129

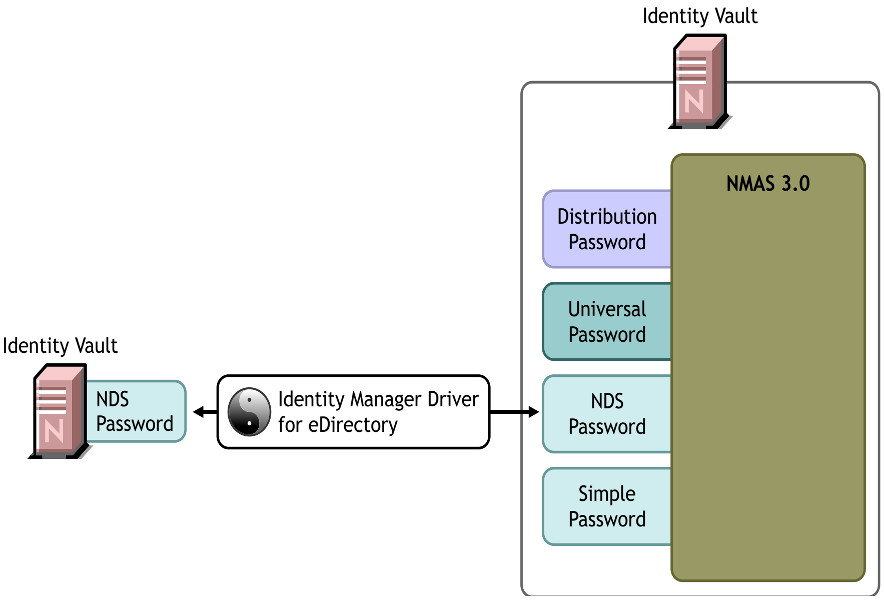
Advantages and Disadvantages of Scenario 1

Table 5-11 *Advantages: eDirectory to eDirectory Password Synchronization Using NDS Password*

Advantages	Disadvantages
Simple configuration. Just include the correct attributes in the driver filter. If you are deploying Identity Manager and eDirectory 8.7.3 in stages, this method can help you deploy gradually. ◆ You don't need to add the new password synchronization policies to driver configurations. ◆ Does not require Universal Password to be implemented in the Identity Vault. ◆ Can be used with connected vaults running eDirectory 8.6.2 or later. ◆ Does not require NMAS 2.3. Enforces the basic password restrictions you can set for NDS Password.	This method synchronizes passwords between Identity Vaults. Passwords cannot be synchronized to other connected systems. Does not update Universal Password or Distribution Password. Because this method does not use NMAS, you can't validate passwords against Advanced Password Rules in password policies for passwords coming from another Identity Vault. Because this method does not use NMAS, you can't reset passwords on the connected Identity Vault if the passwords don't comply with the NMAS password policy. E-mail notifications are not provided for password synchronization failures. Check Password Status operations from the iManager task are not supported. (Distribution Password is required for this feature.)

The following diagram shows that, as in DirXML 1.*x*, the Identity Manager Driver for eDirectory can be used to synchronize the NDS password between two Identity Vaults. This scenario does not go through NMAS.

Figure 5-6 Using NDS Password to Synchronize between Two Identity Vaults



Setting Up Scenario 1

To set up this kind of password synchronization, configure the driver.

Universal Password Deployment

Not necessary.

Password Policy Configuration

None.

Password Synchronization Settings

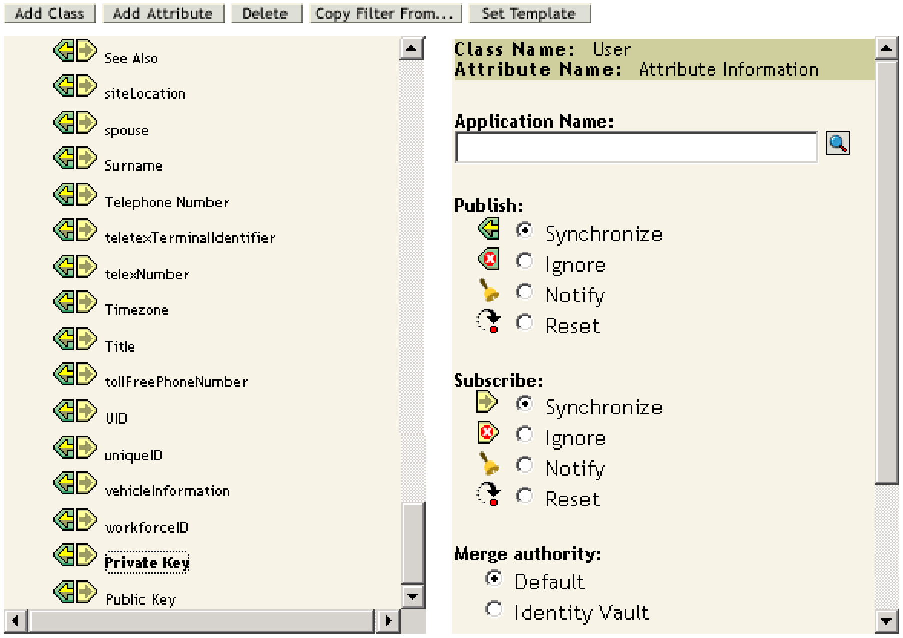
None. The settings on the Password Synchronization page for a driver have no effect on this method of synchronizing NDS Password.

Driver Configuration

Remove the Password Synchronization policies listed in Section 5.3.4, "Policies Required in the Driver Configuration," on page 107. Those policies are intended to support Universal Password and Distribution Password. NDS Password is synchronized by using Public Key and Private Key attributes instead of these policies.

Make sure that the driver filter for both Identity Vault drivers is synchronizing the Public Key and Private Key attributes for all object classes for which passwords should be synchronized. The following figure shows an example.

Figure 5-7 *Synchronizing the Private and Public Key Attributes*



Class Name: User
Attribute Name: Attribute Information

Application Name:

Publish:
● Synchronize
○ Ignore
○ Notify
○ Reset

Subscribe:
● Synchronize
○ Ignore
○ Notify
○ Reset

Merge authority:
● Default
○ Identity Vault

Troubleshooting Scenario 1

- Turn on the DSTrace option.
- Check the driver Filter to make sure the Public Key and Private Key attributes are being synchronized, not ignored.
- See also the tips in Section 5.13, "Troubleshooting Password Synchronization," on page 173.

5.8.3 Scenario 2: Using Universal Password to Synchronize Passwords

With Identity Manager, you can synchronize a connected system password with the Universal Password in the Identity Vault.

When Universal Password is updated, the NDS Password, Distribution Password, or Simple Password can also be updated, depending on your settings in the NMAS password policy.

Any connected system can publish passwords to Identity Manager, though not all connected systems can provide the user's actual password. For example, Active Directory can publish a user's actual password to Identity Manager. Although PeopleSoft does not provide a password from the PeopleSoft system itself, it can provide an initial password created in a policy in the driver configuration, such as a password based on the user's employee ID or last name. Not all drivers can subscribe to password changes from Identity Manager. See Section 5.2, "Connected System Support for Password Synchronization," on page 100.

- "Advantages and Disadvantages of Scenario 2" on page 130
- "Setting Up Scenario 2" on page 131
- "Troubleshooting Scenario 2" on page 136

Advantages and Disadvantages of Scenario 2

Table 5-12 *Advantages: Synchronizing by Using Universal Password*

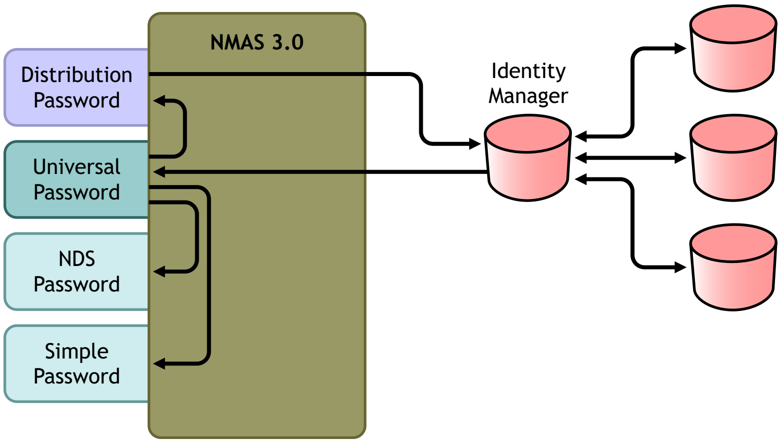
Advantages	Disadvantages
Allows synchronization of passwords to and from the Identity Vault and the connected system. Allows passwords to be validated against the NMAS password policy. Allows e-mail notifications for failed password operations, such as when a password coming from a connected system does not comply with Password. Supports the Check Password Status task in iManager, if Universal Password is being synchronized with Distribution Password and if the connected system supports checking passwords. NMAS enforces the Advanced Password Rules in your password policies, if you have the rules enabled. If a password coming from a connected system does not comply, an error is generated, and an e-mail notification is sent if you have specified that option. If you don't want password policy rules enforced, you can deselect *Enable Advanced Password Rules* in the NMAS password policy.	By design, resetting passwords in the connected system is not supported with this method because the Distribution Password and Universal passwords might not be the same, depending on your settings in the password policies.

Figure 5-8 illustrates the following flow for this scenario:

1. Passwords come in through Identity Manager.
2. Identity Manager goes through NMAS to directly update Universal Password.
3. NMAS synchronizes the Universal Password with the Distribution Password and other passwords according to the NMAS password policy settings.
4. Identity Manager retrieves the Distribution Password to distribute to connected systems that are set to accept passwords.

Although multiple connected systems are shown as connecting to Identity Manager in this figure, keep in mind that you individually create the settings for each connected system driver.

Figure 5-8 Using Universal Password to Synchronize Passwords



Setting Up Scenario 2

To set up this kind of password synchronization:

- "Universal Password Deployment" on page 131
- "Password Policy Configuration" on page 131
- "Password Synchronization Settings" on page 133
- "Driver Configuration" on page 134

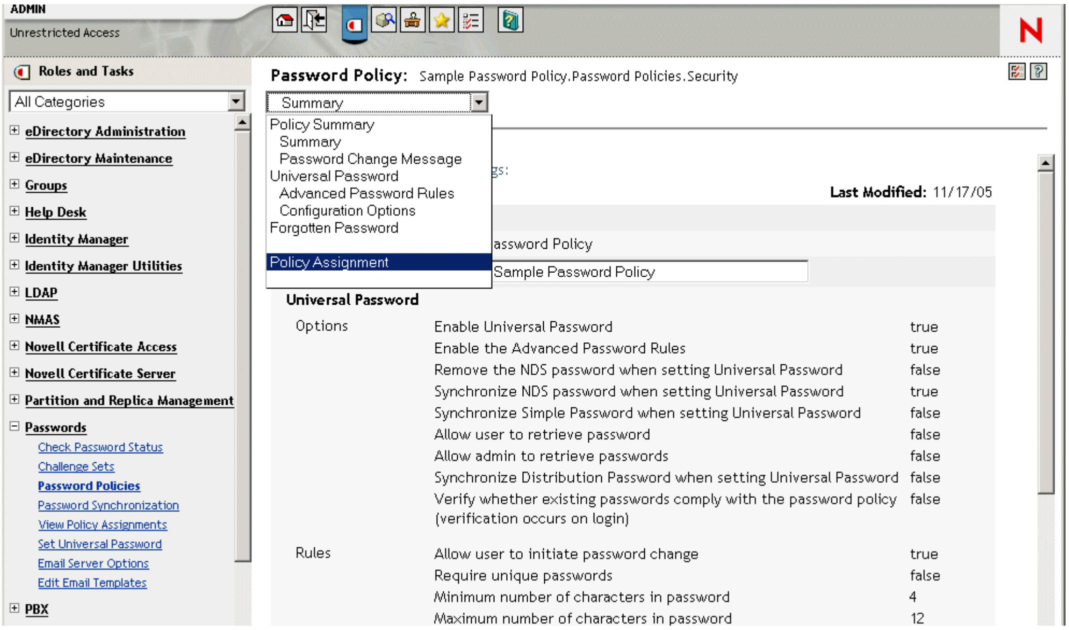
Universal Password Deployment

Make sure your environment is ready to use Universal Password. See Section 5.4, "Preparing to Use Identity Manager Password Synchronization and Universal Password," on page 111.

Password Policy Configuration

Make sure that an NMAS password policy is assigned to the parts of the Identity Vault that you want to have this kind of password synchronization.

1. In iManager, select *Passwords* > *Password Policies*.
2. Select a policy, then click *Edit*.
3. Browse to and select the object where you want password synchronization to occur.



Password Policy: Sample Password Policy.Password Policies.Security

gs:

Last Modified: 11/17/05

assword Policy

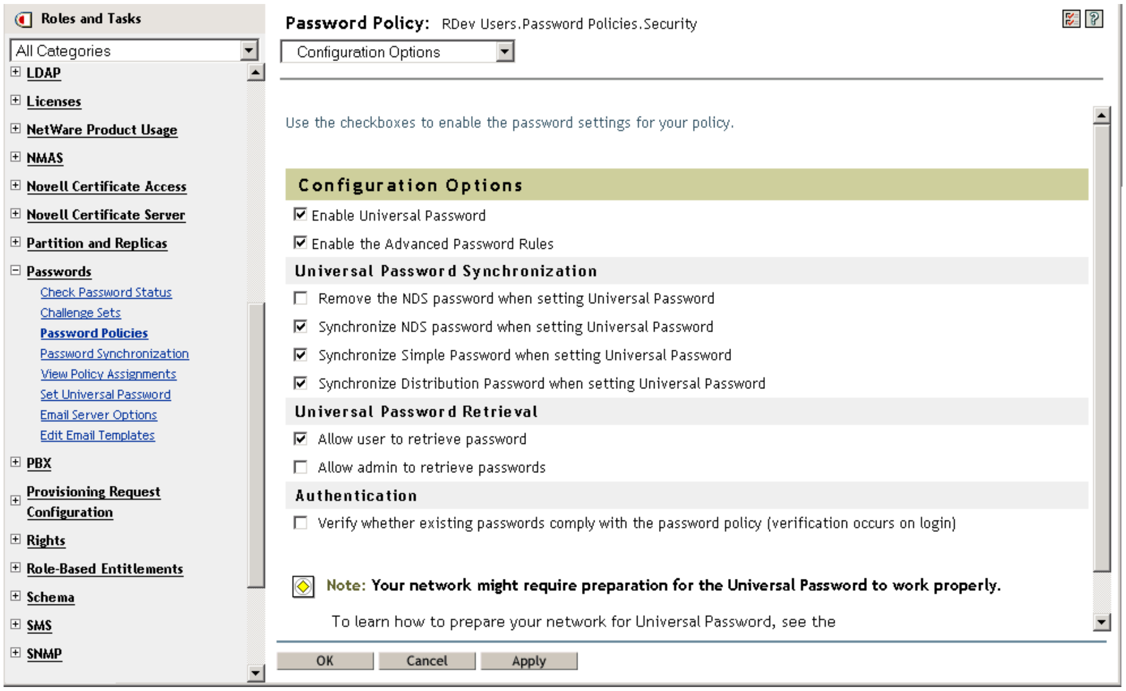
Sample Password Policy

Universal Password

Options	Enable Universal Password	true
	Enable the Advanced Password Rules	true
	Remove the NDS password when setting Universal Password	false
	Synchronize NDS password when setting Universal Password	true
	Synchronize Simple Password when setting Universal Password	false
	Allow user to retrieve password	false
	Allow admin to retrieve passwords	false
	Synchronize Distribution Password when setting Universal Password	false
	Verify whether existing passwords comply with the password policy (verification occurs on login)	false
Rules	Allow user to initiate password change	true
	Require unique passwords	false
	Minimum number of characters in password	4
	Maximum number of characters in password	12

You can assign the policy to the entire tree structure (by browsing to and selecting the Login Policy object in the Security container), a partition root container, a container, or a specific user. To simplify management, we recommend that you assign password policies as high in the tree as possible.

4 In the password policy, make sure that the following are selected:



Password Policy: RDev Users.Password Policies.Security

Use the checkboxes to enable the password settings for your policy.

Configuration Options

☑ Enable Universal Password

☑ Enable the Advanced Password Rules

Universal Password Synchronization

☐ Remove the NDS password when setting Universal Password

☑ Synchronize NDS password when setting Universal Password

☑ Synchronize Simple Password when setting Universal Password

☑ Synchronize Distribution Password when setting Universal Password

Universal Password Retrieval

☑ Allow user to retrieve password

☐ Allow admin to retrieve passwords

Authentication

☐ Verify whether existing passwords comply with the password policy (verification occurs on login)

Note: Your network might require preparation for the Universal Password to work properly.

To learn how to prepare your network for Universal Password, see the

- *Enable Universal Password*
- *Synchronize NDS Password when setting Universal Password*

- *Synchronize Distribution Password when setting Universal Password*

  Because Identity Manager retrieves the Distribution Password to distribute passwords to connected systems, it's important that this option be selected to allow bidirectional password synchronization.

5 Complete your password policy as desired.

NMAS enforces the Advanced Password Rules in your password policies, if you have the rules enabled. If you don't want password policy rules enforced, deselect *Enable the Advanced Password Rules*.

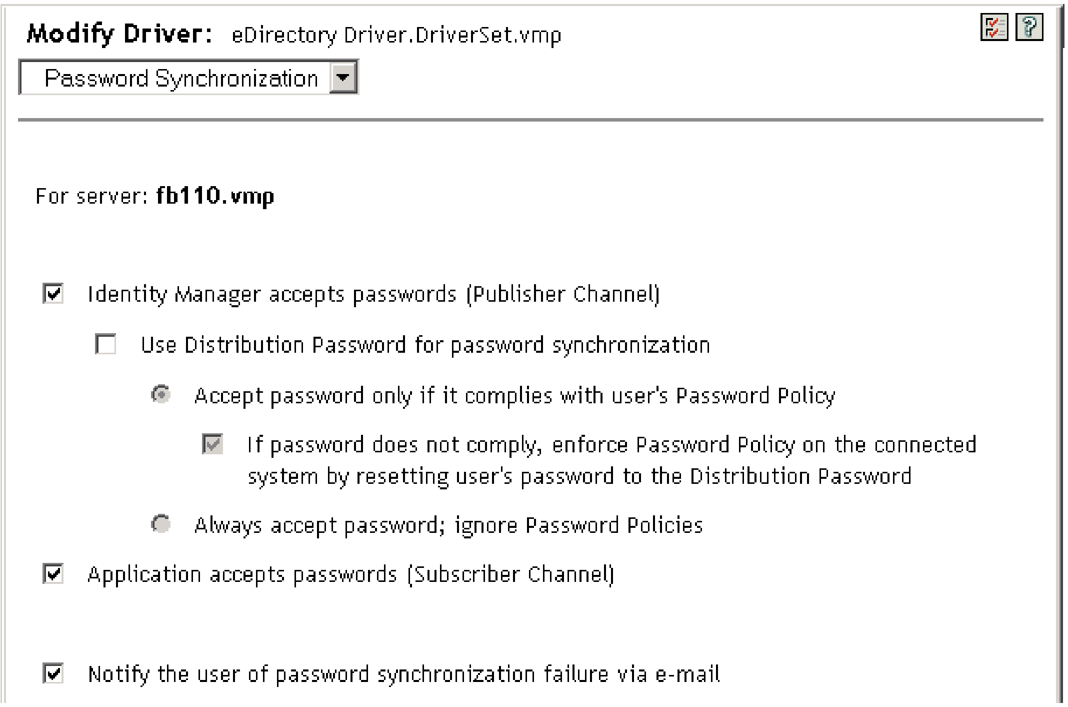
If you are using Advanced Password Rules, make sure they don't conflict with the password policies on any connected systems that are subscribing to passwords.

Password Synchronization Settings

1 In iManager, select *Passwords* > *Password Synchronization*.

2 Search for drivers for the connected systems, then select a driver.

3 Create settings for the driver for the connected system.



Modify Driver: eDirectory Driver.DriverSet.vmp

Password Synchronization ▼

For server: **fb110.vmp**

☑ Identity Manager accepts passwords (Publisher Channel)

□ Use Distribution Password for password synchronization

● Accept password only if it complies with user's Password Policy

☑ If password does not comply, enforce Password Policy on the connected system by resetting user's password to the Distribution Password

○ Always accept password; ignore Password Policies

☑ Application accepts passwords (Subscriber Channel)

☑ Notify the user of password synchronization failure via e-mail

Make sure that the following are selected:

- *Identity Manager accepts passwords (Publisher Channel)*

  A message is displayed on the page if the driver manifest does not contain a “password-publish” capability. This is to inform users that passwords cannot be retrieved from the application and can only be published by creating a password in a the driver configuration using a policy.

- *Application accepts passwords (Subscriber Channel)*

  If the connected system does not support accepting passwords, the option is dimmed.

These settings allow for bidirectional password synchronization if it is supported by the connected system.

You can adjust the settings to match your business policies for the authoritative source for passwords. For example, if a connected system should subscribe to passwords but not publish, select only *Application accepts passwords (Subscriber Channel)*.

4 Make sure that *Use Distribution Password for password synchronization* is not selected.

In this scenario, Identity Manager updates the Universal Password directly. The Distribution Password is still used to distribute passwords to connected systems, but is updated from the Universal Password by NMAS instead of by Identity Manager.

5 (Optional) Select the following if desired:

- *Notify the user of password synchronization failure via e-mail*

  Keep in mind that e-mail notifications require the Internet EMail Address attribute on the eDirectory User object to be populated.

  E-mail notifications are non-invasive. They do not affect the processing of the XML document that triggered the e-mail. If they fail, they are not retried unless the operation itself is retried. However, debug messages for e-mail notifications are written to the trace file.

Driver Configuration

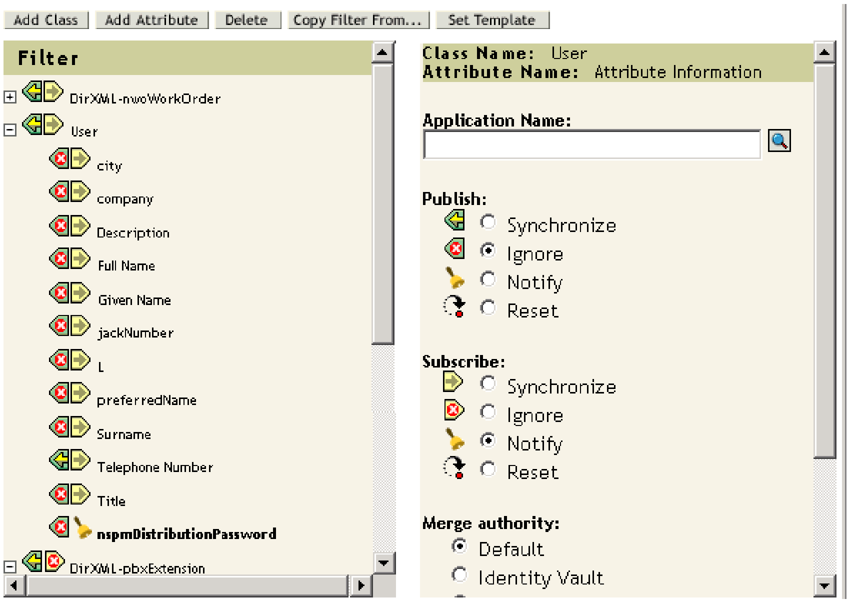
1 Make sure that the required Identity Manager script password synchronization policies are included in the driver configurations for each driver that should participate in password synchronization.

The policies must be in the correct location and the correct order in the driver configuration. For the list of policies, see Section 5.3.4, “Policies Required in the Driver Configuration,” on page 107.

The Identity Manager sample configurations already contain the policies. If you are upgrading an existing driver, you can add the policies by using the instructions in Section 5.7, “Upgrading Existing Driver Configurations to Support Password Synchronization,” on page 116.

2 Set the filter correctly for nspmDistributionPassword attribute:

- For the Publisher channel, set the driver filter to *Ignore for the nspmDistributionPassword* attribute for all object classes.
- For the Subscriber channel, set the driver filter to *Notify for the nspmDistribution Password* attribute for all object classes that should subscribe to password changes.



Filter

- DirXML-nwoWorkOrder
- User
  - city
  - company
  - Description
  - Full Name
  - Given Name
  - jackNumber
  - L
  - preferredName
  - Surname
  - Telephone Number
  - Title
  - **nspmDistributionPassword**
- DirXML-pbxExtension

Class Name: User
Attribute Name: Attribute Information

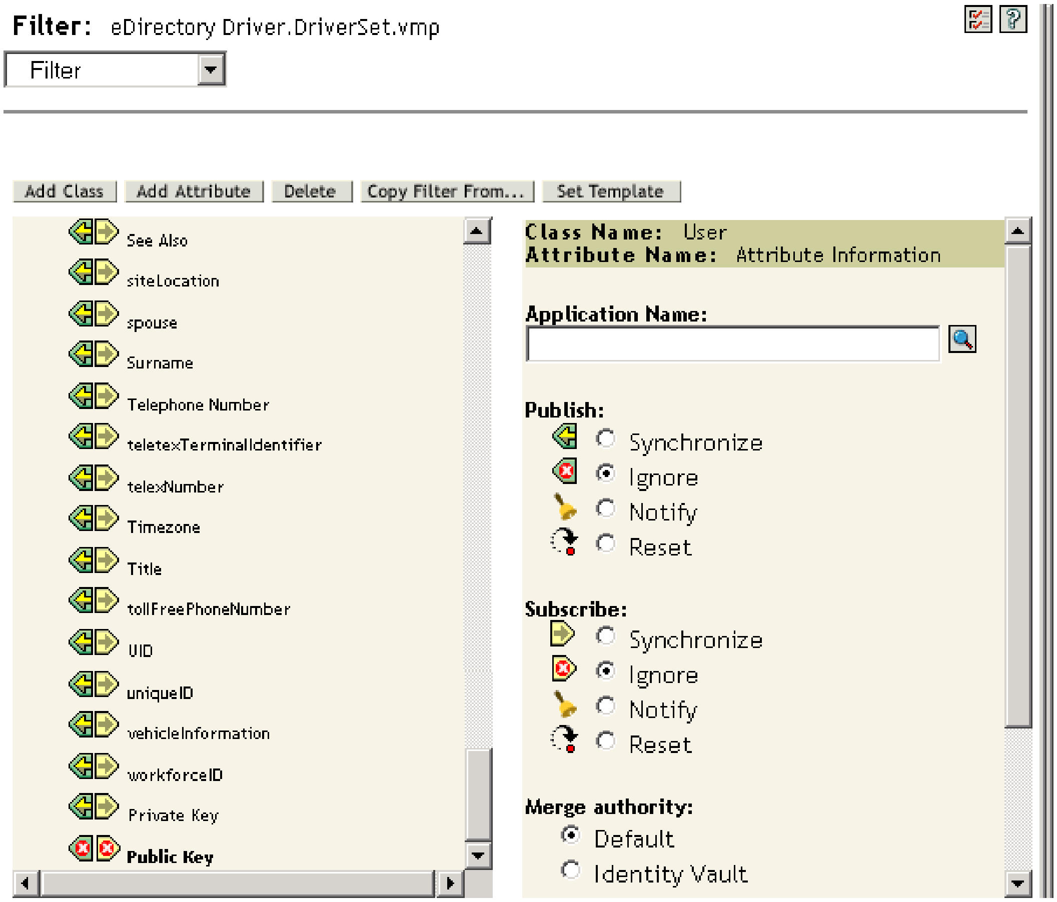
Application Name:

Publish:
○ Synchronize
● Ignore
○ Notify
○ Reset

Subscribe:
○ Synchronize
○ Ignore
● Notify
○ Reset

Merge authority:
● Default
○ Identity Vault

3 For all objects that have *Notify* set for the nspmDistributionPassword attribute, set both the Public Key and Private Key attributes to *Ignore*.



Filter: eDirectory Driver.DriverSet.vmp

See Also
siteLocation
spouse
Surname
Telephone Number
teletexTerminalIdentifier
telexNumber
Timezone
Title
tollFreePhoneNumber
UID
uniqueID
vehicleInformation
workforceID
Private Key
Public Key

Class Name: User
Attribute Name: Attribute Information

Application Name:

Publish:
○ Synchronize
● Ignore
○ Notify
○ Reset

Subscribe:
○ Synchronize
● Ignore
○ Notify
○ Reset

Merge authority:
● Default
○ Identity Vault

4 To ensure password security, make sure that you control who has rights to Identity Manager objects.

Troubleshooting Scenario 2

- "Flowchart for Scenario 2" on page 136
- "Trouble Logging in to the Identity Vault" on page 137
- "Trouble Logging in to Another Connected System that Subscribes to Passwords" on page 138
- "E-Mail Not Generated on Password Failure" on page 139
- "Error When Using Check the Object Password" on page 139
- "Helpful DSTrace Commands" on page 139

Also see the tips in Section 5.13, "Troubleshooting Password Synchronization," on page 173.

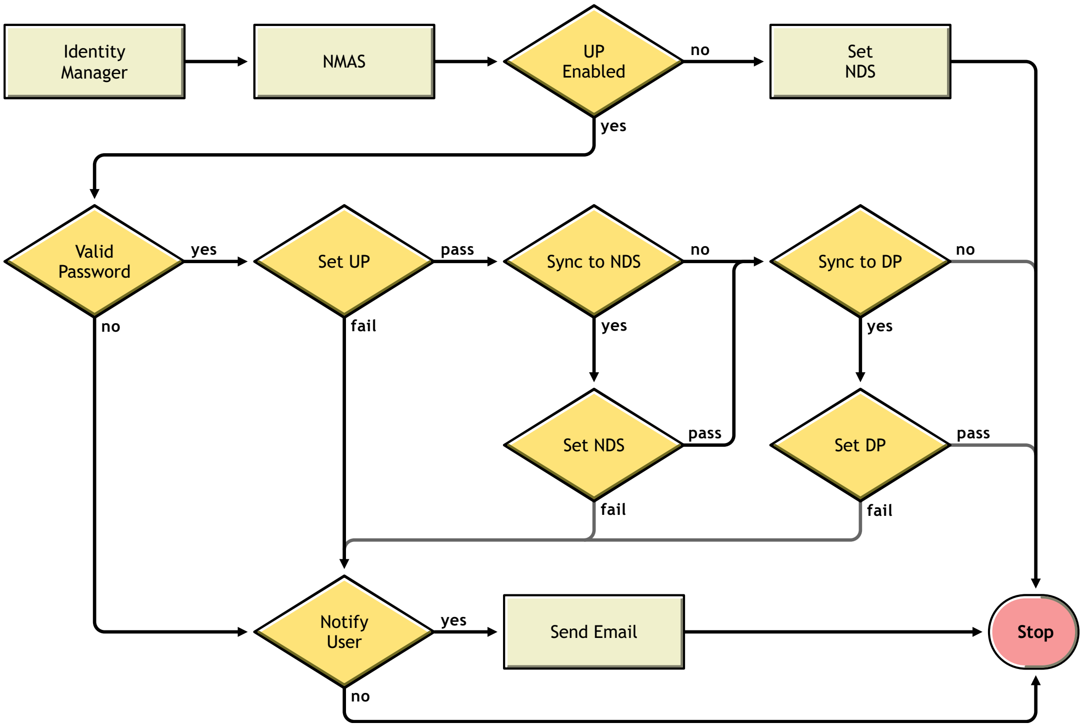
Flowchart for Scenario 2

Figure 5-9 illustrates how NMAS handles the password it receives from Identity Manager. The password is synchronized to Universal Password in this scenario. NMAS decides how to handle the password based on the following:

- Whether Universal Password is enabled in the NMAS password policy.

- Whether Advanced Password Rules are enabled that incoming passwords must comply with.
- What the other settings are in the password policy for synchronizing Universal Password with the other passwords.

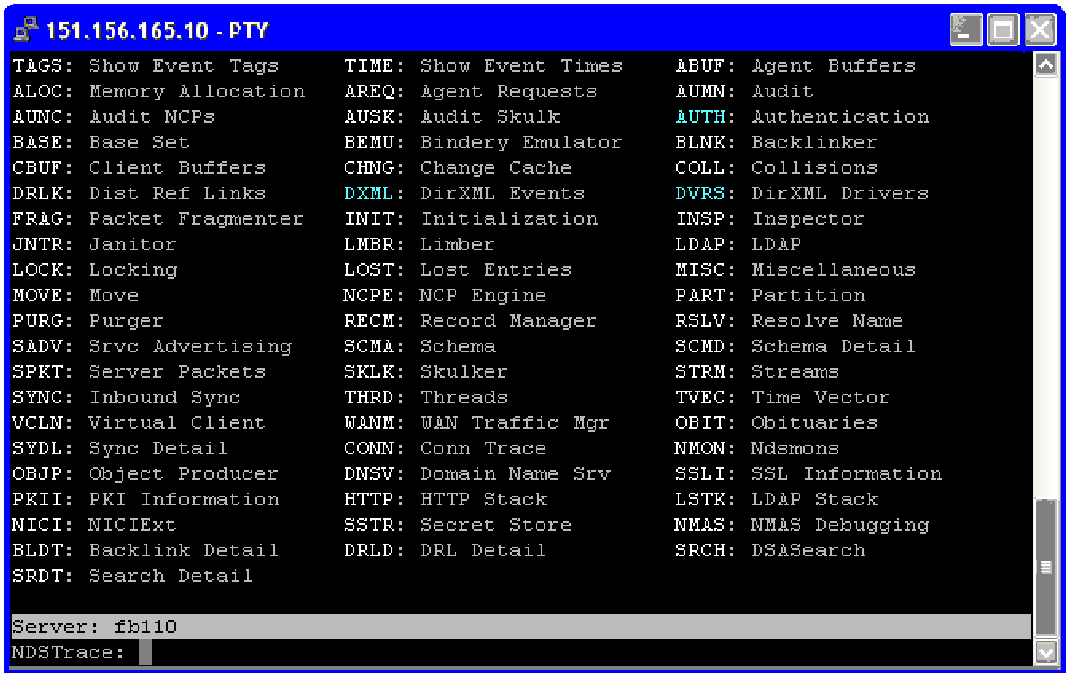
Figure 5-9 *How NMAS Handles the Password It Receives from Identity Manager*



Trouble Logging in to the Identity Vault

- Turn on the *+AUTH*, *+DXML*, and *+DVRS* settings in DSTrace.

Figure 5-10 *DSTrace Commands*



```
TAGS: Show Event Tags       TIME: Show Event Times     ABUF: Agent Buffers
ALOC: Memory Allocation     AREQ: Agent Requests       AUMN: Audit
AUNC: Audit NCPs            AUSK: Audit Skulk          AUTH: Authentication
BASE: Base Set              BEMU: Bindery Emulator     BLNK: Backlinker
CBUF: Client Buffers        CHNG: Change Cache         COLL: Collisions
DRLK: Dist Ref Links        DXML: DirXML Events        DVRS: DirXML Drivers
FRAG: Packet Fragmenter     INIT: Initialization       INSP: Inspector
JNTR: Janitor               LMBR: Limber               LDAP: LDAP
LOCK: Locking               LOST: Lost Entries         MISC: Miscellaneous
MOVE: Move                  NCPE: NCP Engine           PART: Partition
PURG: Purger                RECM: Record Manager       RSLV: Resolve Name
SADV: Srvc Advertising      SCMA: Schema               SCMD: Schema Detail
SPKT: Server Packets        SKLK: Skulker              STRM: Streams
SYNC: Inbound Sync          THRD: Threads              TVEC: Time Vector
VCLN: Virtual Client        WANM: WAN Traffic Mgr      OBIT: Obituaries
SYDL: Sync Detail           CONN: Conn Trace           NMON: Ndsmons
OBJP: Object Producer       DNSV: Domain Name Srv      SSLI: SSL Information
PKII: PKI Information       HTTP: HTTP Stack           LSTK: LDAP Stack
NICI: NICIExt               SSTR: Secret Store         NMAS: NMAS Debugging
BLDT: Backlink Detail       DRLD: DRL Detail           SRCH: DSASearch
SRDT: Search Detail

Server: fb110
NDSTrace:
```

- Verify that the `<password>` or `<modify-password>` elements are being passed to Identity Manager. To verify that they are being passed, watch the trace screen with those options turned on.
- Verify that the password is valid according to the rules of the password policy.
- Check the NMAS password policy configuration and assignment. Try assigning the policy directly to a user to make sure the correct policy is being used.
- On the Password Synchronization page for the driver, make sure that *DirXML accepts passwords* is selected.
- In the password policy, make sure that *Synchronize Distribution Password when setting Universal Password* is selected.

Trouble Logging in to Another Connected System that Subscribes to Passwords

This section is for troubleshooting cases where this connected system is publishing passwords to Identity Manager, but another connected system that is subscribing to passwords does not appear to be receiving the changes from this system. Another name for this relationship is a secondary connected system, meaning that it receives passwords from the first connected system through Identity Manager.

- Turn on the *+DXM*L and *+DVRS* settings in DSTrace to see Identity Manager rule processing
- Set the Identity Manager trace level for the driver to *3*.
- Make sure the Password Synchronization *Identity Manager Accepts Passwords* option is selected.
- Check the driver filter to make sure the nspmDistributionPassword attribute is set correctly, as explained in Step 2 on page 134.

- Verify that the <*password*> for an Add or <`modify-password`> element is being sent to the connected system. To verify, watch the DSTrace screen or file with the trace options turned on as noted in the first items.
- Verify that the driver configuration includes the Identity Manager script password policies in the correct location and correct order, as described in Section 5.3.4, "Policies Required in the Driver Configuration," on page 107.
- Compare the NMAS password policy in the Identity Vault with any password policies enforced by the connected system, to make sure they are compatible.

E-Mail Not Generated on Password Failure

- Turn on the *+DXML* setting in DSTrace to see Identity Manager rule processing.
- Set the Identity Manager trace level for the driver to *3*.
- Verify that the rule to generate e-mail is selected.
- Verify that the Identity Vault object contains the correct user e-mail address in the Internet EMail Address attribute.
- In the Notification Configuration task, make sure the SMTP server and the e-mail template are configured correctly. See Section 5.12, "Configuring E-Mail Notification," on page 161.

Error When Using Check the Object Password

The Check Password Status task in iManager causes the driver to check object password action. If you have problems, review the following:

- If the Check Object Password returns -603, the Identity Vault object does not contain an nspmDistributionPassword attribute. Check the driver filter for the correct settings for the nspmDistributionPassword attributes. Also, make sure that the password policy has *Synchronize Distribution Password when Setting Universal Password* selected.
- If the Check Object Password returns `Not Synchronized`, verify that the driver configuration contains the appropriate Password Synchronization policies.
- Compare the NMAS password policy in the Identity Vault with any password policies enforced by the connected system, to make sure they are compatible.
- Check Object Password operates from the Distribution Password. If the Distribution Password is not being updated, Check Object Password might not report that passwords are synchronized.
- Keep in mind that for the Identity Manager driver only, Check Password Status is checking the NDS Password instead of the Distribution Password.

Helpful DSTrace Commands

+DXML: To view Identity Manager rule processing and potential error message

+DVRS: To view Identity Manager driver messages

+AUTH: To view NDS password modifications

5.8.4 Scenario 3: Synchronizing an Identity Vault and Connected Systems, with Identity Manager Updating the Distribution Password

In this scenario, Identity Manager updates the Distribution Password directly, and allows NMAS to determine how the other Identity Vault passwords are synchronized.

Any connected system can publish passwords to Identity Manager, though not all connected systems can provide the user's actual password. For example, Active Directory can publish a user's actual password to Identity Manager. Although PeopleSoft does not provide a password from the PeopleSoft system itself, it can provide an initial password created in a policy in the driver configuration, such as a password based on the user's employee ID or last name. Not all drivers can subscribe to password changes from Identity Manager. See Section 5.2, "Connected System Support for Password Synchronization," on page 100.

- "Advantages and Disadvantages of Scenario 3" on page 140
- "Setting Up Scenario 3" on page 141
- "Troubleshooting Scenario 3" on page 145

Advantages and Disadvantages of Scenario 3

Table 5-13 *Advantages: Synchronizing an Identity Vault and Connected Systems by Updating the Distribution Password*

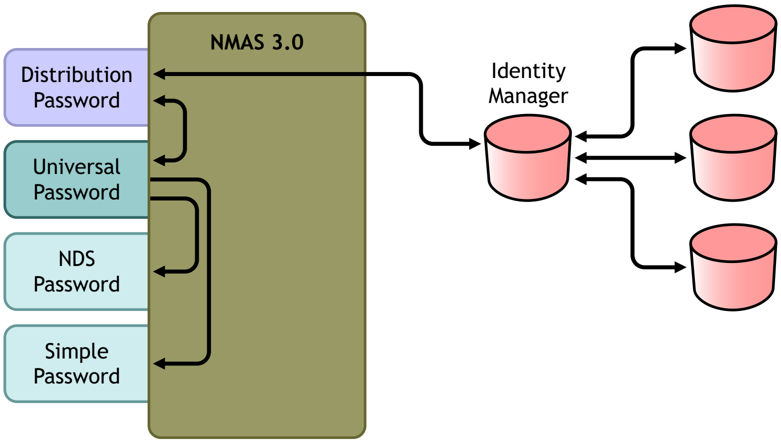
Advantages	Disadvantages
Allows synchronization of passwords between the Identity Vault and connected systems. Lets you choose whether or not to enforce password policies for passwords coming from connected systems. You can specify that notification be sent if password synchronization fails. If you are enforcing password policies, you can choose to reset a password on the connected system to the Distribution Password if the password doesn't comply.	

The figure in this scenario illustrates the following flow:

1. Passwords come in through Identity Manager.
2. Identity Manager goes through NMAS to directly update Distribution Password
3. Identity Manager also uses the Distribution Password to distribute to connected systems that you have specified should accept passwords
4. NMAS synchronizes Universal Password with the Distribution Password, and with other passwords according to the password policy settings.

Although multiple connected systems are shown as connecting to Identity Manager in Figure 5-11, keep in mind that you individually create the settings for each connected system driver.

Figure 5-11 Synchronizing an Identity Vault and Connected Systems by Updating the Distribution Password



Setting Up Scenario 3

To set up this kind of password synchronization:

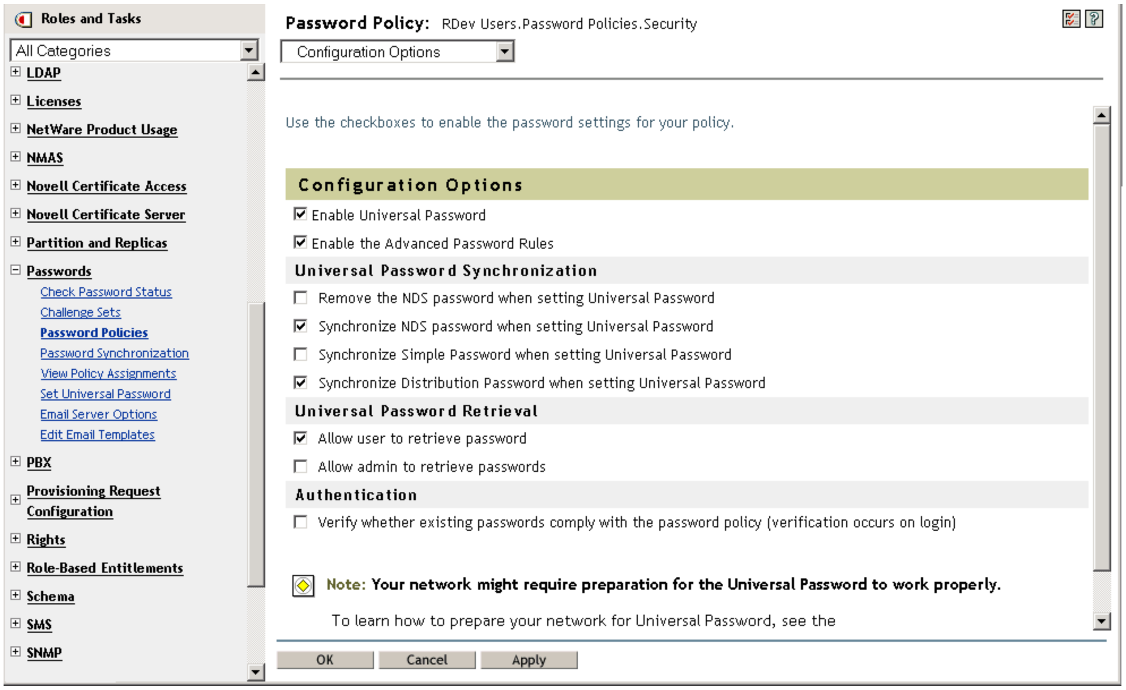
- "Universal Password Deployment" on page 141
- "Password Policy Configuration" on page 141
- "Password Synchronization Settings" on page 142
- "Driver Configuration" on page 144

Universal Password Deployment

Make sure that your environment is ready to use Universal Password. See Section 5.4, "Preparing to Use Identity Manager Password Synchronization and Universal Password," on page 111.

Password Policy Configuration

1. In iManager, select *Passwords* > *Password Policies*.
2. Make sure a password policy is assigned to the parts of the Identity Vault tree that you want to have this kind of password synchronization. You can assign it to the entire tree structure, a partition root container, a container, or a specific user. To simplify management, we recommend that you assign password policies as high in the tree as possible.
3. In the password policy, make sure the following are selected:



Password Policy: RDev Users.Password Policies.Security

Configuration Options

Use the checkboxes to enable the password settings for your policy.

Configuration Options

☑ Enable Universal Password

☑ Enable the Advanced Password Rules

Universal Password Synchronization

□ Remove the NDS password when setting Universal Password

☑ Synchronize NDS password when setting Universal Password

□ Synchronize Simple Password when setting Universal Password

☑ Synchronize Distribution Password when setting Universal Password

Universal Password Retrieval

☑ Allow user to retrieve password

□ Allow admin to retrieve passwords

Authentication

□ Verify whether existing passwords comply with the password policy (verification occurs on login)

Note: **Your network might require preparation for the Universal Password to work properly.**

To learn how to prepare your network for Universal Password, see the

- *Enable Universal Password*
- *Synchronize NDS Password when setting Universal Password*
- *Synchronize Distribution Password when setting Universal Password*

  Because Identity Manager retrieves the Distribution Password to distribute passwords to connected systems, it's important that this option be selected to allow bidirectional password synchronization.

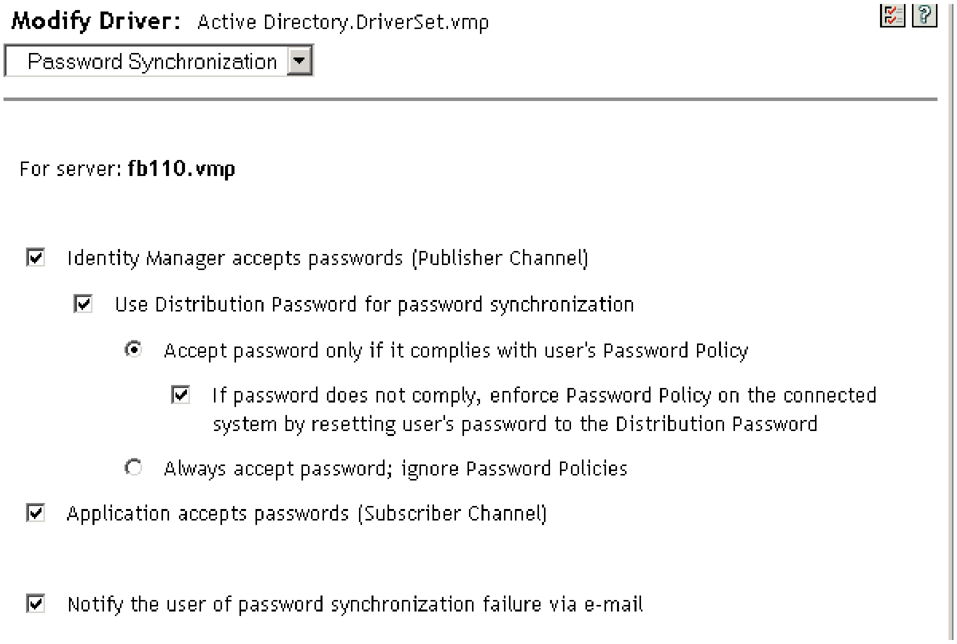
4 If you are using Advanced Password Rules, make sure that they don't conflict with the password policies on any connected systems that are subscribing to passwords.

Password Synchronization Settings

1 In iManager, select *Passwords* > *Password Synchronization*.

2 Search for drivers for the connected systems, then select a driver.

3 Create settings for the driver for the connected system.



Modify Driver: Active Directory.DriverSet.vmp

Password Synchronization ▾

For server: **fb110.vmp**

☑ Identity Manager accepts passwords (Publisher Channel)

☑ Use Distribution Password for password synchronization

● Accept password only if it complies with user's Password Policy

☑ If password does not comply, enforce Password Policy on the connected system by resetting user's password to the Distribution Password

○ Always accept password; ignore Password Policies

☑ Application accepts passwords (Subscriber Channel)

☑ Notify the user of password synchronization failure via e-mail

Make sure that the following are selected:

- *Identity Manager accepts passwords (Publisher Channel)*
- *Use Distribution Password for password synchronization*

  A message is displayed on the page if the driver manifest does not contain a "password-publish" capability. This is to inform users that passwords cannot be retrieved from the application and can only be published by creating a password in the driver configuration using a policy.
- *Application accepts passwords (Subscriber Channel)*

These settings allow for bidirectional password synchronization if it is supported by the connected system.

You can adjust the settings to match your business policies for the authoritative source for passwords. For example, if a connected system should subscribe to passwords but not publish, select only *Application accepts passwords (Subscriber Channel)*.

4 Specify whether you want NMAS password policies to be enforced or ignored, using the options under *Use Distribution Password for password synchronization*.

5 (Conditional) If you have specified that you want password policies to be enforced, also specify whether you want Identity Manager to reset the connected system password if it does not comply.

6 (Optional) Select the following if desired:

- *Notify the user of password synchronization failure via e-mail*

  Keep in mind that e-mail notifications require the Internet EMail Address attribute on the eDirectory user object to be populated.

  E-mail notifications are noninvasive. They do not affect the processing of the XML document that triggered the email. If they fail, they are not retried unless the operation itself is retried. However, debug messages for e-mail notifications are written to the trace file.

Driver Configuration

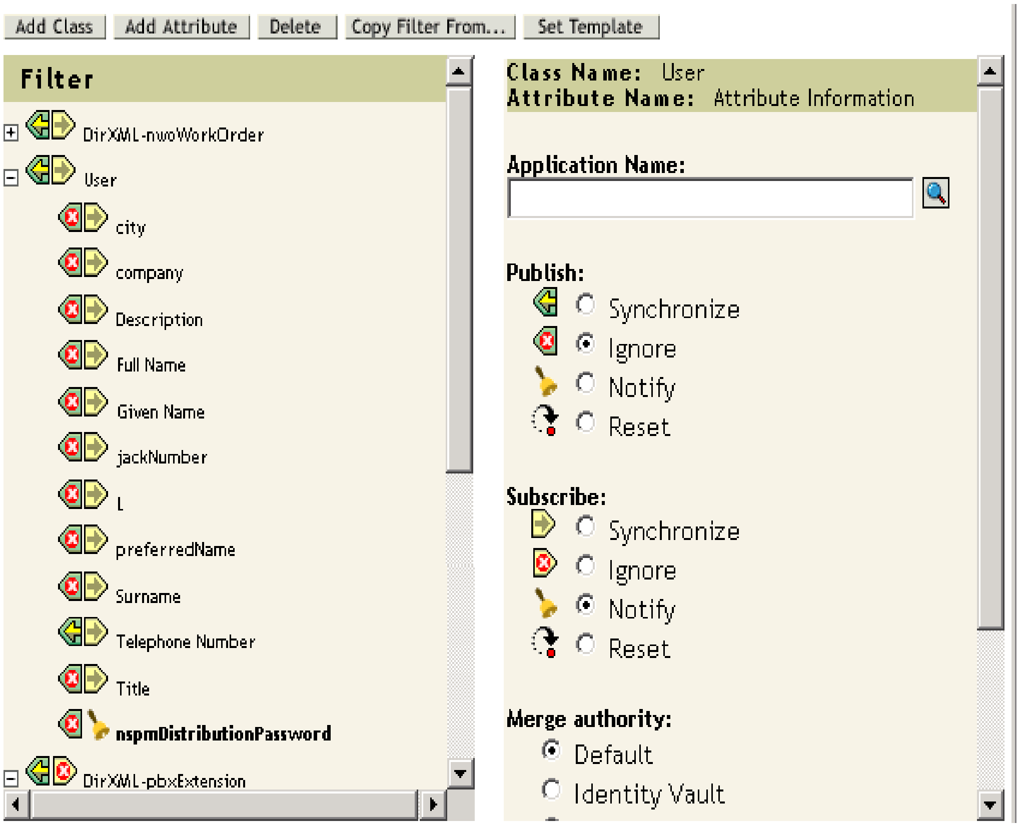
1 Make sure that the required Identity Manager script password synchronization policies are included in the driver configurations for each driver that should participate in password synchronization.

The policies must be in the correct location and the correct order in the driver configuration. For the list of policies, see Section 5.3.4, “Policies Required in the Driver Configuration,” on page 107.

The Identity Manager sample configurations already contain the policies. If you are upgrading an existing driver, you can add the policies using the instructions in Section 5.7, “Upgrading Existing Driver Configurations to Support Password Synchronization,” on page 116.

2 Set the filter correctly for nspmDistributionPassword attribute:

- For the Publisher channel, set the driver filter to *Ignore* for the nspmDistributionPassword attribute for all object classes.
- For the Subscriber channel, set the driver filter to *Notify* for the nspmDistribution Password attribute for all object classes that should subscribe to password changes.



Class Name: User
Attribute Name: Attribute Information

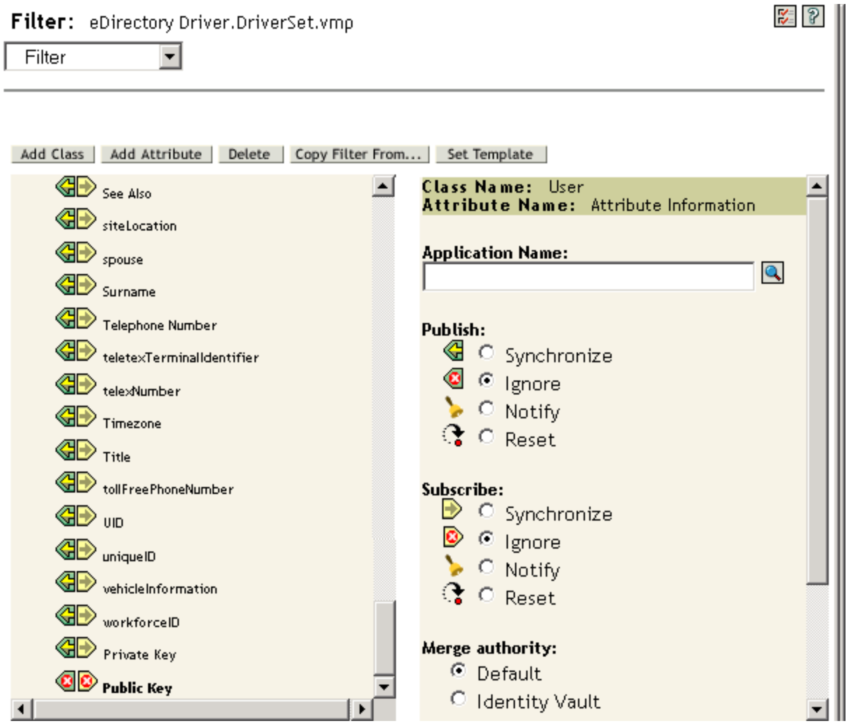
Application Name:

Publish:
○ Synchronize
● Ignore
○ Notify
○ Reset

Subscribe:
○ Synchronize
○ Ignore
● Notify
○ Reset

Merge authority:
● Default
○ Identity Vault

3 For all objects that have *Notify* set for the nspmDistributionPassword attribute, set both the Public Key and Private Key attributes in the driver filter to *Ignore*.



Filter: eDirectory Driver.DriverSet.vmp

Filter

See Also
siteLocation
spouse
Surname
Telephone Number
teletexTerminalIdentifier
telexNumber
Timezone
Title
tollFreePhoneNumber
UID
uniqueID
vehicleInformation
workforceID
Private Key
Public Key

Class Name: User
Attribute Name: Attribute Information

Application Name:

Publish:
○ Synchronize
● Ignore
○ Notify
○ Reset

Subscribe:
○ Synchronize
● Ignore
○ Notify
○ Reset

Merge authority:
● Default
○ Identity Vault

4 To ensure password security, make sure that you control who has rights to Identity Manager objects.

Troubleshooting Scenario 3

- "Flowchart for Scenario 3" on page 145
- "Trouble Logging in to eDirectory" on page 146
- "Trouble Logging in to Another Connected System that Subscribes to Passwords" on page 147
- "E-Mail Not Generated on Password Failure" on page 148
- "Error When Using Check Password Status" on page 148
- "Helpful DSTrace Commands" on page 148

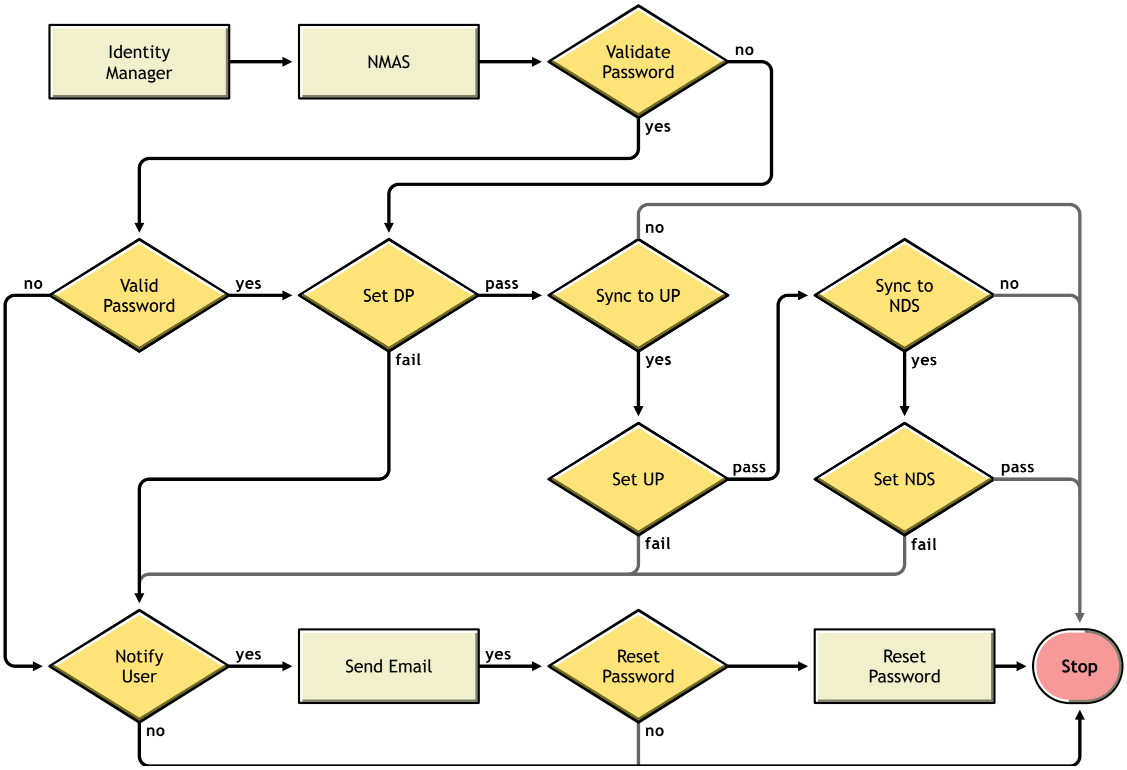
Also see the tips in Section 5.13, "Troubleshooting Password Synchronization," on page 173.

Flowchart for Scenario 3

Figure 5-12 illustrates how NMAS handles the password it receives from Identity Manager. The password is synchronized to the Distribution Password in this scenario, and NMAS decides the following:

- How to handle the password based on whether you have specified that incoming passwords should be validated against password policy rules (if Universal Password and Advanced Password Rules are enabled).
- What the other settings are in the password policy for synchronizing Universal Password with the other passwords.

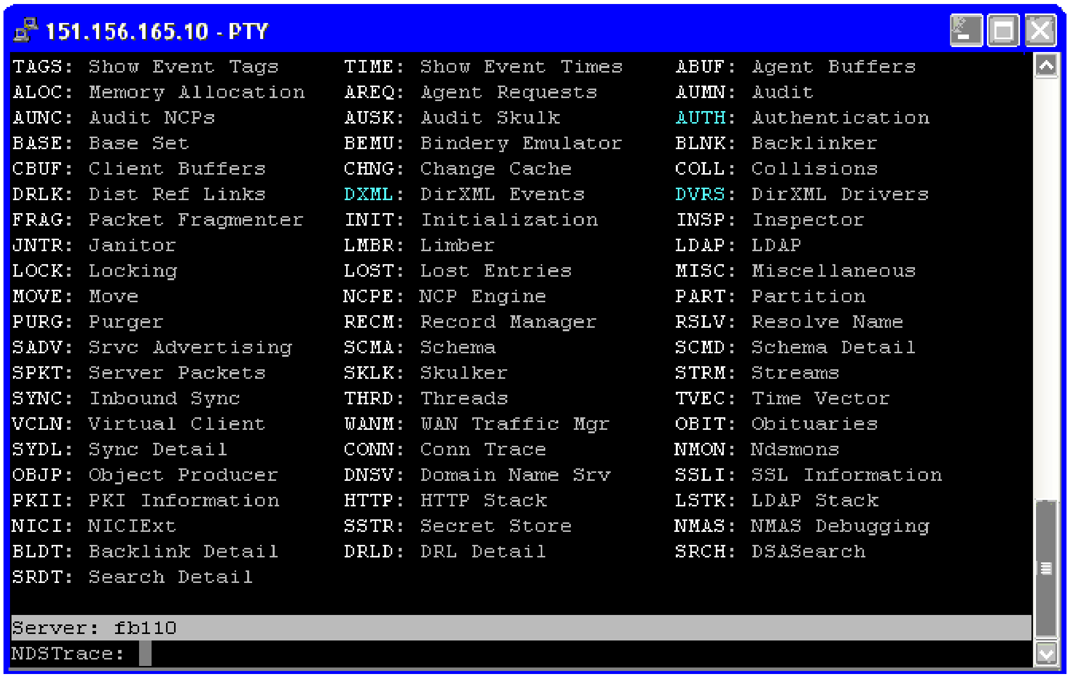
Figure 5-12 *Password from Identity Manager is Synchronized to the Distribution Password*



Trouble Logging in to eDirectory

- Turn on the *+AUTH*, *+DXML*, and *+DVRS* settings in DSTrace

Figure 5-13 *DSTrace commands*



```
TAGS: Show Event Tags       TIME: Show Event Times      ABUF: Agent Buffers
ALOC: Memory Allocation     AREQ: Agent Requests        AUMN: Audit
AUNC: Audit NCPs            AUSK: Audit Skulk           AUTH: Authentication
BASE: Base Set              BEMU: Bindery Emulator      BLNK: Backlinker
CBUF: Client Buffers        CHNG: Change Cache          COLL: Collisions
DRLK: Dist Ref Links        DXML: DirXML Events         DVRS: DirXML Drivers
FRAG: Packet Fragmenter     INIT: Initialization        INSP: Inspector
JNTR: Janitor               LMBR: Limber                LDAP: LDAP
LOCK: Locking               LOST: Lost Entries          MISC: Miscellaneous
MOVE: Move                  NCPE: NCP Engine            PART: Partition
PURG: Purger                RECM: Record Manager        RSLV: Resolve Name
SADV: Srvc Advertising      SCMA: Schema                SCMD: Schema Detail
SPKT: Server Packets        SKLK: Skulker               STRM: Streams
SYNC: Inbound Sync          THRD: Threads               TVEC: Time Vector
VCLN: Virtual Client        WANM: WAN Traffic Mgr       OBIT: Obituaries
SYDL: Sync Detail           CONN: Conn Trace            NMON: Ndsmons
OBJP: Object Producer       DNSV: Domain Name Srv       SSLI: SSL Information
PKII: PKI Information       HTTP: HTTP Stack            LSTK: LDAP Stack
NICI: NICIExt               SSTR: Secret Store          NMAS: NMAS Debugging
BLDT: Backlink Detail       DRLD: DRL Detail            SRCH: DSASearch
SRDT: Search Detail

Server: fb110
NDSTrace:
```

- Verify that the `<password>` or `<modify-password>` elements are being passed to Identity Manager. To verify, watch the DSTtrace screen or file with the trace options turned on as noted in the first item.
- Verify that the password is valid according to the rules of the NMAS password policy.
- Check the NMAS password policy configuration and assignment. Try assigning the policy directly to the user to make sure the correct policy is being used.
- On the Password Synchronization page for the driver, make sure that *Identity Manager accepts passwords (Publisher Channel)* is selected.
- In the NMAS password policy, make sure that *Synchronize Distribution Password when setting Universal Password* is selected.
- In the NMAS password policy, make sure that *Synchronize NDS Password when setting Universal Password* is selected, if this is desired.
- If users are logging in through the Novell Client or ConsoleOne, check the version. Legacy Novell Clients and ConsoleOne might not be able to log in to the Identity Vault if the Universal Password is not synchronized with the NDS Password.

  Versions of the Novell Client and ConsoleOne that are aware of the Universal Password are available. See the *NMAS 3.0 Administration Guide* (http://www.novell.com/documentation/nmas30/index.html).
- Some legacy utilities authenticate by using the NDS Password, and also cannot log in to the Identity Vault if the Universal Password is not synchronized with the NDS Password. If you don't want to use the NDS Password for most users, but you have administrator or help desk users who need to authenticate with legacy utilities, try using a different password policy for help desk users so you can specify different Universal Password synchronization options for them.

Trouble Logging in to Another Connected System that Subscribes to Passwords

This section is for troubleshooting situations where this connected system is publishing passwords to Identity Manager, but another connected system that is subscribing to passwords does not appear to be receiving the changes from this system. Another name for this relationship is a secondary connected system, meaning that it receives passwords from the first connected system through Identity Manager.

- Turn on the *+DXML* and *+DVRS* settings in DSTrace to see Identity Manager rule processing and potential errors
- Set the Identity Manager trace level for the driver to *3*.
- Make sure that the *Identity Manager accepts passwords (Publisher Channel)* option is selected in the Password Synchronization page.
- In the password policy, make sure that *Synchronize Distribution Password when setting Universal Password* is not selected.

  Identity Manager uses the Distribution Password to synchronize passwords to connected systems. Universal Password must be synchronized with the Distribution Password for this synchronization method.
- Check the driver filter for the nspmDistributionPassword attribute.
- Verify that the `<password>` element for an Add or a `<modify-password>` element has been converted to Add and Modify attribute operations for the nspmDistributionPassword. To verify, watch the DSTrace screen or file with the options turned on as noted in the first item.

- Verify that the driver configuration includes the Identity Manager script password policies in the correct location and correct order, as described in Section 5.3.4, "Policies Required in the Driver Configuration," on page 107.
- Compare the password policy in the Identity Vault with any password policies enforced by the connected system, to make sure they are compatible.

E-Mail Not Generated on Password Failure

- Turn on the *+DXML* setting in DSTrace to see Identity Manager rule processing
- Set the Identity Manager trace level for the driver to *3*.
- Verify that the rule to generate e-mail is selected.
- Verify that the Identity Vault object contains the correct value in the Internet EMail Address attribute.
- In the Notification Configuration task, make sure the SMTP server and the e-mail template are configured. See Section 5.12, "Configuring E-Mail Notification," on page 161.

E-mail notifications are non-invasive. They do not affect the processing of the XML document that triggered the e-mail. If they fail, they are not retried unless the operation itself is retried. Debug messages for e-mail notifications are written to the trace file.

Error When Using Check Password Status

The Check Password Status task in iManager causes the driver to perform a check object password action.

- Make sure the connected system supports checking passwords. See Section 5.2, "Connected System Support for Password Synchronization," on page 100.

  If the driver manifest does not indicate that the connected system supports password-check capability, this operation is not available through iManager.
- If the Check Object Password returns -603, the Identity Vault object does not contain an nspmDistributionPassword attribute. Check the driver filter, and the *Synchronize Universal to Distribution* option within the password policy.
- If the Check Object Password returns `Not Synchronized`, verify that the driver configuration contains the appropriate Identity Manager Password Synchronization policies.
- Compare the password policy in the Identity Vault with any password policies enforced by the connected system, to make sure they are compatible.
- *Check Object Password* checks the Distribution Password. If the Distribution Password is not being updated, *Check Object Password* might not report that passwords are synchronized
- Keep in mind that for the Identity Vault, *Check Password Status* checks the NDS Password instead of the Universal Password. This means that if the user's password policy does not specify to synchronize the NDS Password with the Universal Password, the passwords are always reported as being not synchronized. In fact, the Distribution Password and the password on the connected system might be in sync, but Check Password Status won't be accurate unless both the NDS Password and the Distribution Password are synchronized with the Universal Password.

Helpful DSTrace Commands

+DXML: To view Identity Manager rule processing and potential error message.

+*DVRS*: To view Identity Manager driver messages.

+*AUTH:* To view NDS password modifications.

5.8.5 Scenario 4: Tunneling

Identity Manager enables you to synchronize passwords among connected systems while keeping the Identity Vault password separate. This is referred to as “tunneling.”

In this scenario, Identity Manager updates the Distribution Password directly. This scenario is almost the same as Section 5.8.4, “Scenario 3: Synchronizing an Identity Vault and Connected Systems, with Identity Manager Updating the Distribution Password,” on page 140. The difference is that you make sure the Universal Password and the Distribution Password are not being synchronized. You do this either by not using NMAS password policies, or by using password policies with the option disabled for *Synchronize Distribution Password when setting Universal Password*.

- “Advantages and Disadvantages of Scenario 4” on page 149
- “Setting Up Scenario 4” on page 150
- “Troubleshooting Scenario 4” on page 152

Advantages and Disadvantages of Scenario 4

Table 5-14 *Advantages of Tunneling*

Advantages	Disadvantages
Allows synchronization of passwords among connected systems, while keeping the Identity Vault password separate. The password policy does not need to have Universal Password enabled, but the environment must support Universal Password. Supports the Check Password Status task in iManager, if the connected system supports it. You can specify that notification be sent if password synchronization fails. You can reset a connected system password that does not comply with password policy. If Universal Password and Advanced Password Rules are enabled, password policies are enforced if you specify that they should be enforced, and passwords on connected systems can be reset.	If Universal Password or Advanced Password Rules are not enabled, password policies are not enforced, and passwords on connected systems cannot be reset.

Figure 5-14 illustrates the following flow:

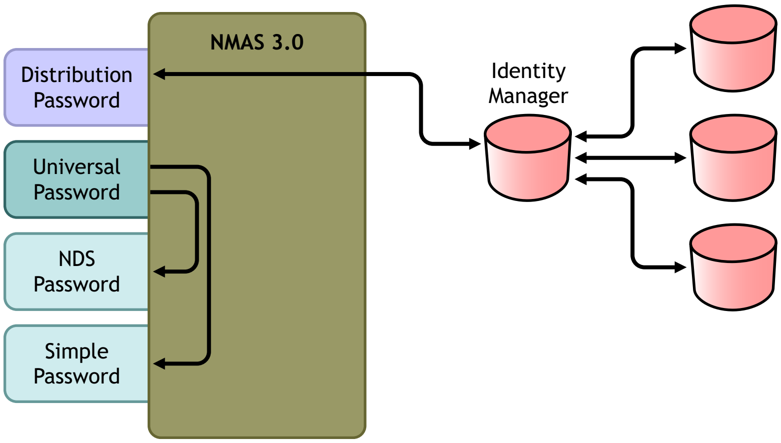
1. Passwords come in through Identity Manager.

2. Identity Manager goes through NMAS to directly update the Distribution Password.
3. Identity Manager also uses the Distribution Password to distribute passwords to connected systems that you have specified should accept passwords.

The key to this scenario is that in the NMAS password policy, *Synchronize Universal Password with Distribution Password* is disabled. Because the Distribution Password is not synchronized with the Universal Password, Identity Manager synchronizes passwords among connected systems without affecting passwords in the Identity Vault.

Although multiple connected systems are shown as connecting to Identity Manager in this figure, keep in mind that you individually create the settings for each connected system driver.

Figure 5-14 *Tunneling, with Identity Manager Updating the Distribution Password*



Setting Up Scenario 4

To set up this kind of password synchronization, configure the following:

- "Universal Password Deployment" on page 150
- "Password Policy Configuration" on page 151
- "Password Synchronization Settings" on page 151
- "Driver Configuration" on page 151

Universal Password Deployment

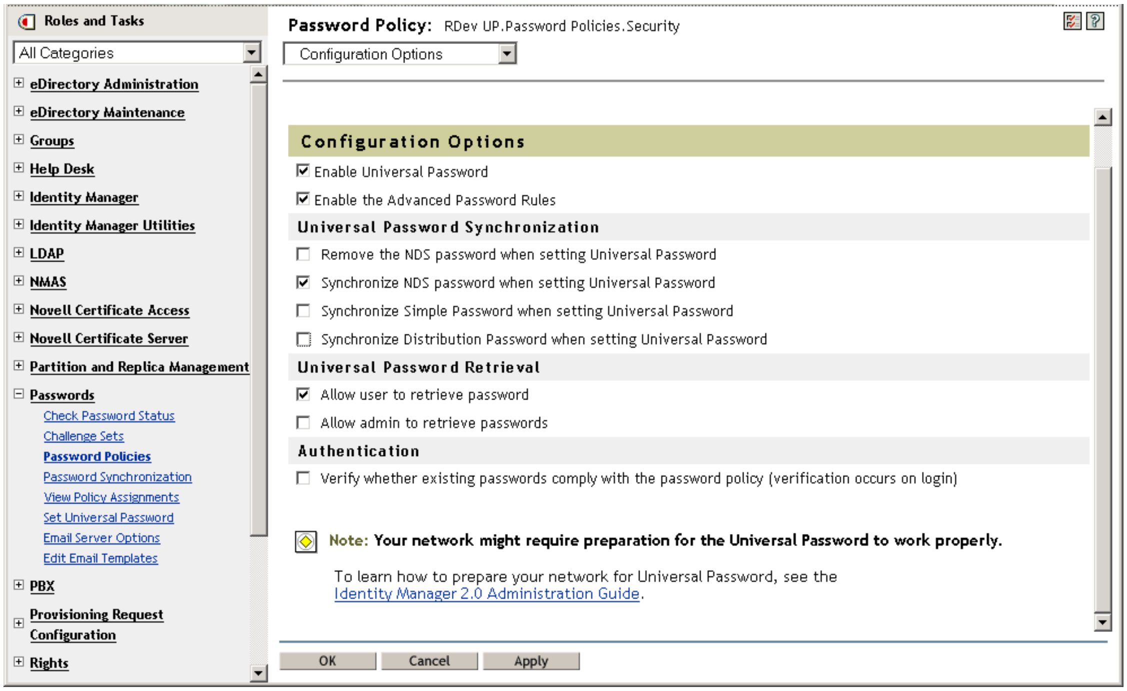
Although you don't need to have password policies with Universal Password enabled, your environment must still must be using eDirectory 8.7.3, which supports Universal Password. See Section 5.4, "Preparing to Use Identity Manager Password Synchronization and Universal Password," on page 111.

Password Policy Configuration

Review your password policy to confirm the following:

- Make sure *Synchronize Distribution Password when setting Universal Password* is not selected.

  This is the key to tunneling passwords without the Identity Vault password being affected. By not synchronizing the Universal Password with the Distribution Password, you keep the Distribution Password separate, for use only by Identity Manager for connected systems. Identity Manager acts as a conduit, distributing passwords to and from other connected systems, without affecting the Identity Vault password.



Password Policy: RDev UP.Password Policies.Security

Configuration Options

☑ Enable Universal Password

☑ Enable the Advanced Password Rules

Universal Password Synchronization

□ Remove the NDS password when setting Universal Password

☑ Synchronize NDS password when setting Universal Password

□ Synchronize Simple Password when setting Universal Password

□ Synchronize Distribution Password when setting Universal Password

Universal Password Retrieval

☑ Allow user to retrieve password

□ Allow admin to retrieve passwords

Authentication

□ Verify whether existing passwords comply with the password policy (verification occurs on login)

Note: Your network might require preparation for the Universal Password to work properly.

To learn how to prepare your network for Universal Password, see the Identity Manager 2.0 Administration Guide.

- Complete the other password policy settings as desired.

  The other password settings in the password policy are optional.

Password Synchronization Settings

Use the same settings as Password Synchronization Settings in Section 5.8.4, “Scenario 3: Synchronizing an Identity Vault and Connected Systems, with Identity Manager Updating the Distribution Password,” on page 140.

Driver Configuration

Use the same settings as Driver Configuration in Section 5.8.4, “Scenario 3: Synchronizing an Identity Vault and Connected Systems, with Identity Manager Updating the Distribution Password,” on page 140.

Troubleshooting Scenario 4

If password synchronization is set up for tunneling, the Distribution Password is different than the Universal Password and the NDS Password.

- "Trouble Logging in to Another Connected System that Subscribes to Passwords" on page 152
- "E-Mails Not Generated on Password Failure" on page 152
- "Error When Using Check Password Status" on page 153
- "Helpful DSTrace Commands" on page 153

See also the tips in Section 5.13, "Troubleshooting Password Synchronization," on page 173.

Trouble Logging in to Another Connected System that Subscribes to Passwords

This section is for troubleshooting situations where this connected system is publishing passwords to Identity Manager, but another connected system that is subscribing to passwords does not appear to be receiving the changes from this system. Another name for this relationship is a secondary connected system, meaning that it receives passwords from the first connected system through Identity Manager.

- Turn on the *+DXM*L and *+DVRS* settings in DSTrace to see Identity Manager rule processing and potential errors.
- Set the Identity Manager trace level for the driver to *3*.
- Make sure that the *Identity Manager accepts passwords (Publisher Channel)* option is selected on the Password Synchronization page.
- In the password policy, make sure that *Synchronize Distribution Password when setting Universal Password* is not selected.

  Identity Manager uses the Distribution Password to synchronize passwords to connected systems. The Universal Password must be synchronized with the Distribution Password for this synchronization method.
- Make sure the driver filter has the correct settings for the nspmDistributionPassword attribute.
- Verify that the <`password`> element for an Add and a <`modify-password`> element have been converted to Add and Modify attribute operations for the nspmDistributionPassword. To verify, watch the DSTrace screen or file with the trace options turned on as noted in the first item.
- Verify that the driver configuration includes the Identity Manager script password policies in the correct location and correct order, as described in Section 5.3.4, "Policies Required in the Driver Configuration," on page 107.
- Compare the password policy in the Identity Vault with any password policies enforced by the connected system, to make sure they are compatible.

E-Mails Not Generated on Password Failure

- Turn on the *+DXML* setting in DSTrace to see Identity Manager rule processing.
- Set the Identity Manager trace level for driver to *3*.
- Verify that the rule to generate e-mail is selected.

- Verify that the Identity Vault object contains the correct value in the Internet EMail Address attribute.
- In the Notification Configuration task, check the SMTP server and the e-mail template. See Section 5.12, “Configuring E-Mail Notification,” on page 161.

E-mail notifications are non-invasive. They do not affect the processing of the XML document that triggered the e-mail. If they fail, they are not retried unless the operation itself is retried. Debug messages for e-mail notifications are written to the trace file.

Error When Using Check Password Status

The Check Password Status task in iManager causes the driver to be perform a Check Object Password action.

- Make sure that the connected system supports checking passwords. See Section 5.2, “Connected System Support for Password Synchronization,” on page 100.

  This operation is not available through iManager if the driver manifest does not indicate that the connected system supports password-check capability.
- If the Check Object Password action returns -603, the Identity Vault object does not contain an nspmDistributionPassword attribute. Check the Identity Manager attribute filter, and the *Synchronize Universal to Distribution* option within the password policy.
- If the Check Object Password action returns `Not Synchronized`, verify that the driver configuration contains the appropriate Identity Manager password synchronization policies.
- Compare the password policy in the Identity Vault with any password policies enforced by the connected system, to make sure they are compatible.
- The Check Object Password action checks the Distribution Password. If the Distribution Password is not being updated, Check Object Password might not report that passwords are synchronized

Helpful DSTrace Commands

+*DXML*: To view Identity Manager rule processing and potential error messages.

+*DVRS*: To view Identity Manager driver messages.

+*AUTH*: To view NDS password modifications.

+*DCLN*: To view NDS DCLient messages.

5.8.6 Scenario 5: Synchronizing Application Passwords to the Simple Password

This scenario is a specialized use of password synchronization features. Using Identity Manager and NMAS, you can take a password from a connected system and synchronize it directly to the Identity Vault Simple Password. If the connected system provides only hashed passwords, you can synchronize them to the Simple Password without reversing the hash. Then, other applications can authenticate to the Identity Vault by using the same clear text or hashed password through LDAP or the Novell Client, with NMAS components configured to use the Simple Password as the login method.

If the password in the connected system is in clear text, it can be published as it is from the connected system into the Identity Vault Simple Password store.

If the connected system provides only hashed passwords (MD5, SHA, SHA1,or UNIX Crypt are supported), you must publish them to the Simple Password with an indication of the kind of hash, such as {MD5}.

For another application to authenticate with the same password, you need to customize the other application to take the user's password and authenticate to the Simple Password using LDAP.

NMAS compares the password value from the application with the value in the Simple Password. If the password stored in the Simple Password is a hash value, NMAS first uses the password value from the application to create the correct type of hash value, before comparing. If the password from the application and the Simple Password are the same, NMAS authenticates the user.

In this scenario, Universal Password cannot be used.

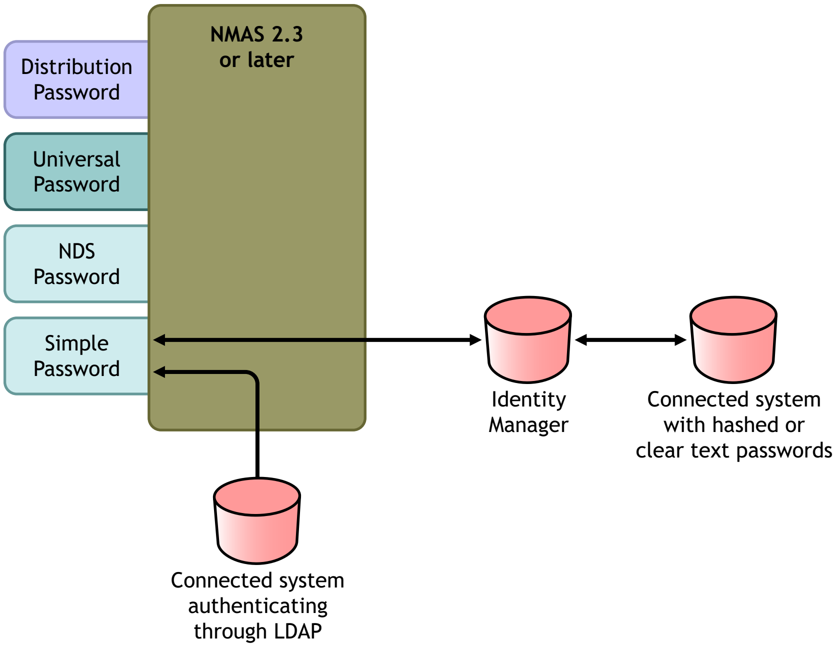
- "Advantages of Synchronizing to the NDS Password" on page 154
- "Setting Up Scenario 5" on page 155

Advantages of Synchronizing to the NDS Password

Table 5-15 *Advantages of Synchronizing to the NDS Password*

Advantages	Disadvantages
◆ Lets you update the Simple Password directly. ◆ Lets you synchronize a hashed password and use it to authenticate for more than one application, without reversing the hash.	◆ This scenario does not allow the use of Universal Password. ◆ Forgotten Password and Password Self-Service features can still be used to the extent they are supported for the NDS Password, but they do not work for the Simple Password. ◆ Because the Set Universal Password task is dependent on Universal Password, the administrator cannot set a user's password in the Identity Vault by using that task.

Figure 5-15 Synchronizing to the NDS Password



Setting Up Scenario 5

- "Password Policy Configuration" on page 155
- "Password Synchronization Settings" on page 155
- "Driver Configuration" on page 155

Password Policy Configuration

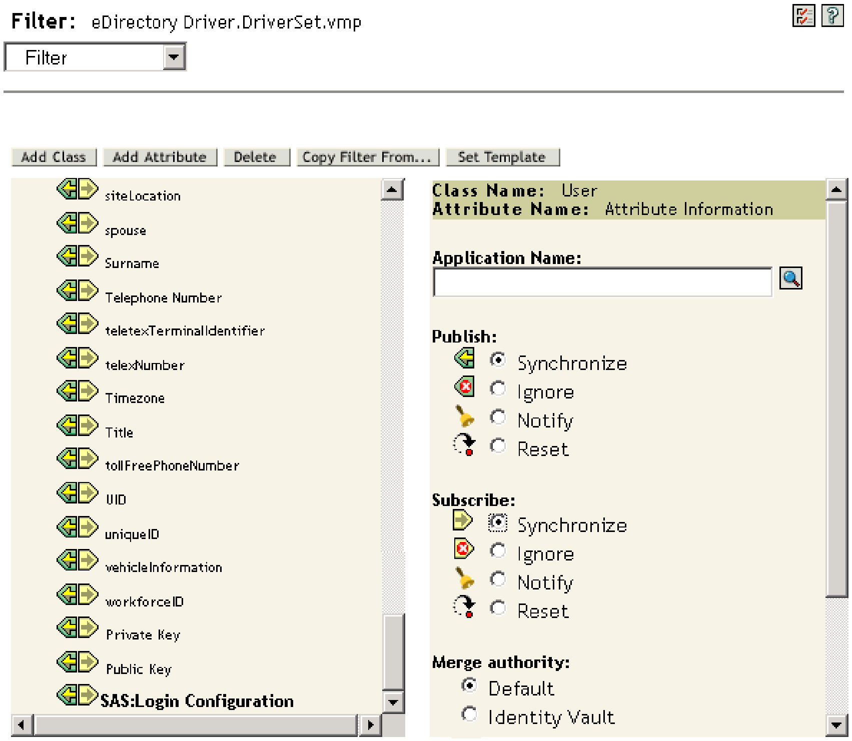
No password policy is required for users for this scenario. Universal Password cannot be used.

Password Synchronization Settings

For this scenario, you use Identity Manager Script to directly modify the SAS:Login Configuration attribute. This means that the Password Synchronization global configuration values (GCVs), which are set by using the Password Synchronization page in iManager, have no effect.

Driver Configuration

1 Make sure that the SAS:Login Configuration attribute in the filter has the setting of *Synchronize* for both Publisher and Subscriber channels.



Filter: eDirectory Driver.DriverSet.vmp

siteLocation
spouse
Surname
Telephone Number
teletexTerminalIdentifier
telexNumber
Timezone
Title
tollFreePhoneNumber
UID
uniqueID
vehicleInformation
workforceID
Private Key
Public Key
SAS:Login Configuration

Class Name: User
Attribute Name: Attribute Information

Application Name:

Publish:
● Synchronize
○ Ignore
○ Notify
○ Reset

Subscribe:
● Synchronize
○ Ignore
○ Notify
○ Reset

Merge authority:
● Default
○ Identity Vault

2 Configure the driver policies to publish the password from the connected system.

3 For hashed passwords, configure the driver policies to prepend the type of hash (if it is not already provided by the application):

- {MD5}*hashed_password*

  This password is Base64 encoded.

- {SHA}*hashed_password*

  This password is Base64 encoded.

- {CRYPT}*hashed_password*

Clear text passwords and Unix Crypt password hashes are not Base64 encoded.

4 To place the password into the Simple Password, configure the driver policies to modify the SAS:Login Configuration attribute.

The following example illustrates how to use a modify-attr element within a modify operation to change the Simple Password to an MD5 hashed password:

```
<modify-attr attr-name="SAS:Login Configuration>
    <add-value>
        <value>{MD5}2tEgXrIHtAnGHOzH3ENslg==</value>
    </add-value>
</modify-attr>
```

For clear text passwords, follow this example.

```
<modify-attr attr-name="SAS:Login Configuration>
    <add-value>
        <value>clearpwd</value>
    </add-value>
</modify-attr>
```

For add operations, the add-attr element would contain one of the following:

```
<add-attr attr-name="SAS:Login Configuration>
    <value>{MD5}2tEgXrIHtAnGHOzH3ENslg==</value>
</add-attr>
```

or

```
<add-attr attr-name="SAS:Login Configuration>
    <value>clearpwd</value>
</add-attr>
```

5.9 Setting Up Password Filters

Some connected systems can provide the user's actual password to Identity Manager.

To capture passwords on Active Directory, NIS, and NT Domain, you must do some minor setup to install password filters on connected systems.

- Section 5.9.1, “Setting Up Password Synchronization Filters for Active Directory and NT Domain,” on page 157
- Section 5.9.2, “Setting Up Password Synchronization Filters for NIS,” on page 157

5.9.1 Setting Up Password Synchronization Filters for Active Directory and NT Domain

This information is in the Password Synchronization sections in the driver implementation guides for the Identity Manager Drivers for Active Directory and NT Domain, at Identity Manager Drivers (http://www.novell.com/documentation/dirxmldrivers/index.html).

The Identity Manager driver for AD or NT Domain needs to be installed on only one Windows machine. The other domain controllers don't need the driver installed, but each domain controller does need a `pwfilter.dll` file installed to capture passwords so they can be sent to Identity Manager.

To simplify your setup and administration, a utility is provided that lets you do this for all domain controllers from the Windows machine where the driver is installed.

5.9.2 Setting Up Password Synchronization Filters for NIS

The Identity Manager Driver for NIS 3.0 can operate with three UNIX authentication data stores: files, NIS and NIS+. A PAM module is provided to capture passwords and send them to the Identity Manager Driver for NIS.

The deployment of the PAM module for the NIS Driver is explained in the *Identity Manager Driver for NIS Implementation Guide*, at Identity Manager Drivers (http://www.novell.com/documentation/lg/dirxmldrivers/index.html).

5.10 Managing Password Synchronization

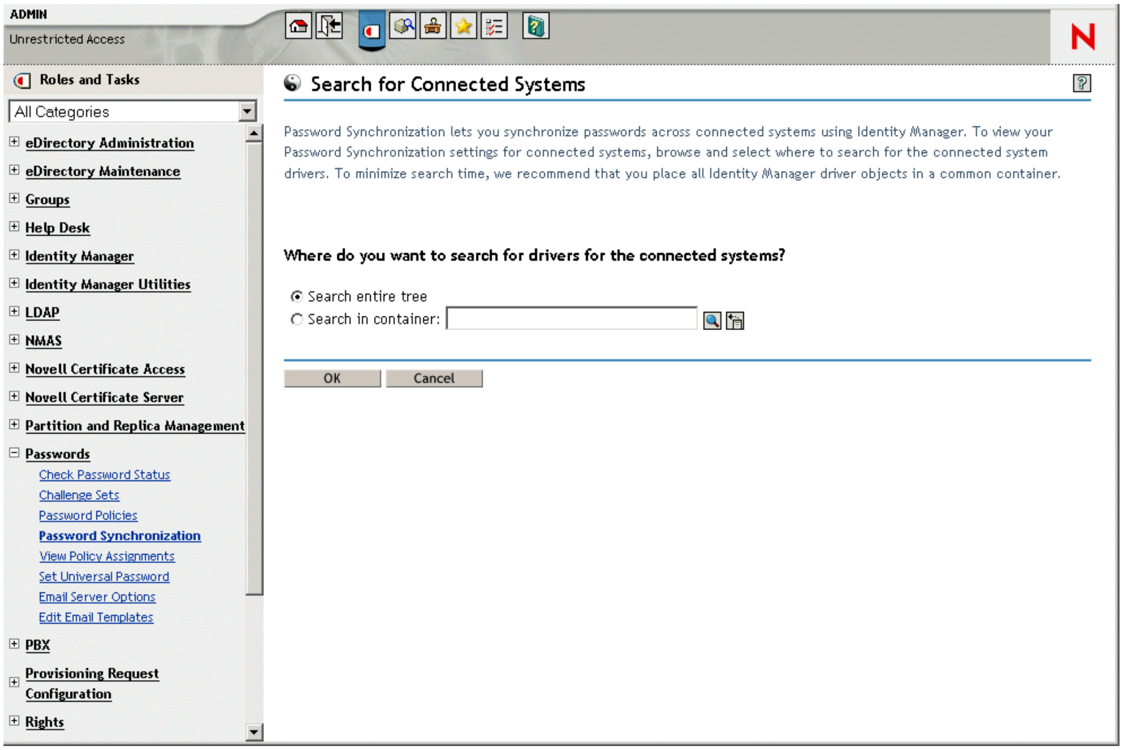
- “Setting the Flow of Passwords Across Systems” on page 158
- “Enforcing Password Policies on Connected Systems” on page 159
- “Keeping the eDirectory Password Separate from the Synchronized Password” on page 159

5.10.1 Setting the Flow of Passwords Across Systems

To view how your systems are set up to accept or publish passwords:

1 In iManager, select *Passwords* > *Password Synchronization*.

2 Search for drivers for the connected systems.



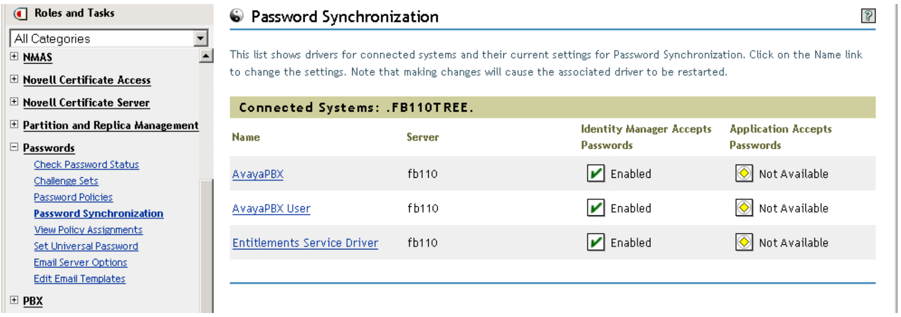
Search for Connected Systems

Password Synchronization lets you synchronize passwords across connected systems using Identity Manager. To view your Password Synchronization settings for connected systems, browse and select where to search for the connected system drivers. To minimize search time, we recommend that you place all Identity Manager driver objects in a common container.

Where do you want to search for drivers for the connected systems?

● Search entire tree
○ Search in container:

The search results show the settings for password flow to and from Identity Manager and the connected systems.



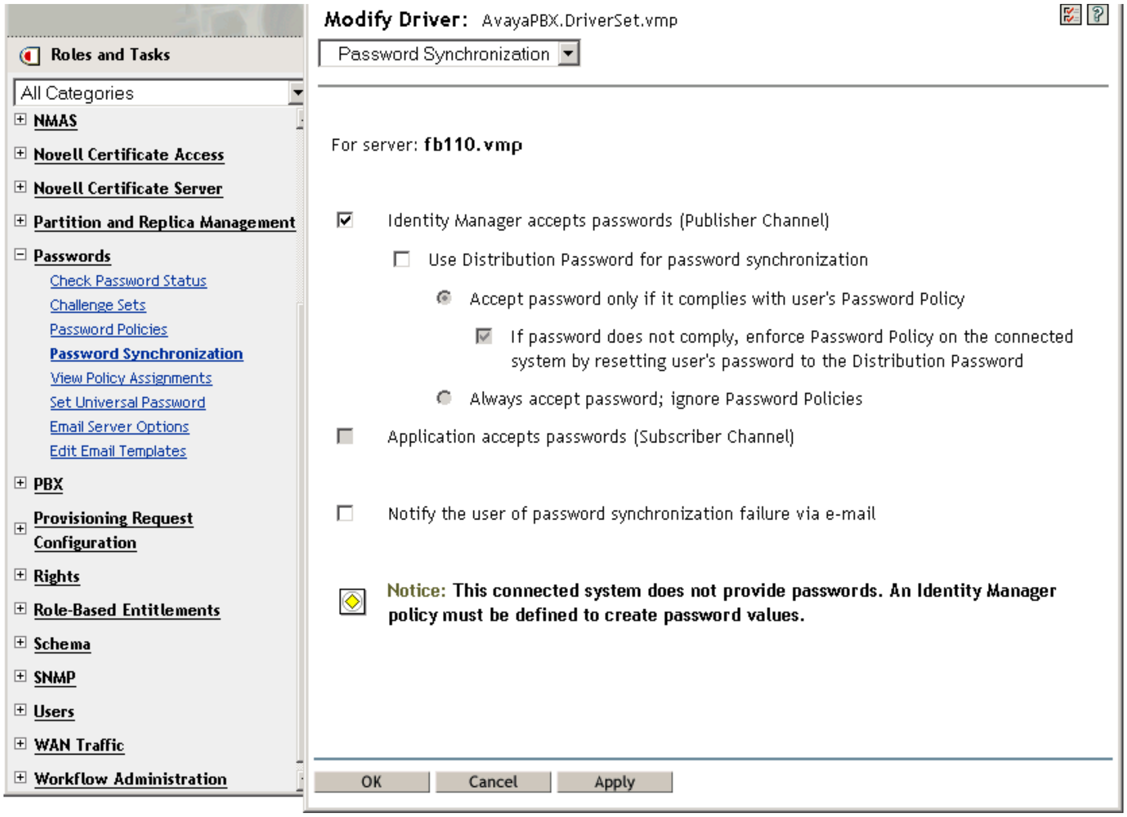
Password Synchronization

This list shows drivers for connected systems and their current settings for Password Synchronization. Click on the Name link to change the settings. Note that making changes will cause the associated driver to be restarted.

Connected Systems: .FB110TREE.

Name	Server	Identity Manager Accepts Passwords	Application Accepts Passwords
AvayaPBX	fb110	☑ Enabled	◇ Not Available
AvayaPBX User	fb110	☑ Enabled	◇ Not Available
Entitlements Service Driver	fb110	☑ Enabled	◇ Not Available

To make changes to these settings, click a connected system driver name.



Modify Driver: AvayaPBX.DriverSet.vmp

Password Synchronization

For server: **fb110.vmp**

☑ Identity Manager accepts passwords (Publisher Channel)

☐ Use Distribution Password for password synchronization

● Accept password only if it complies with user's Password Policy

☑ If password does not comply, enforce Password Policy on the connected system by resetting user's password to the Distribution Password

○ Always accept password; ignore Password Policies

☐ Application accepts passwords (Subscriber Channel)

☐ Notify the user of password synchronization failure via e-mail

Notice: **This connected system does not provide passwords. An Identity Manager policy must be defined to create password values.**

On the Modify Driver page, you can set whether password policy is enforced for passwords coming in to Identity Manager, and whether a password policy is enforced on the connected system by resetting the connected system password.

The settings on this page are global configuration values (GCVs), which are stored per server. See Section 5.3.3, “Using Global Configuration Values to Control Password Synchronization,” on page 104.

5.10.2 Enforcing Password Policies on Connected Systems

If you are using Advanced Password Rules and are using Identity Manager Password Synchronization, we recommend that you do the following:

1. Research the password policies for all the connected systems.
2. Make sure that the Advanced Password Rules are compatible with password policies on the connected systems.

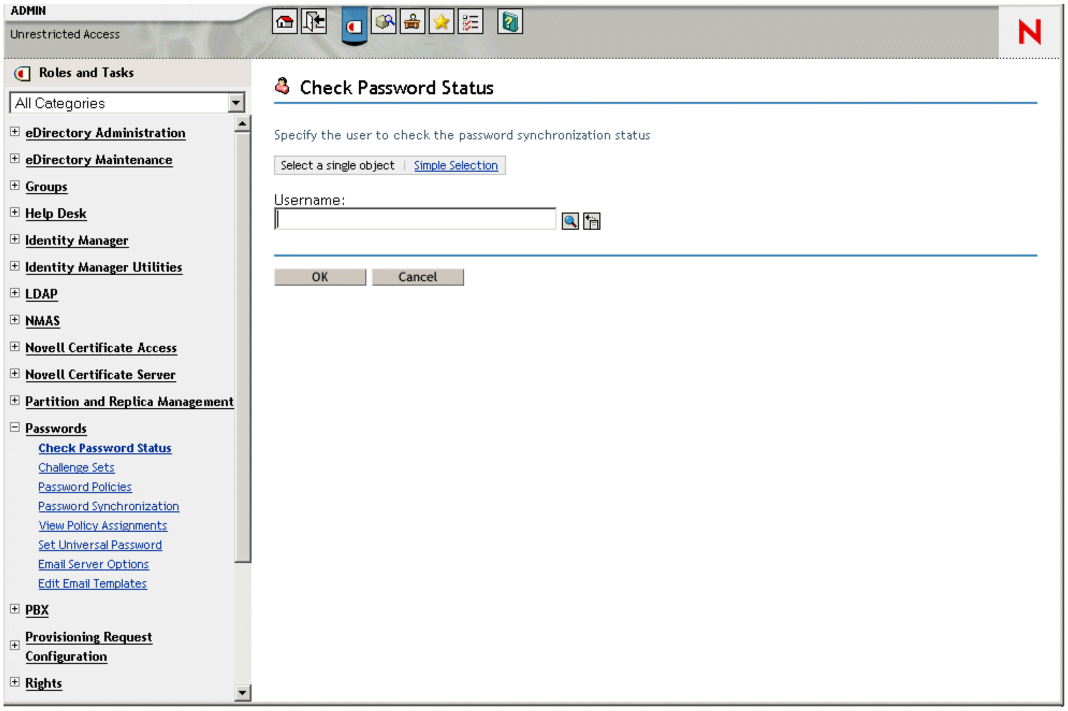
5.10.3 Keeping the eDirectory Password Separate from the Synchronized Password

This scenario is described in Section 5.8.5, “Scenario 4: Tunneling,” on page 149.

5.11 Checking the Password Synchronization Status for a User

You can determine whether the Distribution Password for a specific user is the same as the password in the connected system.

1 In iManager, select *Passwords* > *Check Password Status*.



Specify the user to check the password synchronization status

2 Browse to and select a user.

The *Check Password Status* task causes the driver to perform a Check Object Password action.

Not all drivers support password check. Those that do must contain a password-check capability in the driver's manifest. iManager does not allow password check operations to be sent to drivers that do not contain this capability in the manifest.

The Check Object Password action checks the Distribution Password. If the Distribution Password is not being updated, Check Object Password might report that passwords are not synchronized.

The Distribution Password is not updated if either of the following occurs:

- You are using the synchronization method described in Section 5.8.2, "Scenario 1: Using NDS Password to Synchronize between Two Identity Vaults," on page 126.
- You are synchronizing Universal Password (as in Section 5.8.3, "Scenario 2: Using Universal Password to Synchronize Passwords," on page 129), but you have not enabled the password policy configuration option to synchronize Universal Password to Distribution Password.

NOTE: Keep in mind that for the Identity Vault, the Check Password Status action checks the NDS Password instead of the Universal Password. Therefore, if the user's password policy does not specify to synchronize the NDS Password with the Universal Password, the passwords are always reported as being not synchronized. In fact, the Distribution Password and the password on the connected system might be in sync, but Check Password Status won't be accurate unless both the NDS Password and the Distribution Password are synchronized with the Universal Password.

5.12 Configuring E-Mail Notification

iManager tasks enable you to specify the e-mail server and customize the templates for e-mail notifications.

E-mail templates are provided to allow Password Synchronization and Password Self-Service to send automated e-mails to users.

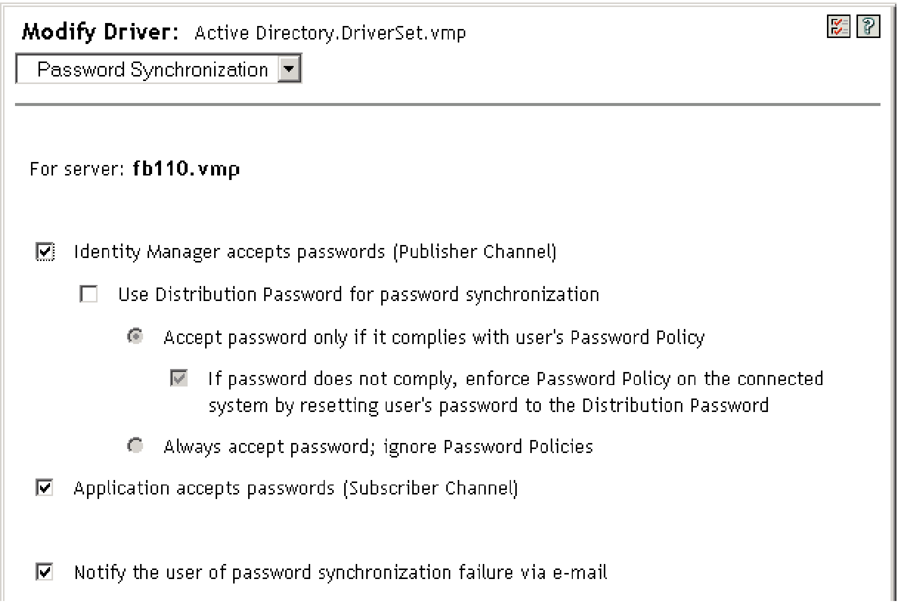
You don't create the templates. They are provided by the application that uses them. The e-mail templates are Template objects in the Identity Vault, and they are placed in the Security container, usually found at the root of your tree. Although they are Identity Vault objects, you should edit them only through iManager.

This is a modular framework. As new applications are added that use e-mail templates, the templates can be installed along with the applications that use them.

You control whether e-mail messages are sent, based on your choices in iManager. For Forgotten Password, e-mail notifications are sent only if you choose to use one of the Forgotten Password actions that causes an e-mail to be sent: e-mailing a password to the user, or e-mailing a password hint to the user. See “Providing Users with Forgotten Password Self-Service” in the *Password Management Administration Guide* (http://www.novell.com/documentation/password_management/index.html).

When you select *Notify the user of password synchronization failure via e-mail*, Password Synchronization is configured to send e-mail for failed password sync operations only, and only for the drivers you specify.

Figure 5-16 *Configuring Password Synchronization*



Modify Driver: Active Directory.DriverSet.vmp

Password Synchronization ▾

For server: **fb110.vmp**

☑ Identity Manager accepts passwords (Publisher Channel)

☐ Use Distribution Password for password synchronization

● Accept password only if it complies with user's Password Policy

☑ If password does not comply, enforce Password Policy on the connected system by resetting user's password to the Distribution Password

○ Always accept password; ignore Password Policies

☑ Application accepts passwords (Subscriber Channel)

☑ Notify the user of password synchronization failure via e-mail

In addition, you need to make sure that the SMTP authentication information is included in the driver policies.

- Section 5.12.1, “Prerequisites,” on page 162
- Section 5.12.2, “Setting Up the SMTP Server To Send E-Mail Notification,” on page 162
- “Setting Up E-Mail Templates for Notification” on page 164
- Section 5.12.4, “Providing SMTP Authentication Information in Driver Policies,” on page 164
- Section 5.12.5, “Adding Your Own Replacement Tags to E-Mail Notification Templates,” on page 166
- Section 5.12.6, “Sending E-Mail Notifications to the Administrator,” on page 172
- Section 5.12.7, “Localizing E-Mail Notification Templates,” on page 172

5.12.1 Prerequisites

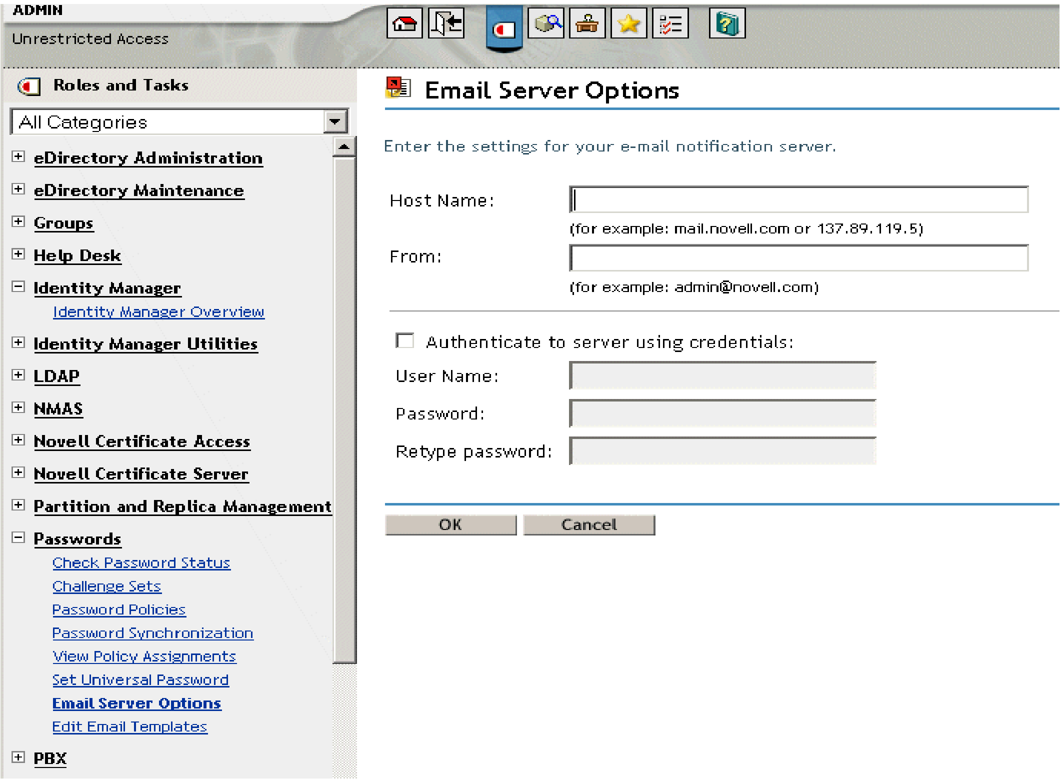
- ❑ Make sure that your Identity Vault users have the Internet EMail Address attribute populated.
- ❑ If you are using e-mail notifications for Password Synchronization, make sure that the Password Synchronization driver policies contain the password for the SMTP server. See Section 5.12.4, “Providing SMTP Authentication Information in Driver Policies,” on page 164.
- ❑ If you are concerned that some users might not have the e-mail address populated, or if you want an e-mail record of all failure notifications, consider choosing a password administrator account that all e-mail notifications are sent to, in addition to the user.

  This e-mail address should be in the *To* field of the Identity Manager script policy. For more information, see Section 5.12.6, “Sending E-Mail Notifications to the Administrator,” on page 172.
- ❑ If eDirectory and Identity Manager are on a UNIX server, the server must hold a replica of the e-mail template objects.

  These objects are located in the Security container, at the root. This means that the server needs a replica of the root partition.

5.12.2 Setting Up the SMTP Server To Send E-Mail Notification

1. In iManager, select *Passwords* > *Email Server Options*.



Email Server Options

Enter the settings for your e-mail notification server.

Host Name:

(for example: mail.novell.com or 137.89.119.5)

From:

(for example: admin@novell.com)

☐ Authenticate to server using credentials:

User Name:

Password:

Retype password:

2 Type the following information:

- The host name.
- The name (for example, Administrator) that you want to appear in the *From* field of the e-mail message.
- The username and password for authenticating to the server, if necessary.

3 Click *OK*.

4 If you are using Password Synchronization with your Identity Manager drivers and want to use the e-mail notification feature, you must also do the following:

4a If your SMTP server requires authentication before sending e-mail, make sure that the driver policies contain the password. See Section 5.12.4, “Providing SMTP Authentication Information in Driver Policies,” on page 164 for instructions.

Specifying the authentication information in the Email Server Options page in Step 2 is sufficient for Forgotten Password notifications, but not for Password Synchronization notifications.

4b Restart Identity Manager drivers that need to be updated with the changes.

The driver reads the templates and SMTP server information only at startup time.

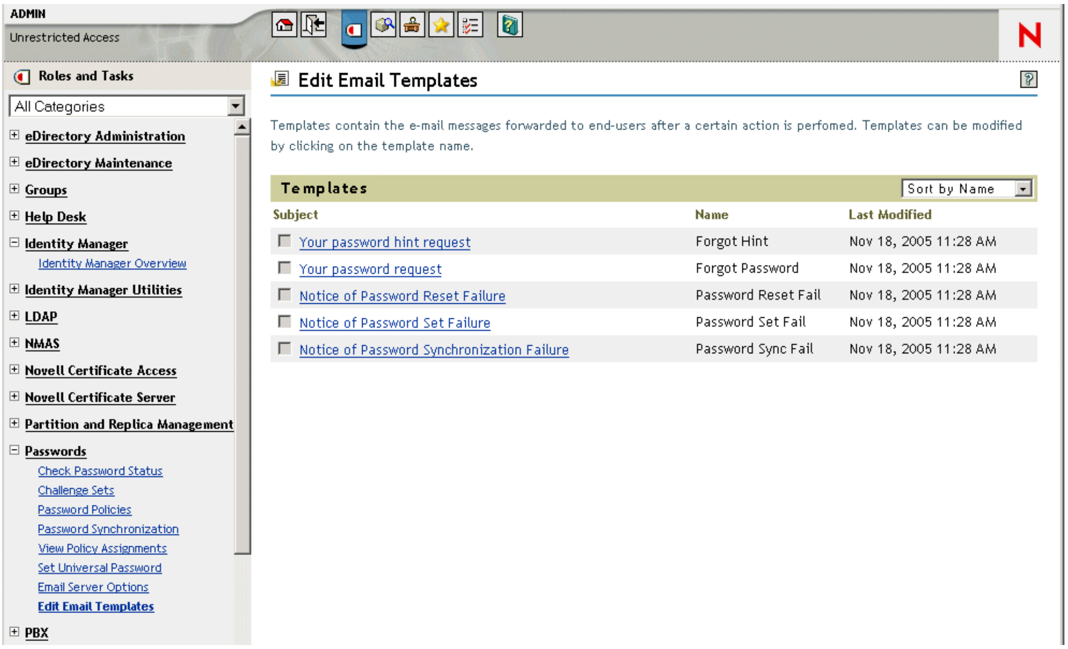
5 Customize the e-mail templates as described in “Setting Up E-Mail Templates for Notification” on page 164.

After the e-mail server is set up, e-mail messages can be sent by the applications that use them, if you are using the features that cause messages to be sent.

5.12.3 Setting Up E-Mail Templates for Notification

You can customize these templates with your own text. The name of the template indicates what it is used for.

1 In iManager, select *Passwords > Edit Email Templates*.



Edit Email Templates

Templates contain the e-mail messages forwarded to end-users after a certain action is perfomed. Templates can be modified by clicking on the template name.

Templates — Sort by Name

Subject	Name	Last Modified
□ Your password hint request	Forgot Hint	Nov 18, 2005 11:28 AM
□ Your password request	Forgot Password	Nov 18, 2005 11:28 AM
□ Notice of Password Reset Failure	Password Reset Fail	Nov 18, 2005 11:28 AM
□ Notice of Password Set Failure	Password Set Fail	Nov 18, 2005 11:28 AM
□ Notice of Password Synchronization Failure	Password Sync Fail	Nov 18, 2005 11:28 AM

2 Edit the templates as desired.

Keep in mind that if you want to add any replacement tags, some additional tasks might be required. Follow the instructions in Section 5.12.5, "Adding Your Own Replacement Tags to E-Mail Notification Templates," on page 166.

3 Restart Identity Manager drivers that need to be updated with the changes.

The driver reads the templates and SMTP server information only at startup time.

5.12.4 Providing SMTP Authentication Information in Driver Policies

You specify the username and password for the SMTP server in Section 5.12.2, "Setting Up the SMTP Server To Send E-Mail Notification," on page 162. For Forgotten Password e-mail notifications, this is sufficient.

However, for Password Synchronization e-mail notifications, you also need to include the password in the driver policies. The Metadirectory engine can access the username, but not the passwords. The driver policy must provide it.

You must complete this procedure if the following conditions exist:

- The SMTP server is secured and requires authentication before sending e-mail.

- You are using Identity Manager Password Synchronization with an Identity Manager driver
- In the Password Synchronization settings for the driver, you have selected *Notify the user of password synchronization failure via e-mail*.

To add the SMTP server password to the driver policy:

1 Make sure the driver has the policies that are necessary for using Password Synchronization.

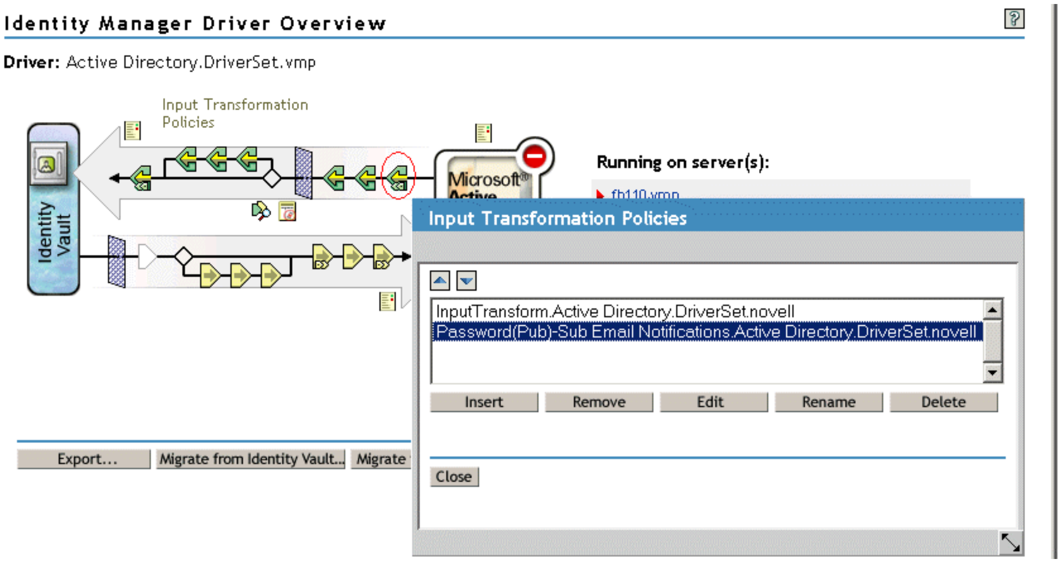
  These policies are provided in the sample driver configurations, or can be added as described in Section 5.7, "Upgrading Existing Driver Configurations to Support Password Synchronization," on page 116.

2 In iManager, select *Identity Manager > Identity Manager Overview.*

3 Search for the driver sets, or browse and select a container that holds the driver set.

4 In the Identity Manager Driver Overview, click the icon for the driver.

5 Select an Input Transformation icon or an Output Transformation icon.



Identity Manager Driver Overview

Driver: Active Directory.DriverSet.vmp

Running on server(s):

fb110.vmp

Input Transformation Policies

InputTransform.Active Directory.DriverSet.novell
Password(Pub)-Sub Email Notifications.Active Directory.DriverSet.novell

6 Select a policy, then click *Edit*.

7 Click a rule.

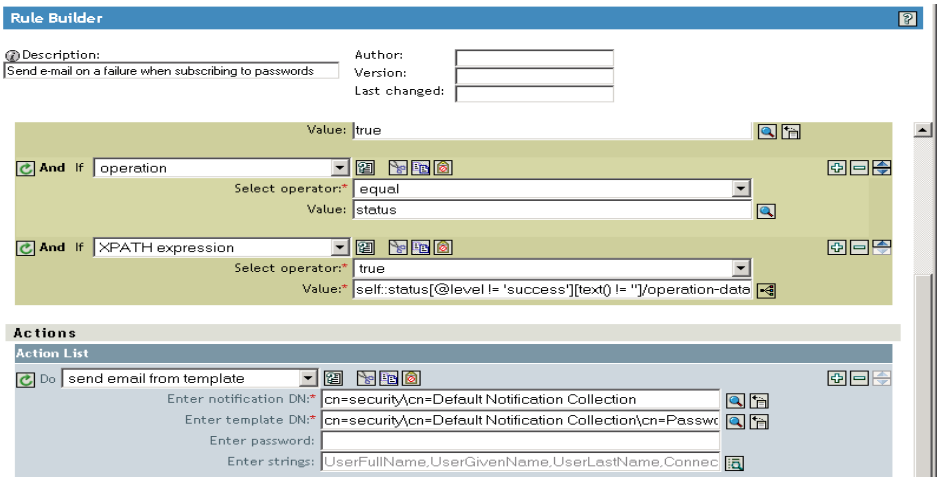
8 Specify the password for the SMTP server in the rules that include Do Send E-mail from Template actions.

  For example, if you are using the sample driver configurations, the following Password Synchronization policies need to be modified.

Policy Set	Policy Name	Rule Name
Input Transformation	Password(Pub)-Sub Email Notifications	◆ Send e-mail on a failure when subscribing to passwords ◆ Send e-mail on failure to reset the connected system password using the Identity Manager data store password

Policy Set	Policy Name	Rule Name
Output Transformation	Password(Sub)-Pub Email Notifications	◆ Send e-mail for a failed publish password operation

The following figure shows an example of a Do Send E-mail from Template action that requires the password.



Rule Builder

Description: Send e-mail on a failure when subscribing to passwords

Author:

Version:

Last changed:

Value: true

And If operation

Select operator:* equal

Value: status

And If XPATH expression

Select operator:* true

Value:* self::status[@level != 'success'][text() != '']/operation-data

Actions

Action List

Do send email from template

Enter notification DN:* cn=security\cn=Default Notification Collection

Enter template DN:* cn=security\cn=Default Notification Collection\cn=Passwc

Enter password:

Enter strings: UserFullName,UserGivenName,UserLastName,Connec

The password is obfuscated when it is stored in the Identity Vault.

9 Select (mark) the rule, then click *OK*.

5.12.5 Adding Your Own Replacement Tags to E-Mail Notification Templates

The e-mail notification templates have some tags defined by default, to help you personalize the message for the user. You can also add your own tags.

Your ability to add tags is dependent on the application that is using the e-mail template.

- "Adding Replacement Tags to Password Synchronization E-Mail Notification Templates" on page 166
- "Adding Replacement Tags to Forgotten Password E-Mail Notification Templates" on page 172

Adding Replacement Tags to Password Synchronization E-Mail Notification Templates

You can add replacement tags to the e-mail notification templates for Password Synchronization, but these tags won't work unless you also define them in every password synchronization policy rule that refers to the e-mail notification template. When using a DoSendEmailFromTemplate action, all replacement tags declared within the template must be defined as child arg-strings elements of the action.

For example, Identity Manager provides default replacement tags that are included with the e-mail notification templates. Identity Manager also provides default password synchronization policies in the driver configurations. Each default tag provided with the e-mail template is also defined in each rule of the password synchronization policy that uses that e-mail template.

For example, the UserGivenName tag is one of the default tags defined in the e-mail template named Password Set Fail. A policy rule named *Send e-mail on a failure when subscribing to passwords* refers to that e-mail template in a DoSendEmailFromTemplate action. This rule is used in a policy to notify to a user when a password fails to synchronize. The same UserGivenName tag is defined as an arg-string element in that rule.

Like this example, each new tag you add must be defined in both the e-mail template and the policy rules that refer to the e-mail template, so that the Metadirectory engine knows how to insert the correct data in place of the replacement tag when sending the e-mail to the user.

You can refer to the tags in the Identity Manager driver configurations that shipped with Identity Manager as examples.

Keep in mind the following guidelines:

- The items called replacement tags in the e-mail templates are called tokens in the context of Policy Builder.
- You should use Policy Builder to make it easier to define the argument strings for the replacement tags, as explained in the steps in this section.
- The tags you add might be defined to be any of the following:
  - Any Source or Destination attribute for the user

    Unlike adding tags for the e-mail templates for Forgotten Password, simply adding a tag that has the same name as an attribute on the User object in the Identity Vault does not cause the tag to work. As with all tags used in password synchronization e-mail notification templates, you must also define the tag in the policy that is referring to the e-mail template.
  - A global configuration value
  - An XPATH expression

  This is in contrast to tags for the e-mail templates for Forgotten Password, which are limited to eDirectory user attributes.
- Unlike adding tags for the e-mail templates for Forgotten Password (which require you to use the exact name of an eDirectory user attribute), you can name the replacement tags any name you choose, as long as it matches the name used to define the tag in the policies that reference the e-mail template.

To define the tags in a policy, find all the policies that refer to the e-mail notification template, and use Policy Builder to add the tags to them. In each policy, edit each rule that refers to the template.

One way to make sure that you find all the policies that refer to the e-mail notification templates is to export your driver configurations, then search the XML for a do-send-e-mail action that has the template equal to the name of the e-mail notification template.

1 In iManager, select *Identity Manager > Identity Manager Overview*.

2 Select the driver set that contains the driver with the policy you want to edit.

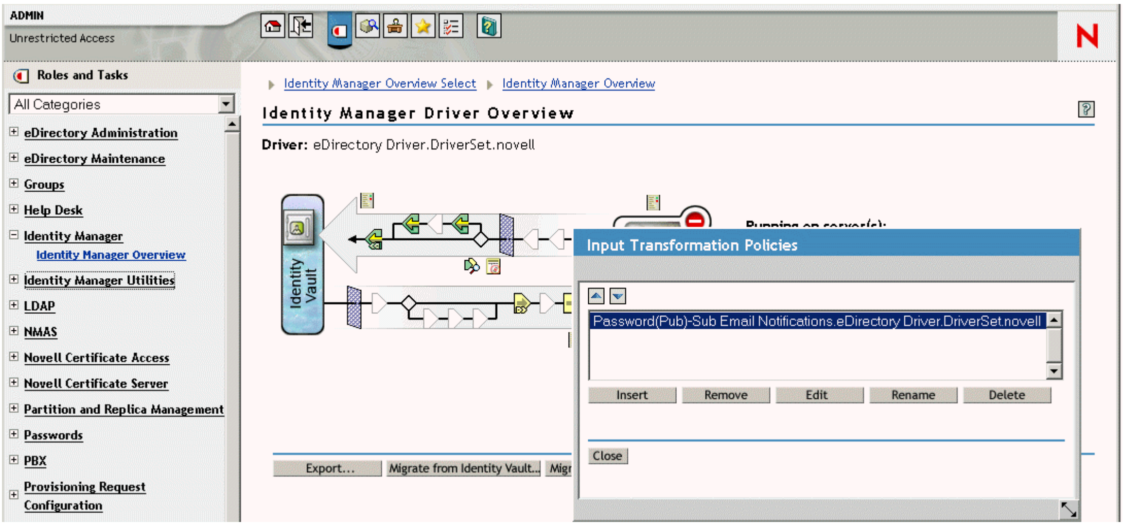
3 Click the icon for the driver that has the policy you want to edit.

4 On the Publisher or Subscriber channel, click the set of policies that contains the policy you want to edit.

For example, the driver configuration for the eDirectory driver that ships with Identity Manager contains a policy in the Input Transformation policy set which references both password synchronization e-mail notification templates.

5 Click the policy, then click *Edit*.

The following figure illustrates how to edit the Password(Pub)-Sub Email Notifications policy for the eDirectory driver:



ADMIN
Unrestricted Access
Roles and Tasks
All Categories
eDirectory Administration
eDirectory Maintenance
Groups
Help Desk
Identity Manager
Identity Manager Overview
Identity Manager Utilities
LDAP
NMAS
Novell Certificate Access
Novell Certificate Server
Partition and Replica Management
Passwords
PBX
Provisioning Request Configuration
Identity Manager Overview Select
Identity Manager Overview
Identity Vault
Running on server(s):
Export...
Migrate from Identity Vault...
Migr
Insert
Remove
Edit
Rename
Delete
Close

Identity Manager Driver Overview

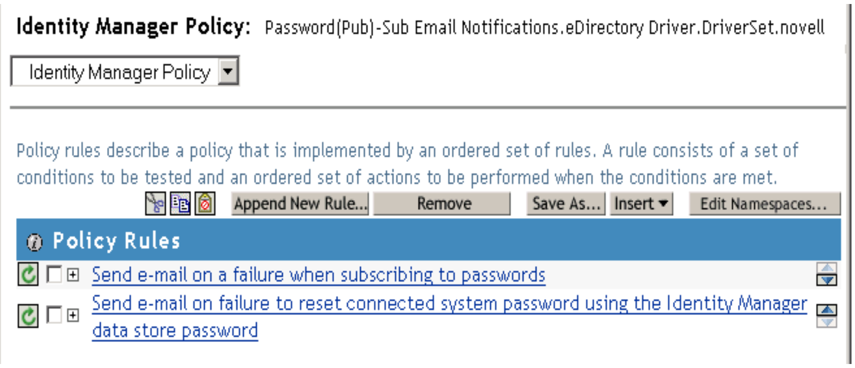
Driver: eDirectory Driver.DriverSet.novell

Input Transformation Policies

Password(Pub)-Sub Email Notifications.eDirectory Driver.DriverSet.novell

6 In the list of rules that opens, click the rule that refers to the e-mail notification template.

For example, in the Password(Pub)-Sub Email Notifications policy, you would see this list of rules. Both of these rules reference one of the password synchronization e-mail templates. You need to edit both rules if you are adding tags to both templates.



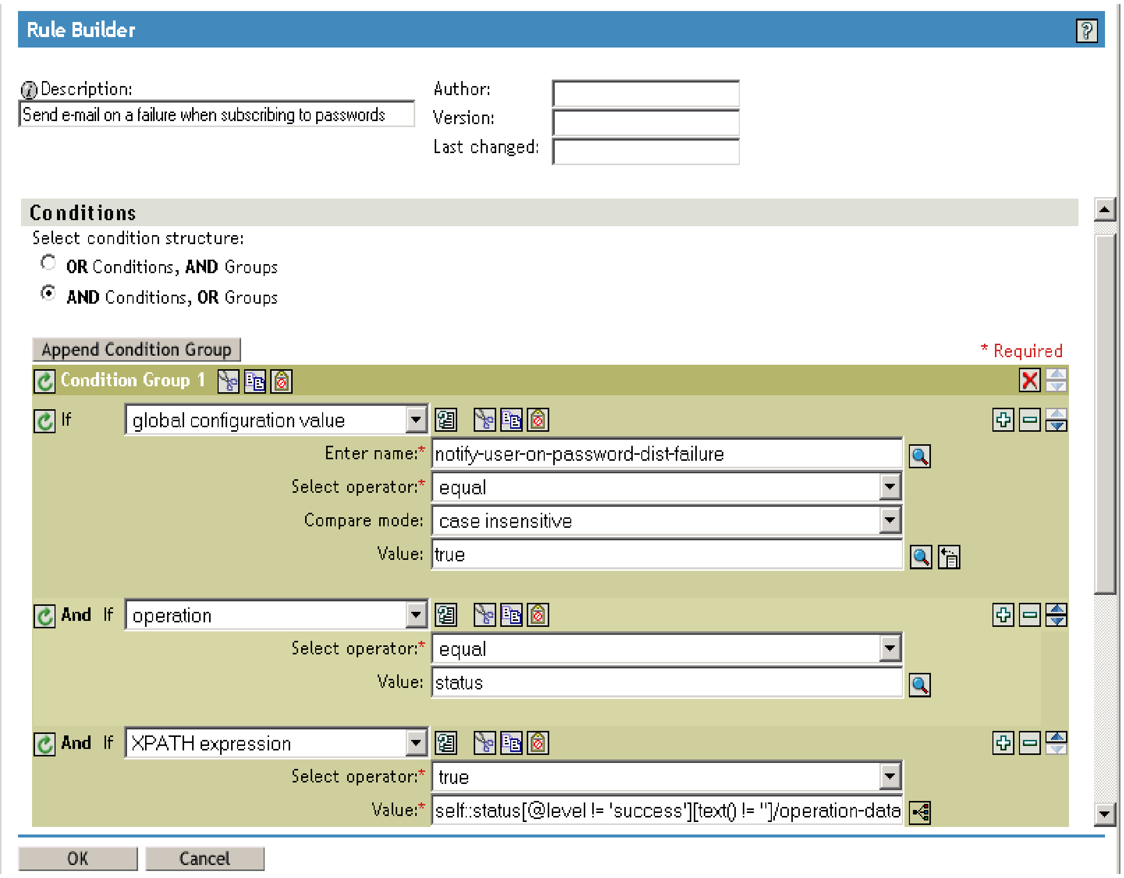
Identity Manager Policy: Password(Pub)-Sub Email Notifications.eDirectory Driver.DriverSet.novell

Policy rules describe a policy that is implemented by an ordered set of rules. A rule consists of a set of conditions to be tested and an ordered set of actions to be performed when the conditions are met.

Policy Rules

- □ Send e-mail on a failure when subscribing to passwords
- □ Send e-mail on failure to reset connected system password using the Identity Manager data store password

If you click the first rule, the following page appears:



Rule Builder

Description: Send e-mail on a failure when subscribing to passwords

Author:

Version:

Last changed:

Conditions

Select condition structure:

○ OR Conditions, AND Groups

● AND Conditions, OR Groups

* Required

Condition Group 1

If global configuration value

Enter name:* notify-user-on-password-dist-failure

Select operator:* equal

Compare mode: case insensitive

Value: true

And If operation

Select operator:* equal

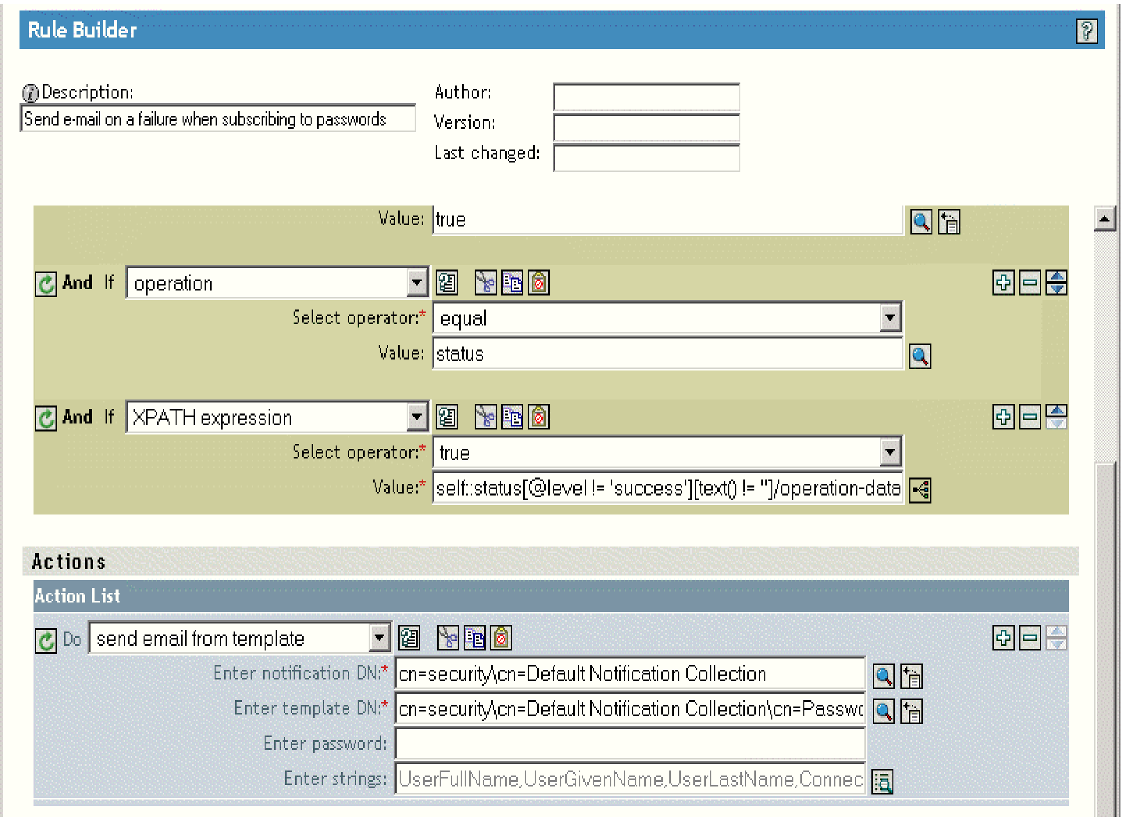
Value: status

And If XPATH expression

Select operator:* true

Value:* self::status[@level != 'success'][text() != '']/operation-data

7 Scroll to the *Actions* section.



Rule Builder

Description: Send e-mail on a failure when subscribing to passwords

Author:

Version:

Last changed:

Value: true

And If operation

Select operator:* equal

Value: status

And If XPATH expression

Select operator:* true

Value:* self::status[@level != 'success'][text() != '']/operation-data

Actions

Action List

Do send email from template

Enter notification DN:* cn=security\cn=Default Notification Collection

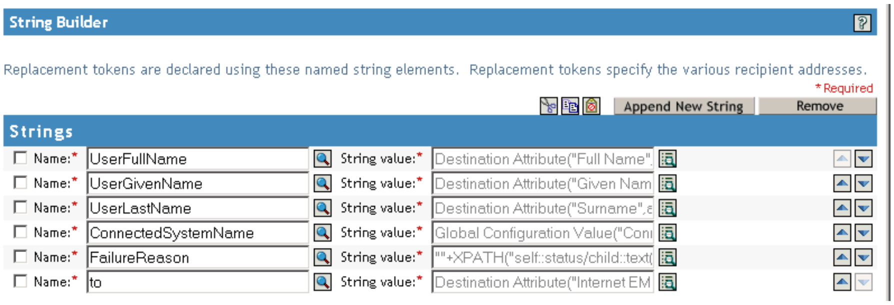
Enter template DN:* cn=security\cn=Default Notification Collection\cn=Passwo

Enter password:

Enter strings: UserFullName,UserGivenName,UserLastName,Connec

8 For the *Do Send Email from Template* rule, click the browse button for the *Enter strings* field.

This opens the string builder. For the example rule, the following figure shows the list of strings you would see. The default tags that are used in the e-mail notification templates are already defined in the password synchronization policies that are part of the Identity Manager driver configurations, like this one. You can use the default tags as an example.



String Builder

Replacement tokens are declared using these named string elements. Replacement tokens specify the various recipient addresses.

* Required

Strings

□ Name:*	UserFullName	String value:*	Destination Attribute("Full Name"
□ Name:*	UserGivenName	String value:*	Destination Attribute("Given Nam
□ Name:*	UserLastName	String value:*	Destination Attribute("Surname",a
□ Name:*	ConnectedSystemName	String value:*	Global Configuration Value("Con
□ Name:*	FailureReason	String value:*	""+XPATH("self::status/child::text(
□ Name:*	to	String value:*	Destination Attribute("Internet EM

9 To define a tag that you could use in an e-mail notification template, click *Append New String*, then enter a name for the tag.

Make sure that the name is exactly the same name you use in the e-mail notification template.

10 In the *String value* field, click the browse button to help you define the tag.

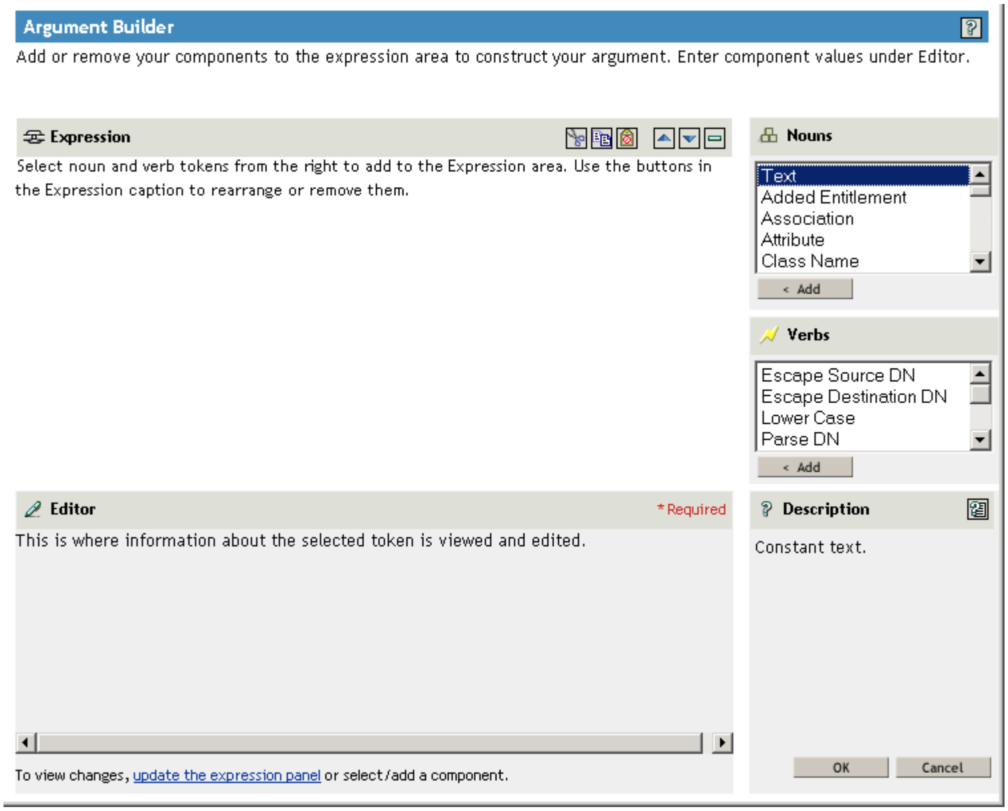
11 On the Argument Builder page, specify what value should be brought in when this tag is used in an e-mail notification template.

You can define the tag to be any of the following:

- Any Source or Destination attribute for the user

  Unlike adding tags for the e-mail templates for Forgotten Password, simply adding a tag that has the same name as an attribute on the user object in the Identity Vault does not cause the tag to work. As with all tags used in password synchronization e-mail notification templates, you must also define the tag in the policy that is referring to the e-mail template.
- A global configuration value
- An XPATH expression

The following figure illustrates how to define the tag:



Argument Builder

Add or remove your components to the expression area to construct your argument. Enter component values under Editor.

Expression

Select noun and verb tokens from the right to add to the Expression area. Use the buttons in the Expression caption to rearrange or remove them.

Editor

This is where information about the selected token is viewed and edited.

To view changes, update the expression panel or select/add a component.

Description

Constant text.

After you define the tag and click *OK*, it shows up as one of the strings in the String Builder page.

12 Make sure you click *OK* to complete all the pages, so that your changes to the policy are saved.

13 Repeat the steps to edit the rules in all the policies that refer to the e-mail notification template.

14 Add the tag you defined in the policy to the e-mail notification template, using the exact name you used in the policies.

At this point, you can use the tag name in the body of the e-mail notification template.

15 Save the changes and restart the driver.

Adding Replacement Tags to Forgotten Password E-Mail Notification Templates

Using the following guidelines, you can add tags to the e-mail notification templates for Forgotten Password:

- You can add only tags that correspond to LDAP attributes on the User object that the message is being sent to.
- The name of the tag you add must be exactly the same as the LDAP attribute name on the user object.

  To see how LDAP attributes correspond to eDirectory attribute names, you can refer to the Schema Mapping Policy that is provided in the Identity Manager Driver for LDAP.
- No other configuration is necessary.

5.12.6 Sending E-Mail Notifications to the Administrator

The default configuration is for the e-mail notification to go only to the user. The policies that ship with Identity Manager use the e-mail address from the Identity Vault object for the user that is affected.

However, you can configure the password synchronization policies so that e-mail notifications also go to the administrator. To do this, you must modify the Identity Manager script for one of the policies.

Send a Blind Copy to the administrator by defining the token with the administrator's e-mail address.

To copy an administrator, modify the policy that generates the e-mail (such as PublishPasswordEmails.xml, in which the policy looks up the e-mail address to send notifications) and add an additional `<arg-string>` element with the administrator's e-mail address.

The following example illustrates the additional arg-string element:

```
<arg-string name="to">

   <token-text>Admin@company.com</token-text>

</arg-string>
```

Make sure to restart the driver after making these changes.

5.12.7 Localizing E-Mail Notification Templates

Keep in mind the following:

- The default templates are in English, but you can edit the text to use other languages.
- The names and the definitions of the replacement tags must remain in English, so that the arg-string token definitions in the policies match the names of the replacement tags.
- For Forgotten Password e-mail notifications only, to specify what encoding you want on your mail item, you need to add a setting in the `portalservlet.properties` file. For example:

  ```
  ForgottenPassword.MailEncoding=EUC-JP
  ```

  If this setting doesn't exist, then no encoding is used on the mail transformation.

- For Password Synchronization e-mail messages, an XML attribute named charset can be specified on the following elements: <`mail`>, <`message`>, and <‘>.

  For information on using these elements, see the *DirXML Driver for Manual Task Service Implementation Guide* (http://www.novell.com/documentation/dirxmldrivers/index.html), which gives more detail on the e-mail templates.

5.13 Troubleshooting Password Synchronization

- See the tips in Section 5.8, “Implementing Password Synchronization,” on page 124.
- Make sure you have the Simple Password Login Method installed with NMAS.
- Make sure you have a copy of the root of the tree on the servers where you need to NMAS to enforce password policies on eDirectory login methods or on passwords from connected systems being synchronized by Identity Manager.
- Make sure that the users requiring password synchronization are replicated on the same server with the driver that is synchronizing the passwords. As with other driver functions, the driver can manage only the users that are in a master or read/write replica on the same server.
- Make sure SSL is configured properly between the Web server and the Identity Vault.
- If you see an error about a password not complying when a user is initially created, but the password is set correctly in the Identity Vault, the default password in the driver policy might not conform to the password policy that applies to that user.

  The following scenario uses the Active Directory driver. However, the same issue could occur for another driver.

  Providing an Initial Password: You want the Active Directory driver to provide the initial password for a user when the driver creates a new User object in the Identity Vault to match a user in Active Directory. The sample configuration for the Active Directory driver sends the initial password as a separate operation from adding the user, and the sample configuration also includes a policy that provides a default password for a user if no password is provided by Active Directory.

  Because adding the user and setting the password are done separately, a new user always receives the default password, even if only momentarily. The default password is soon updated because the Active Directory driver sends the password immediately after adding the user. If the default password does not comply with the Identity Vault password policy for the user, an error is displayed.

  For example, if a default password created by using the user’s surname is too short to comply with the password policy, you might see a -216 error saying the password is too short. However, the situation is soon rectified if the Active Directory driver then sends an initial password that does comply

  Regardless of the driver you are using, if you want a connected system that is creating User objects to provide the initial password, consider doing one of the actions listed below. These measures are especially important if the initial password does not come with the Add event but instead comes in a subsequent event.

  - Change the policy on the Publisher channel that creates the default password, so that the default password conforms to the password policies that have been defined for your organization in the Identity Vault. (Select *Passwords*, then select *Password Policies*.)

    When the initial password comes from the authoritative application, it replaces the default password.

  This option is preferable because we recommend that a default password policy exist in order to maintain a high level of security within the system.

  - On the Publisher channel, remove the policy that creates the default password. In the sample configuration, this policy is provided in the Command Transformation policy set. Adding a user without a password is allowed in the Identity Vault. The assumption for this option is that the password for the newly created User object eventually comes through the Publisher channel, and the User object exists without a password for only a short time.

- Password policies are assigned with a tree-centric perspective. In contrast, Password Synchronization is set up per driver. Drivers are installed on a per-server basis and can manage only those users who are in a master or read/write replica.

  To get the results you expect from Password Synchronization, make sure that the containers that are in a master or read/write replica on the server running the drivers for Password Synchronization match the containers where you have assigned password policies with Universal Password enabled. Assigning a password policy to a partition root container ensures that all users in that container and subcontainers are assigned the password policy.

- Helpful DSTrace commands:

  +*DXML*: To view Identity Manager rule processing and potential error messages.
  +*DVRS*: To view Identity Manager driver messages.
  +*AUTH*: To view NDS password modifications.
  +*DCLN*: To view NDS DCLient messages.

Creating and Using Entitlements

6

Identity Manager allows you to synchronize data between connected systems. Entitlements allow you to set up criteria for a person or group that, once met, initiate an event to grant or revoke access to business resources within the connected system. This gives you one more level of control and automation for granting and revoking resources.

There are two aspects to making entitlements work: creating the entitlement and managing the entitlement. You create entitlements through iManager or through Designer. To create an entitlement through iManager, select the *Create Entitlement* Option under the Identity Manager Utilities heading in iManager. For more information, see Section 6.4, “Writing Entitlements in XML through iManager,” on page 180.

You can also use Designer to create entitlements and deploy them into existing Identity Manager drivers. Designer allows you to create entitlements through the Entitlement Wizard, which gives you a graphical interface through which to create the entitlement, and steps you through the process. In iManager, you create entitlements through a simple interface, but you add additional properties through an XML editor. Because it has a graphical interface, we recommend using Designer for creating and editing entitlements.

After you create entitlements (or use entitlements that come already configured with certain Identity Manager drivers), you need to manage them. Entitlements are managed by two packages or agents: through iManager as Role-Based Entitlement Policies or through the User Application in workflow-based provisioning. For entitlements used in workflow-based provisioning, see “*Identity Manager 3.5.1 User Application: Administration Guide*Configuring Provisioning Request Definitions” in the *Identity Manager 3.5.1 User Application: Administration Guide*.

Role-Based Entitlement policies allow you to grant business resources if the criteria are met. For example, if a user meets criteria 1, 2, and 3, then through a Role-Based Entitlement policy, the user becomes a member of Group H; but if the user meets 4 and 5, he or she becomes a member of Group I. In order for this entitlement to work through workflow-based provisioning, approval is first required.

6.1 Terminology

Following are some terms that you will see used throughout this chapter.

Table 6-1 *Terminology*

Terms	Explanation
Entitlement	An Identity Vault object that represents a business resource in a connected system.
Entitlement Agent	Grants and revokes entitlements. For Role-Based Entitlements, the agent is the Entitlement Service driver.
Grant or Revoke	The interpretation of granting or revoking an entitlement is controlled by Global Configuration Variables (GCVs) on an Identity Manager driver.
Entitlement Consumer	Anything that uses entitlement-related information. Entitlement consumers include iManager, the User Application, and Identity Manager policies.

6.2 Creating Entitlements: Overview

- Section 6.2.1, "Identity Manager Drivers with Configurations that Support Entitlements," on page 177
- Section 6.2.2, "Enabling Entitlements on Other Identity Manager Drivers," on page 178

You must know beforehand what you want to accomplish with entitlements. Entitlements work from the functionality you build into Identity Manager drivers through policies. These driver policies implement rules and process the events between the Identity Vault and the connected system. If the policies in the Identity Manager driver do not specify what you want to do, entitlements cannot work. For example, if you don't specify the action section of the Check User Modify for Group Membership rule in the Command policy, attempts to grant or revoke a group membership entitlement are ignored.

You need to know precisely what you want to accomplish with Identity Manager, then you can correctly design granting and revoking capabilities for any connected system resources. The following four-step procedure can help you plan to create and use entitlements:

1. Know what you want to accomplish in your business situation. You can design and implement almost anything through Identity Manager, but you need to know what you want to do before implementing something that isn't defined. Make a numbered list of what you want to do.
2. Define an entitlement that represents one point from your numbered list. You can create valueless and valued entitlements. Valued entitlements can get their values from an external query, they can be administrator defined, or they can be free form. There are examples in Section 6.4.6, "Example Entitlements To Help You Create Your Own Entitlements," on page 190.

3. Add policies to the Identity Manager Driver to implement the designed entitlement. To create a policy for an Identity Manager driver, you need to be conversant in XSLT or DirXML script, in the way the connected system handles and receives information, and with the way Novell® eDirectory™ stores information. Unless you are a good DirXML* programmer, this is a job for consultants.
4. Set up a managing agent to grant or revoke the entitlement. If you want an automated process, use Role-Based Entitlements; if you want a manual process, use workflow-based provisioning.

6.2.1 Identity Manager Drivers with Configurations that Support Entitlements

Identity Manager includes a number of drivers with configuration files that already contain entitlements, policies to implement the entitlements, and the driver enabled to listen for entitlement activities. You must enable entitlements as you initially install the driver in order to make the preconfigured elements part of the driver. The following drivers have configuration files that support entitlements:

- Active Directory
- Exchange
- GroupWise
- LDAP
- NIS
- Lotus Notes
- NT Domain
- RACF

These preconfigured drivers fulfill the first three of the four steps outlined above. The types of example entitlements the drivers contain can be used for the most common scenarios: granting and revoking user accounts, groups, and email distribution lists. These include:

- Active Directory: Grant and revoke accounts, group membership, Exchange Mailbox
- Exchange 5.5: Grant and revoke mailbox and group membership
- GroupWise: Grant and revoke accounts, grant and revoke members of distribution lists
- LDAP: Grant and revoke user accounts
- Linux and UNIX: Grant and revoke accounts
- Lotus Notes: Grant and revoke user accounts and group memberships
- NT Domain: Grant and revoke user accounts and group membership
- RACF: Grant and revoke group accounts and group memberships

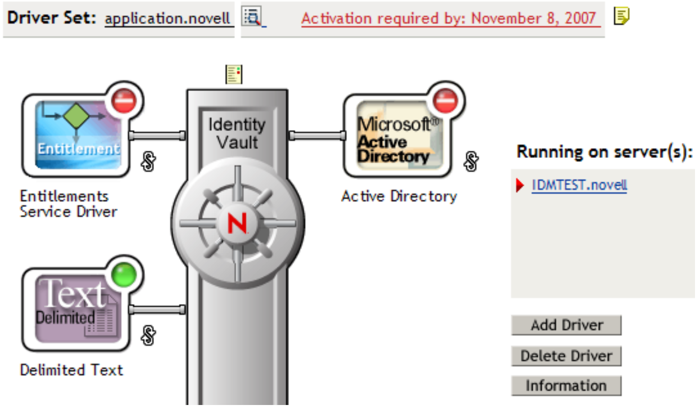
These are example entitlements and policies that you can use as is (if they meet your needs); you can also tweak them to meet your needs, or you can use them as examples and make your own through iManager or Designer. Again, if you want to use the preconfigured driver's entitlements, you must enable entitlements when you initially create the preconfigured driver in Designer or iManager; preconfigured entitlements cannot be added later without re-creating the driver.

If you have been using entitlements with Identity Manager 2.x and you want to use those entitlements with Identity Manager 3.5.1, run the *Upgrade Entitlements* option under *Identity Manager Utilities*.

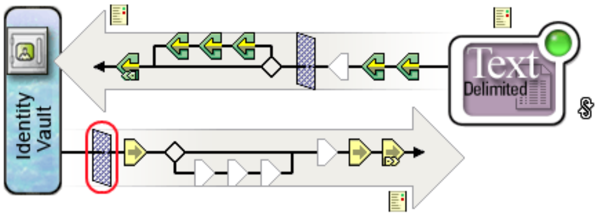
6.2.2 Enabling Entitlements on Other Identity Manager Drivers

You can still use entitlements on Identity Manager drivers that do not contain entitlement preconfigurations. To enable your driver to support entitlements, add the DirXML-EntitlementRef attribute to your driver filter. To do this:

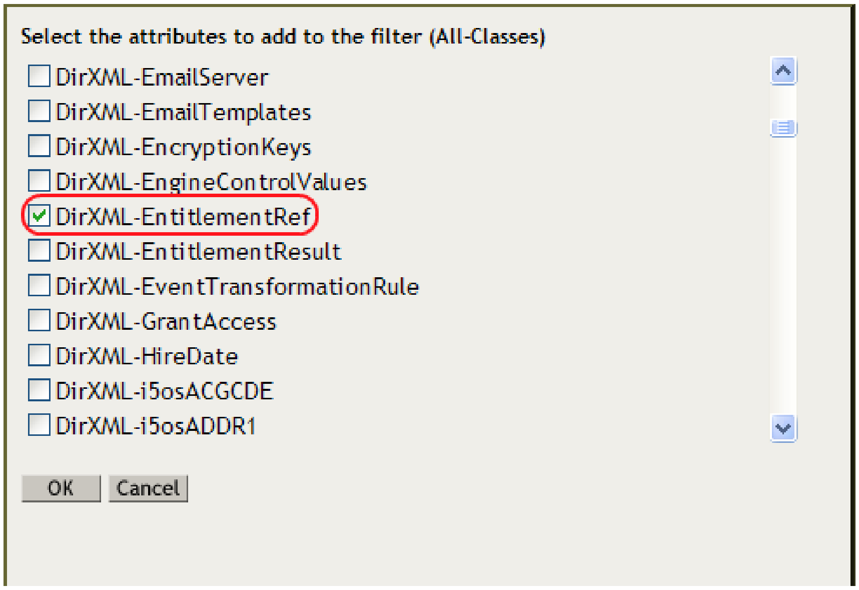
1. Select *Identity Manager > Identity Manager Overview*.
2. Browse to the driver set where the driver resides, then click *Search*.
3. From the Identity Manager Overview page, click the Driver object.


Driver Set: application.novell
Activation required by: November 8, 2007
Running on server(s):
IDMTEST.novell

4. On the Driver Overview page click the *Driver Filter* icon right of the Identity Vault (circled in red).



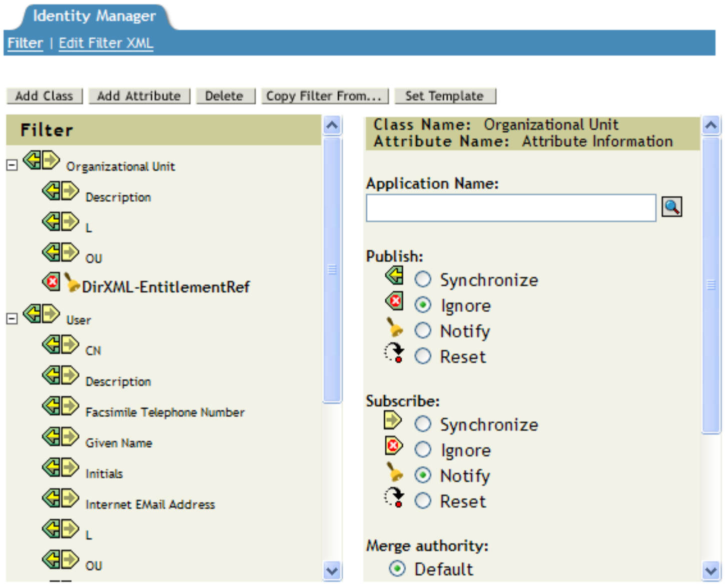
5. Select *Add Attribute*, then scroll to the bottom and select *Show all attributes*.
6. Select the *DirXML-EntitlementRef* attribute, then click *OK*.



Select the attributes to add to the filter (All-Classes)

- ☐ DirXML-EmailServer
- ☐ DirXML-EmailTemplates
- ☐ DirXML-EncryptionKeys
- ☐ DirXML-EngineControlValues
- ☑ DirXML-EntitlementRef
- ☐ DirXML-EntitlementResult
- ☐ DirXML-EventTransformationRule
- ☐ DirXML-GrantAccess
- ☐ DirXML-HireDate
- ☐ DirXML-i5osACGCDE
- ☐ DirXML-i5osADDR1

7 Select *DirXML-EntitlementRef* in the Filter page, then under the Subscribe heading select *Notify*.



Class Name: Organizational Unit
Attribute Name: Attribute Information

Application Name:

Publish:
- ○ Synchronize
- ● Ignore
- ○ Notify
- ○ Reset

Subscribe:
- ○ Synchronize
- ○ Ignore
- ● Notify
- ○ Reset

Merge authority:
- ● Default

8 Click *OK* to save the changes.

This process is performed automatically when you create entitlements through Designer on a driver.

6.3 Entitlement Prerequisites

- ❑ eDirectory 8.7.3 or later
- ❑ Identity Manager 2.x or 3.x
- ❑ An Entitlements Service driver

You must have an Entitlements Service driver in each driver set where you want to use entitlements. This requires a very simple, one-time setup for each driver set.

❑ A driver configuration that supports entitlements

Before you can use entitlements with a connected system, do one of the following:

- Import the Identity Manager driver configuration for the driver and specify that the driver has entitlements enabled.
- Enable your driver to support entitlements. To do this:
    1. Create entitlements using iManager or Designer (Designer is preferred).
    2. Add the DirXML-EntitlementRef attribute to your driver filter as described in Section 6.2.2, "Enabling Entitlements on Other Identity Manager Drivers," on page 178.
    3. Write policies to implement the entitlements you create in Step 1.

6.4 Writing Entitlements in XML through iManager

To help you better understand what needs to go into an entitlement, you can look at the entitlements and policies in one of the preconfigured drivers, Active Directory, that has entitlements enabled. This includes examining Novell's Entitlement DTD (Document Type Definition), then looking at XML examples of writing entitlements based on the DTD.

In this section:

- Section 6.4.1, "What the Active Directory Driver Adds When Entitlements Are Enabled," on page 180
- Section 6.4.2, "Using Novell's Entitlement Document Type Definition (DTD)," on page 184
- Section 6.4.3, "Explaining the Entitlement DTD," on page 185
- Section 6.4.4, "Creating Entitlements Through Designer," on page 188
- Section 6.4.5, "Creating and Editing Entitlements in iManager," on page 188
- Section 6.4.6, "Example Entitlements To Help You Create Your Own Entitlements," on page 190
- Section 6.4.7, "Completing the Creating Entitlements Steps," on page 195

6.4.1 What the Active Directory Driver Adds When Entitlements Are Enabled

The Active Directory driver with entitlements enabled contains the following changes to its structure:

- Adds the DirXML-EntitlementRef attribute to the driver filter. The DirXML-EntitlementRef attribute allows the driver filter to listen for entitlement activities.
- Creates a User Account Entitlement. The User Account entitlement grants or revokes an account in Active Directory for the user. When the account is granted, the user is given an enabled logon account. When the account is revoked, the logon account is either disabled or deleted, depending on how the driver is configured.

- Creates a Group Membership Entitlement. The Group Entitlement grants or revokes membership in a group in Active Directory. The group must be associated with a group in the Identity Vault. When membership is revoked, the user is removed from the group. The group membership entitlement is not enforced on the Publisher channel; if a user is added to a controlled group in Active Directory by some external tool, the user is not removed by the driver. Further, if the entitlement is removed from the user object instead of being simply revoked, the Active Directory driver takes no action.
- Creates an Exchange Mailbox Entitlement. The Group Entitlement grants or revokes an Exchange mailbox for the user in Microsoft Exchange.
- Adds entitlement information to many policies.

The following policies contain additional rules that allow entitlements to work properly:

- InputTransform (driver level). The Check Target Of Add Association For Group Membership Entitlements rule in this policy checks the target of “add-association” for group membership entitlements. Group membership entitlements assigned to users being created in Active Directory cannot be processed until the user is successfully created. Add-association signals that an object has been created by the driver in Active Directory. If the object is also tagged for group entitlement processing, it performs the work now.
- Event Transform (Publisher channel). The Disallow User Account Delete rule in this policy disallows a user account delete in the Identity Vault. When using the User Account Entitlement, managed user accounts are controlled by the entitlement in the Identity Vault. A delete in Active Directory does not delete the controlling object in the Identity Vault. A future change to the object in the Identity Vault or a merge operation might re-create the account in Active Directory.
- Command (Subscriber channel). The Command policy contains the following rules pertaining to entitlements:
    - The User Account Entitlement Change (Delete Option) rule. The User Account Entitlement grants the user an enabled account in Active Directory. Revoking the entitlement disables or deletes the Active Directory account depending on the value you select for the *When account entitlement revoked* global variable. This rule executes when the entitlement is changing and you have selected the Delete option.
    - The User Account Entitlement Change (Disable Option) rule. The User Account Entitlement grants the user an enabled account in Active Directory. Revoking the entitlement disables or deletes the Active Directory account depending on the value you select for the *When account entitlement revoked* global variable. This rule executes when the entitlement is changing and you have selected the Disable option.
    - The Check User Modify for Group Membership Being Granted or Revoked rule.
    - The Check User Modify for Exchange Mailbox Being Granted or Revoked rule.
- Matching (Subscriber channel). This is the Account Entitlement: Do Not Match Existing Accounts rule for this policy. When using the User Account entitlement with the Identity Manager user application or Role-Based Entitlements, accounts are created and deleted (or disabled) by granting or revoking the entitlement. The default policy does not match an existing account in Active Directory if the user is not entitled to an account in Active Directory. Modify or remove this rule if you want the entitlement policy to apply to matching accounts in Active Directory. This might result in the Active Directory account being deleted or disabled.

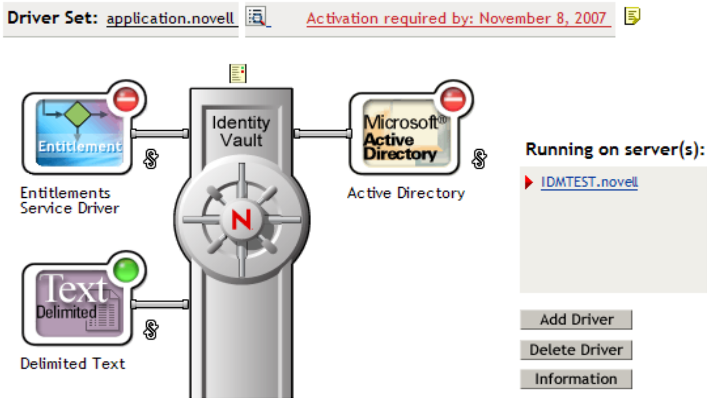
- Creation (Subscriber channel). The Creation policy contains the following rules pertaining to entitlements:
  - Account Entitlement: Block Account Creation When Entitlement Not Granted. When using the User Account entitlement with the Identity Manager user application or Role Based Entitlements, accounts are created only for users that are specifically granted the account entitlement. This rule vetoes user account creation when the entitlement is not granted.
  - Identity Vault Accounts Are Enabled if Login Disabled Does Not Exist.
  - Prepare To Check Group Entitlements After Add. Group entitlements are processed after the add completes, because the added object needs to exist in order to be added to a group. The add is flagged with an operational property that is checked in the input transform when the add processing completes.
  - Signal the Need To Check Exchange Entitlements After the Add.
  - Map User Name to Windows Logon Name. When userPrincipalName is configured to follow the eDirectory user name, set userPrincipalName to the eDirectory object name plus the name of the Active Directory domain.

You can see the actual XML code for each policy by performing the following steps in iManager:

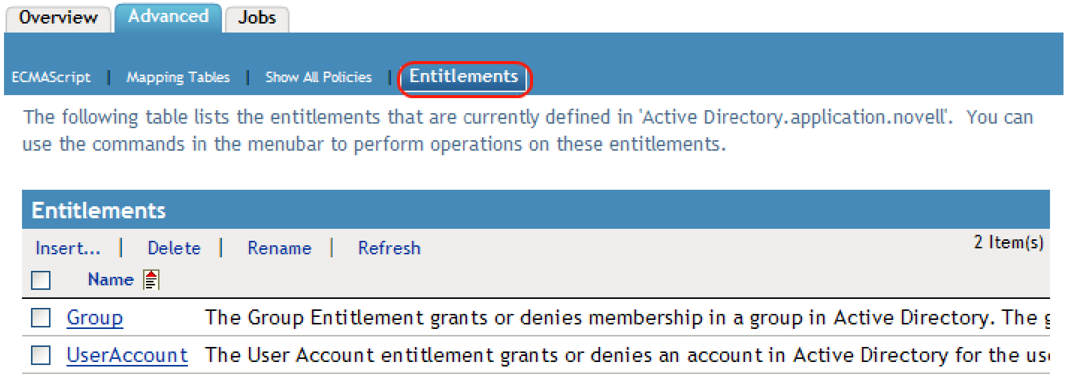
1 Select *Identity Manager > Identity Manager Overview*.

2 Browse to the driver set where the driver resides, then click *Search*.

3 From the Identity Manager Overview page, click the Driver object.



4 On the Driver Overview page click the *Advanced* tab, then click *Entitlements*.


Overview
Advanced
Jobs
ECMAScript | Mapping Tables | Show All Policies | Entitlements

The following table lists the entitlements that are currently defined in 'Active Directory.application.novell'. You can use the commands in the menubar to perform operations on these entitlements.

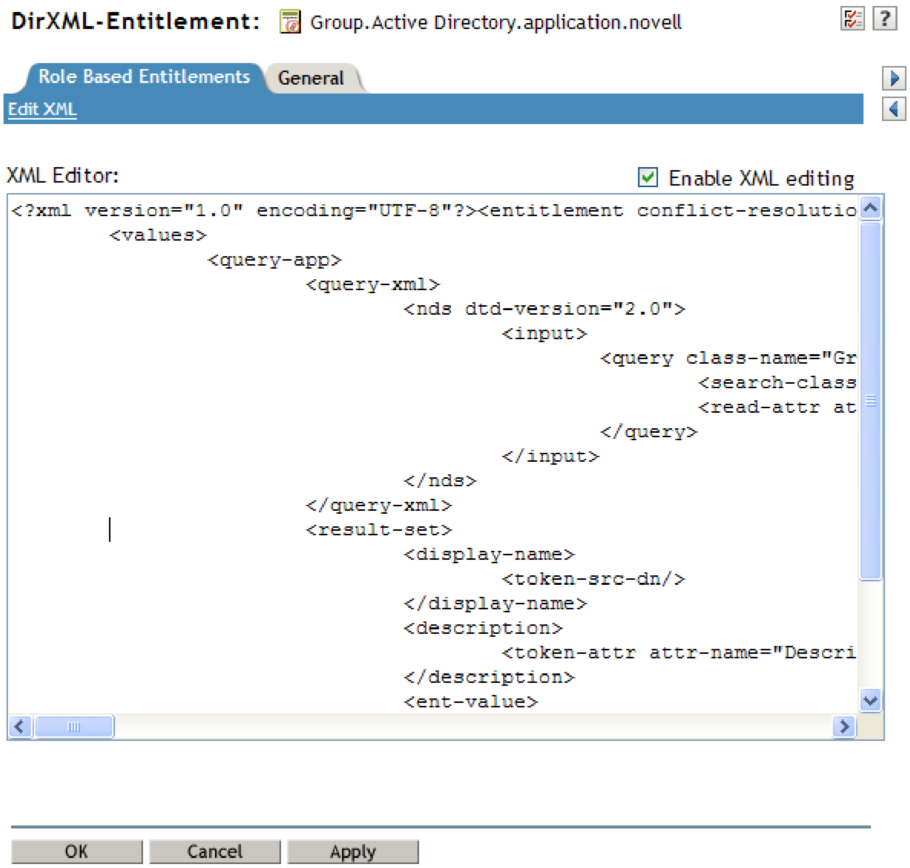
Entitlements

Insert... | Delete | Rename | Refresh

2 Item(s)

□	Name
□	Group	The Group Entitlement grants or denies membership in a group in Active Directory. The g
□	UserAccount	The User Account entitlement grants or denies an account in Active Directory for the us

5 Click the entitlement name to view the policy in XML.

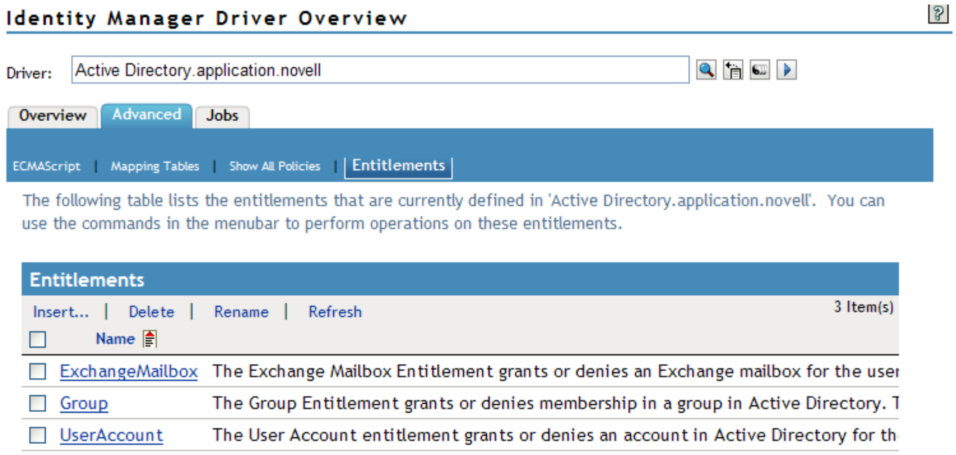


```
<?xml version="1.0" encoding="UTF-8"?><entitlement conflict-resolutio
        <values>
                <query-app>
                        <query-xml>
                                <nds dtd-version="2.0">
                                        <input>
                                                <query class-name="Gr
                                                        <search-class
                                                        <read-attr at
                                                </query>
                                        </input>
                                </nds>
                        </query-xml>
                        <result-set>
                                <display-name>
                                        <token-src-dn/>
                                </display-name>
                                <description>
                                        <token-attr attr-name="Descri
                                </description>
                                <ent-value>
```

For information on creating and editing policies, see *Understanding Policies for Identity Manager 3.5.1* and the selected Identity Manager driver guide (http://www.novell.com/documentation/idm35drivers/index.html) for building policies specific to that driver.

The Active Directory driver with entitlements enabled comes with three entitlements: User Account, Group, and Exchange Mail.

***Figure 6-1** Entitlements That Come with the Active Directory Driver*



The following table lists the entitlements that are currently defined in 'Active Directory.application.novell'. You can use the commands in the menubar to perform operations on these entitlements.

□	Name	
□	ExchangeMailbox	The Exchange Mailbox Entitlement grants or denies an Exchange mailbox for the user
□	Group	The Group Entitlement grants or denies membership in a group in Active Directory. T
□	UserAccount	The User Account entitlement grants or denies an account in Active Directory for th

You can see the XML code for these entitlements as part of the writing samples in Section 6.4.6, “Example Entitlements To Help You Create Your Own Entitlements,” on page 190.

6.4.2 Using Novell’s Entitlement Document Type Definition (DTD)

Some entitlements come predefined on drivers that have entitlements enabled. You can use these entitlements or you can create your own entitlements in iManager or Designer. To help you create your own entitlements, use the following Novell Entitlement DTD as an example to create entitlements.

This explanation of the DTD is followed by four examples of how to write entitlements in this XML format through iManager. If you don’t want to worry about XML formatting, use Designer’s Entitlement Wizard for an easier way to create entitlements.

Novell’s Entitlement DTD

```
<!-*****************************************************************->
<!-- DirXML Entitlements DTD
<!-- Novell Inc.
<!-- 1800 South Novell Place
<!-- Provo, UT 84606-6194
<!-- Version=1.0.0
<!-- Copyright 2005 Novell, Inc. All rights reserved -->
<!--************************************************************* -->
<!--
  Entitlement definition stored in the XmlData attribute of a
      DirXML-Entitlement object.
-->
<!ELEMENT entitlement (values?)>
<!ATTLIST entitlement
  conflict-resolution (priority | union) "priority"
  display-name CDATA #REQUIRED
  description CDATA #REQUIRED
>
<!ELEMENT values (query-app | value+)?>
<!ATTLIST values
  multi-valued (true | false) "true"
>
<!ELEMENT value (#PCDATA)>
<!ELEMENT query-app (query-xml, result-set)>
<!ELEMENT query-xml ANY>
<!ELEMENT result-set (display-name, description, ent-value)>
<!ELEMENT display-name(token-attr | token-src-dn | token-association)>
<!ELEMENT ent-value (token-association | token-src-dn | token-attr)>
<!ELEMENT description (token-association | token-src-dn | token-attr)>
<!ELEMENT token-association EMPTY>
<!ELEMENT token-attr EMPTY>
<!ATTLIST token-attr
  attr-name CDATA #REQUIRED
>
<!ELEMENT token-src-dn EMPTY>
<!--
  Entitlement reference stored in the DirXML-EntitlementRef attribute   of a
DirXML-EntitlementRecipient or a DirXML-SharedProfile object.
-->
```

```
<!ELEMENT ref (src?, id?, param?)>
<!ELEMENT param (#PCDATA)>
<!ELEMENT id (#PCDATA)>
<!ELEMENT src (#PCDATA)>
<!--
    Entitlement result stored in the DirXML-EntitlementResult attribute of a
DirXML-EntitlementRecipient object.
-->
<!ELEMENT result(dn, src, id?, param?, state, status, msg?,timestamp)>
<!ELEMENT dn (#PCDATA)>
<!ELEMENT state (#PCDATA)>
<!ELEMENT status (#PCDATA)>
<!ELEMENT msg ANY>
<!ELEMENT timestamp (#PCDATA)>
<!--
   Cached query results stored in the DirXML-SPCachedQuery attribute of a
DirXML-Entitlement object.
-->
<!ELEMENT items (item*)>
<!ELEMENT item (item-display-name?, item-description?, item-value)>
<!ELEMENT item-display-name (#PCDATA)>
<!ELEMENT item-description (#PCDATA)>
<!ELEMENT item-value (#PCDATA)>
<!--
   Representation of a DirXML-EntitlementRef within the DirXML Script and
within the operation-data of an operation in an XDS document.
-->
<!ELEMENT entitlement-impl (#PCDATA)>
<!ATTLIST entitlement-impl
  name CDATA #REQUIRED
  src CDATA #REQUIRED
  id CDATA #IMPLIED
  state (0 | 1) #REQUIRED
  src-dn CDATA #REQUIRED
  src-entry-id CDATA #IMPLIED
>
```

6.4.3 Explaining the Entitlement DTD

The Entitlement DTD is broken into five parts: definition, reference, result, cached query, and internal reference information. The heading is just a comment and is optional. In the DTD, the heading for the Entitlement Definition is:

```
<!--
Entitlement definition stored in the XmlData attribute of a
      DirXML-Entitlement object.
-->
```

Headings are followed by Elements (ELEMENT) and Attribute lists (ATTLIST). Below is a detailed explanation of the elements and attributes under the Entitlement Definition heading, which is the main heading you need to focus on when creating entitlements.

```
<!ELEMENT entitlement (values?)>
```

The root level element is `<entitlement>`, which can contain a single, optional, child `<values>` element. This is followed by the Attribute list, which includes conflict-resolution, display-name, and description. Conflict resolution uses Priority or Union attribute values.

```
conflict-resolution (priority | union) "priority"
```

Role-Based Entitlements use conflict resolution to determine what should happen when a valued entitlement is applied multiple times to the same object. For example, suppose that user U is a member of Entitlement Policy A and Entitlement Policy B, each of which reference the same valued entitlement E, but with a different set of values. Entitlement E of Entitlement Policy A has values (a, b, c). Entitlement E of Entitlement Policy B has a set of values (c, d, e).

The conflict resolution attribute decides which set of values should apply to user U. If set to union, user U is assigned both sets of values (a, b, c, d, e). If set to priority, user U would get only one set of values, depending upon which Entitlement Policy has a higher priority.

If an entitlement is single-valued, conflicts must be resolved by priority, because a union of values would result in more than one value being applied. Role-Based Entitlements presently uses this attribute; in the future, Workflow Entitlements might also use it.

```
display-name CDATA #REQUIRED
description CDATA #REQUIRED
```

The literal entitlement name is not necessarily what an entitlement should display. The Display-name and Description attributes are intended for end-user display. (In Designer, you have an option to choose a display name for the entitlement instead of using the actual entitlement name.)

```
<!ELEMENT values (query-app | value+)?>
<!ATTLIST values
    multi-valued (true | false) "true"
```

The `<values>` element is optional and indicates that an entitlement is valued. If you do not use this element, it means an entitlement is value-less. An example of a valued entitlement is an entitlement that grants a distribution list. An example of a value-less entitlement is an entitlement that grants an account in an application, such as the User Account entitlement that comes with the Active Directory driver.

Valued entitlements receive their values from three sources. One source is the external application (designated by the `<query-app>` element). Another is from a predefined list of enumerated values (one or more `<value>` elements). The third source is from the entitlement client (a `<values>` element with no `<value>` children). The examples are helpful in explaining the way values work.

Valued entitlements may be single- or multi-valued, and the default is multi-valued. It is the responsibility of the entitlement client to enforce this restriction.

```
<!ELEMENT value (#PCDATA)>
```

Entitlement values are untyped strings.

```
<!ELEMENT query-app (query-xml, result-set)>
```

If values come from an external application (such as an e-mail distribution list), you must specify an application query through the `< query-xml>` element. You extract the results from the query through the `<result-set>` element. We show two examples of this in "Example 2: Application Query Entitlement: External Query" on page 191.

```
<!ELEMENT query-xml ANY>
```

XML queries are XDS formatted. The `<query-xml>` command is used to find and read objects from the connected application. The functionality for DirXML rules, object migration, etc. depends on the driver's implementation of the query command. For more information on XML queries, see Novell's developer documentation on queries (http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/ref/ndsdtd/query.html).

```
<!ELEMENT result-set (display-name, description, ent-value)>
<!ELEMENT display-name(token-attr | token-src-dn | token-association)>
<!ELEMENT ent-value (token-association | token-src-dn | token-attr)>
<!ELEMENT description (token-association | token-src-dn | token-attr)>
<!ELEMENT token-association EMPTY>
<!ELEMENT token-attr EMPTY>
<!ATTLIST token-attr
   attr-name CDATA #REQUIRED
```

Use the result set element to help you interpret the result of an external application query. There are three pieces of data that are of interest: the display name of the value (the display-name child element), the value's description (the description child element), and the literal entitlement value (the ent-value child element), which is not displayed.

The token elements `<token-src-dn>`, `<token-association>`, `<token-attr>` are actually placeholders for XPATH expressions that extract the src-dn attribute value, association value, or any attribute values respectively from an XDS-formatted XML document. The DTD assumes the query result is XDS.

Other Headings in the DTD

The remaining entitlement headings in the Entitlement DTD serve different functions, but they are not items that you need to focus on when creating an entitlement.

```
<!--
Entitlement reference stored in the DirXML-EntitlementRef attribute of a
DirXML-EntitlementRecipient or a DirXML-SharedProfile object.
-->
```

The information stored in the Entitlement Reference portion of the DTD points to an entitlement object. This information is placed there by the managing agent (such as the Role-Based Entitlement driver, `Entitlement.xml`, or the Approval Flow driver, `UserApplication.xml`). This is the triggering event for an action to take place in a connected system. You don't need to do anything with the DTD under this heading, but you can use this information to ensure that the entitlement object is being referenced.

```
<!--
    Entitlement result stored in the DirXML-EntitlementResult attribute of a
DirXML-EntitlementRecipient object.
-->
```

The Entitlement Result portion reports the results about whether an entitlement is granted or revoked. The information includes the state or status of the event and when the event is granted or revoked (through a time stamp). You don't need to do anything with the elements and attributes under this heading.

```
<!--
   Cached query results stored in the DirXML-SPCachedQuery attribute of a
DirXML-Entitlement object.
-->
```

The Entitlement Query portion contains the entitlement values that are gathered from an external application. This information can then be used again if the entitlement client needs to display this information. These values are stored in the `DirXML-SPCachedQuery` attribute of the Entitlement object. You don't need to do anything with the elements and attributes under this heading.

```
<!--
   Representation of a DirXML-EntitlementRef within the DirXML Script and
within the operation-data of an operation in an XDS document.
-->
```

Because the DTD defines values for more than one document, this EntitlementRef portion is actually not part of the Entitlement definition. You don't need to do anything with the elements and attributes under this heading.

6.4.4 Creating Entitlements Through Designer

Although the examples in Section 6.4.5, "Creating and Editing Entitlements in iManager," on page 188 show the actual XML code for writing entitlements, a much easier method of writing entitlements is to use the Designer utility that ships with Identity Manager. After you add an Identity Manager driver to an Identity Vault in the Designer modeler, you can right-click on the driver from the Outline View and select Add Entitlement. You are prompted through the Entitlement Wizard to designate the type of entitlement you want, and the wizard then steps you through the creation process.

For more information on using the Entitlement Wizard, see the Designer for Identity Manager: Administration Guide.

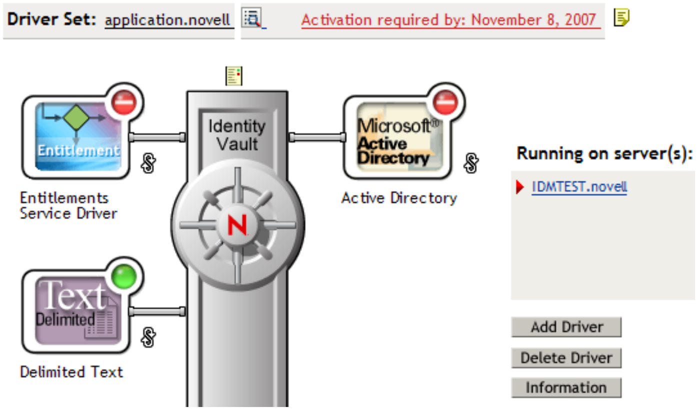
6.4.5 Creating and Editing Entitlements in iManager

Although it is recommended that you use Designer's Entitlement Wizard for creating entitlements, you can create entitlements through iManager.

1 Select *Identity Manager* > *Identity Manager Overview*.

2 Browse to the driver set where the driver resides, then click *Search*.

3 From the Identity Manager Overview page, click the Driver object.

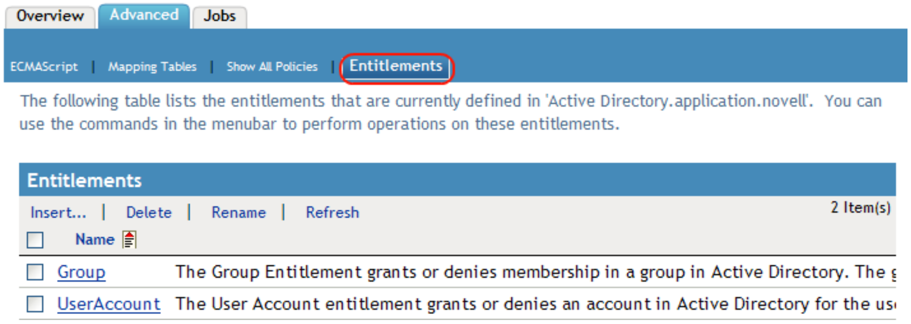


Driver Set: application.novell Activation required by: November 8, 2007

Running on server(s):

IDMTEST.novell

4 On the Driver Overview page click the *Advanced* tab, then click *Entitlements*.



Overview
Advanced
Jobs
ECMAScript | Mapping Tables | Show All Policies | Entitlements

The following table lists the entitlements that are currently defined in 'Active Directory.application.novell'. You can use the commands in the menubar to perform operations on these entitlements.

Entitlements

Insert... | Delete | Rename | Refresh

2 Item(s)

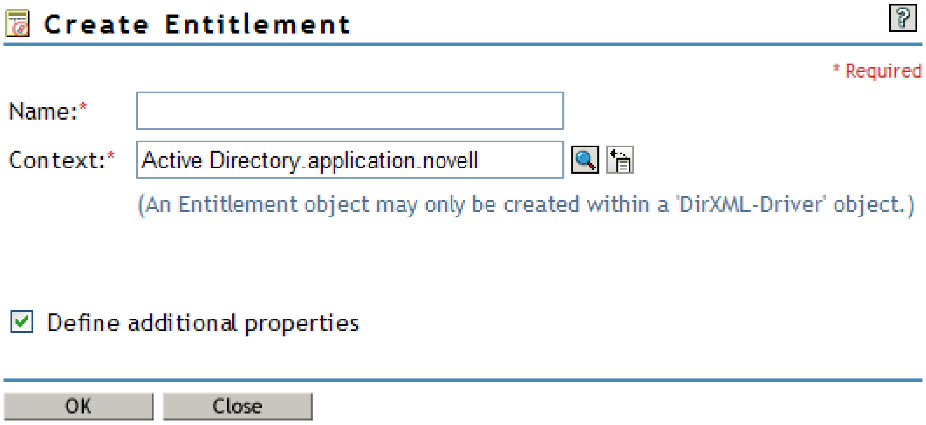
□	Name	
□	Group	The Group Entitlement grants or denies membership in a group in Active Directory. The g
□	UserAccount	The User Account entitlement grants or denies an account in Active Directory for the us

5 Click *Insert* to create an entitlement.

6 Specify a name for the entitlement.

NOTE: It is not a good idea to change an entitlement's name. If you change the entitlement name later on, you also need to change all of the references in the policies that are implementing the entitlement. The entitlement name is stored on the Ref and Result attributes within the policy.

The context for the entitlement is already populated, because the driver object is selected.



Create Entitlement

* Required

Name:*

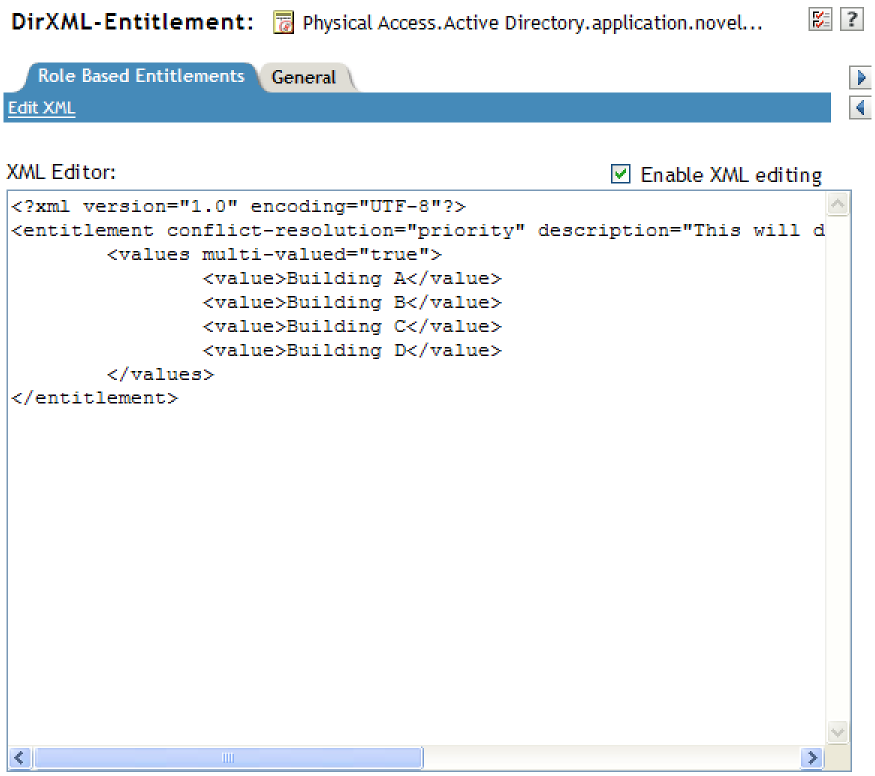
Context:* Active Directory.application.novell

(An Entitlement object may only be created within a 'DirXML-Driver' object.)

☑ Define additional properties

7 Verify *Define additional properties* is checked, then click *OK*.

8 Check *Enable XML editing*, then create your entitlement.

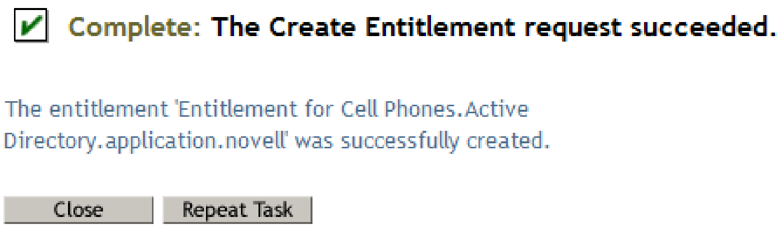


DirXML-Entitlement: Physical Access.Active Directory.application.novel...

XML Editor: ☑ Enable XML editing

```
<?xml version="1.0" encoding="UTF-8"?>
<entitlement conflict-resolution="priority" description="This will d
        <values multi-valued="true">
                <value>Building A</value>
                <value>Building B</value>
                <value>Building C</value>
                <value>Building D</value>
        </values>
</entitlement>
```

9 Read the information message about the creation of the entitlement, then click *Close*.



✓ Complete: The Create Entitlement request succeeded.

The entitlement 'Entitlement for Cell Phones.Active Directory.application.novell' was successfully created.

6.4.6 Example Entitlements To Help You Create Your Own Entitlements

There are two types of entitlements that you can create: value-less and valued. Valued entitlements can get their values from an external query, from an administrator-defined list, or free form. Below are examples of the four types of entitlements you can create.

NOTE: If you see a line that is without the Lesser Than sign (<), it means the line has wrapped and the information is usually displayed on one line, not two (or three). Also remember that, other than the Account Entitlement, these are simply examples of what you can create for each type of valued entitlement.

Example 1: Account Entitlement: Value-less

```
<?xml version="1.0" encoding="UTF-8"?>
<entitlement conflict-resolution="priority"
    description="This is an Account Entitlement"
    display-name="Account Entitlement"/>
```

In this example, the value-less entitlement's name is Account. This is followed by the conflict-resolution line with the default setting of Priority, which in most cases means that if the entitlement is used by Role-Based Entitlements, the RBE with priority sets the value. (However, because this is an example of a value-less entitlement, valued settings don't apply.) The Entitlement description is This is an Account Entitlement, and the display name is Account Entitlement. This information is all you need to create an Account Entitlement, which you can then use to grant an account in an application.

The Active Directory driver with entitlements enabled has a UserAccount entitlement that Active Directory uses to grant or revoke a user account.

```
<?xml version="1.0" encoding="UTF-8"?>
<entitlement conflict-resolution="union"
  description="The User Account entitlement grants or denies an
  account in Active  Directory for the user. When granted, the user
  is given an enabled logon account.  When revoked, the logon
  account is either disabled or deleted depending on how the drive
  is configured."
  display-name="User Account Entitlement" name="UserAccount">
</entitlement>
```

In this example, the conflict resolution is Union, which allows the entitlement to merge the values that are assigned. (Again, valued settings don't apply to value-less entitlements.) The Description field explains what this entitlement is used for and why it was created. This is useful information for those who perform future modifications to the entitlement. The actual name of the entitlement is UserAccount, while the `<display-name>` displays in a managing agent as User Account Entitlement.

Example 2: Application Query Entitlement: External Query

The Group and Exchange Mailbox entitlements that come with an entitlement-enabled Active Directory driver offer examples of application queries. Use this entitlement type when you need external information from a connected system to perform an event.

```
<?xml version="1.0" encoding="UTF-8"?>
<entitlement conflict-resolution="union"
   description="The Group Entitlement grants or denies membership in
   a group in Active Directory. The group must be associated with a
   group in the Identity Vault. When revoked, the user is removed from
   the group. The group membership entitlement is not enforced on the
   publisher channel: If a user is added to a controlled group in
   Active Directory by some external tool, the user is not removed by
   the driver. Further, if the entitlement is removed from the user
   object instead of being simply revoked, the driver takes no action."
   display-name="Group Membership Entitlement" name="Group">
   <values>
  <query-app>
     <query-xml>
    <nds dtd-version="2.0">
       <input>
```

```
      <query class-name="Group"
         scope="subtree">
         <search-class class-name="Group"/>
         <read-attr attr-name="Description"/>
      </query>
       </input>
    </nds>
     </query-xml>
     <result-set>
    <display-name>
        <token-src-dn/>
    </display-name>
    <description>
       <token-attr attr-name="Description"/>
    </description>
    <ent-value>
       <token-association/>
    </ent-value>
     </result-set>
  </query-app>
   </values>
</entitlement>
```

In this example, the Group entitlement uses Union to settle conflicts if the entitlement is applied more than once to the same object. The Union attribute merges the entitlements of all involved Role-Based Entitlement policies, so if one policy revokes an entitlement but another policy grants an entitlement, the entitlement is eventually granted.

The Group description is useful because of its detail, which explains what was set up through rules in the driver's policies. This description is a good example of the detail you need go into when defining entitlements in the first place.

The `<display-name>` is Group Membership Entitlement, which appears in the managing agents, such as iManager for Role-Based Entitlements. The name is the Relative Distinguished Name (RDN) of the entitlement. If you don't define a display name, the entitlement's name is its RDN.

The initial query values look for the class name of Group at the top of the tree and continues through its subtrees. These values come from the connected Active Directory server and the application query starts at the `<nds>` tag. Under the `<query-xml>` tag, this query receives information similar to the following:

```
<instance class-name="Group" src-dn="o=Blanston,cn=group1">
   <association>o=Blanston,cn=group1</association>
   <attr attr-name="Description"> the description for group1</attr>
</instance>
<instance class-name="Group" src-dn="o=Blanston,cn=group2">
   <association>o=Blanston,cn=group2</association>
   <attr attr-name="Description"> the description for group2</attr>
</instance>
<instance class-name="Group" src-dn="o=Blanston,cn=group3">
   <association>o=Blanston, cn=group3</association>
   <attr attr-name="Description"> the description for group3</attr>
</instance>
<!-- ... ->
```

Then, under the <result-set> tag, the information received from the query fills in the various fields. For instance, the <display-name> field would receive o=Blanston,cn=group1. The <description> field would receive the description for group1, and the <ent-value> field would receive o=Blanston,cn=group1. Because more than one group existed and met the query criteria, this information was also collected and shown as other instances.

NOTE: The association format value is unique for every external system, so the format and syntax are different for each external system queried.

Another example is the Exchange Mailbox entitlement.

```
<?xml version="1.0" encoding="UTF-8"?>
<entitlement conflict-resolution="union"
  description="The Exchange Mailbox Entitlement grants or denies an
  Exchange mailbox for the user in Microsoft Exchange."
  display-name="Exchange Mailbox Entitlement" name="ExchangeMailbox">
  <values>
  <query-app>
     <query-xml>
    <nds dtd-version="2.0">
       <input>
      <query class-name="msExchPrivateMDB"
         dest-dn="CN=Configuration," scope="subtree">
         <search-class class-name="msExchPrivateMDB"/>
         <read-attr attr-name="Description"/>
         <read-attr attr-name="CN"/>
      </query>
       </input>
    </nds>
      </query-xml>
      <result-set>
    <display-name>
       <token-attr attr-name="CN"/>
    </display-name>
    <description>
       <token-attr attr-name="Description"/>
    </description>
    <ent-value>
      <token-src-dn/>
    </ent-value>
      </result-set>
  </query-app>
   </values>
</entitlement>
```

In this example, the Exchange Mailbox entitlement uses Union to settle conflicts if the entitlement is applied more than once to the same object. The Union attribute merges the entitlements of all involved Role-Based Entitlement policies, so if one policy revokes an entitlement but another policy grants an entitlement, the entitlement is eventually granted.

The description states that the entitlement grants or revokes an Exchange mailbox for the user in Microsoft Exchange, which is enough detail for what the entitlement does. The display-name is Exchange Mailbox Entitlement, which appears in the managing agents, such as iManager for Role-Based Entitlements. The name is the Relative Distinguished Name (RDN) of the entitlement. If you don't define a display name, the entitlement's name is its RDN.

The initial query values look for the class name of msExchPrivateMDB, which is a Microsoft Exchange function call that begins looking in container Configuration and continues through its subtrees. These values come from the connected Active Directory database and the application query starts at the `<nds>` tag. The class msExchPrivateMDB has no equivalent in eDirectory, so you would need to be conversant in Microsoft Exchange function calls to make such a query. But the query is completed because of the rules and policies that are found in the Active Directory driver.

Entitlement consumers use the information that is retrieved by the query. For example, the entitlement value (ent-value) is passed to Identity Manager policies through the DirXML-EntitlementRef attribute. The display name and description information are displayed by iManager or the User Application and are stored in the DirXML-SPCachedQuery attribute.

Example 3: Administrator-Defined Entitlement: With Lists

The third example is an admin-defined entitlement that creates a grant or revoke event after you select a list entry.

```
<?xml version="1.0" encoding="UTF-8"?>
<entitlement conflict-resolution="union"
    description="This will show Administrator-defined Values">
     <display-name="Admin-defined Entitlement"/>
     <values multi-valued="true">
    <value>Building A</value>
    <value>Building B</value>
    <value>Building C</value>
    <value>Building D</value>
    <value>Building E</value>
    <value>Building F</value>
  </values>
</entitlement>
```

In this example, the entitlement name is Admin-defined, with a defined display name of Admin-defined Entitlement. (You only need to put in a display name if you want the display name to be different from the entitlement’s RDN.) The conflict-resolution line shows the setting of Union, which allows the entitlement to merge the values that are assigned.

The Entitlement description is *This will show Administrator-defined Values*. The multi-value attribute is set to true, which allows the entitlement to assign a value more than once. In this example, the values are corporate building letters: Building A through Building F. Then, through an entitlement client such as an iManager RBE task or through the User App, users or defined-task managers can specify the building information, which is then included in an external application, such as Novell eDirectory.

Example 4: Administrator-Defined Entitlements: Without Lists

The fourth example is an admin-defined entitlement that forces the administrator to type a value before the entitlement can grant or revoke an event. You can use this kind of entitlement if you do not have all of the information at the initial setup so you cannot create a task list.

```
<?xml version="1.0" encoding="UTF-8"?>
<entitlement conflict-resolution="priority"
    description="There will be no pre-defined list">
    <values multi-valued="false"/>
</entitlement>
```

In this example, the entitlement name is Admin-defined (no list), and it uses the entitlement name as the displayed name because there is no display name entry. The conflict resolution is again set to the default of Priority, which means that if the entitlement is used by Role-Based Entitlements, the RBE with priority sets the value. Through an entitlement client, such as an iManager RBE task or through the User App, you specify the building information, which is then included in an external application, such as eDirectory.

6.4.7 Completing the Creating Entitlements Steps

The entitlement creation examples have shown how to work through the first two steps for creating and using entitlements, as described in Section 6.2, "Creating Entitlements: Overview," on page 176. This includes Step 1, making a checklist of what you want to accomplish with entitlements, and Step 2, writing entitlements to address items in the checklist. Step 3, creating policies for the Identity Manager driver, is beyond the scope of this chapter. For information on creating and editing policies, see *Understanding Policies for Identity Manager 3.5.1* and the appropriate *Identity Manager Driver Guide* (http://www.novell.com/documentation/idm35drivers/index.html).

After you create entitlements (or use entitlements that come preconfigured with certain Identity Manager drivers), you now need to manage them, which is Step 4. Entitlements are managed by two packages or agents: through iManager as Role-Based Entitlement Policies or through the User Application in workflow-based provisioning. For entitlements used in workflow-based provisioning, see "Configuring Provisioning Request Definitions" in the *Identity Manager 3.5.1 User Application: Administration Guide*. The rest of this chapter focuses on Role-Based Entitlements.

6.5 Managing Role-Based Entitlements Overview

- Section 6.5.1, "How the Entitlement Service Driver Works," on page 196

Traditionally, entitlements on connected systems are administered on a per-driver basis, solely by creating and editing driver configuration policies such as the ones you create with Policy Builder. In this traditional distributed model, a different administrator often controls each Identity Manager driver and connected system, and the business policies that determine whether a user gets resources on that system are "hard-coded" separately in the driver configuration policies for each connected system driver.

The Role-Based Entitlement model fits an environment where one or a few administrators have authority to control the entitlement policies. This kind of administrator needs to understand Identity Manager in general but does not necessarily need much Identity Manager or XSLT or DirXML script expertise to use the Role-Based Entitlements interface.

Role-Based Entitlement policies allow you to automatically grant or revoke business resources if the criteria are met. Entitlements are like a permission slip to access a resource. With the permission slip, you have access to the designated resource and without such a permission slip, you have no access. As a working example, you can specify that if user meets criteria 1, 2, and 3, then through a Role-Based Entitlement policy, the user becomes a member of Group H; but if the user meets criteria 4 and 5, he or she becomes a member of Group I.

Setting up to manage Role-Based Entitlements is a three step process:

1. If you haven't already done so, enable the DirXML-EntitlementRef attribute on the Identity Manager driver object as described in Section 6.2.2, "Enabling Entitlements on Other Identity Manager Drivers," on page 178.
2. Install the Entitlements Service driver (`Entitlement.xml`) as described in Section 6.6, "Creating an Entitlements Service Driver Object," on page 197.
3. Create Role-Based Entitlement Policies in iManager, as described in Section 6.7, "Creating Entitlement Policies," on page 199.

6.5.1 How the Entitlement Service Driver Works

Role-Based Entitlements relies on the Entitlements Service driver (`Entitlement.xml`). This driver is an engine service that monitors whether users have membership in an Entitlement Policy. If a user meets the dynamic membership criteria of an Entitlement Policy dynamic group, or is statically included, the Entitlements Service driver updates information in the DirXML-EntitlementRef attribute on the User object.

For the systems listed in Section 6.2.1, "Identity Manager Drivers with Configurations that Support Entitlements," on page 177, you can enable entitlements when importing the Identity Manager driver configuration. Identity Manager comes with a number of drivers with configurations that already contain entitlements, policies to implement the entitlements, and the driver enabled to listen for entitlement activities. You can then review the policies provided. These policies support entitlements by monitoring the DirXML-EntitlementRef attribute and granting or revoking entitlements.

The Entitlements Service driver updates the DirXML-EntitlementRef attribute only when one of the following happens:

- You use the Reevaluate Membership task
- You specify in which part of the tree users should be reevaluated
- A user is moved
- A user is renamed
- Any attribute used for membership in an Entitlement Policy is modified

Entitlement policies enable you to grant entitlements on connected systems and rights in Identity Vault. Entitlements on connected systems can be any of the following:

- Accounts
- Membership in e-mail distribution lists
- Group membership
- Attributes for the corresponding objects in connected systems, populated with values you specify
- Placement
- Other entitlements that you customize

Some of the options that you can create with entitlements are demonstrated in the driver configurations that have entitlements enabled.

Because one Entitlements Service driver is used per driver set, an Entitlement policy can manage only users that are in a read/write or master replica on the server that is associated with that driver set.

Role-Based Entitlement policies functionality is based on Identity Manager. Therefore, to administer connected systems, you must have Identity Manager drivers installed and configured properly and Identity Manager plug-ins installed.

In addition, to avoid possible conflicts between Entitlement Policy assignments and Identity Manager driver configurations, you should be aware of your business policies and how they are administered through Identity Manager. Identity Manager Entitlement policies and policies in a driver configuration should not overlap or conflict while they manage an attribute.

6.6 Creating an Entitlements Service Driver Object

Before you can create Entitlement policies, you need an Entitlements Service Driver object. You must create one for each driver set.

If you don't have an object, you are prompted to create one when you click the Role-Based Entitlements role and task.

1 Find out whether you already have an Entitlements Service driver.

   In iManager, click *Role-Based Entitlements* > *Role-Based Entitlements*, then select the driver set and click *OK*.

   - If the No Entitlements Service Driver page appears, continue with Step 2 to create an Entitlements Service Driver object.
   - If a Role-Based Entitlements page appears with a list of Entitlement Policies, you already have an Entitlements Service Driver object. You don't need to complete this procedure. Continue with Section 6.7, "Creating Entitlement Policies," on page 199.

2 In the No Entitlements Service Driver page, click *Yes*.

   The Create Driver Wizard opens.

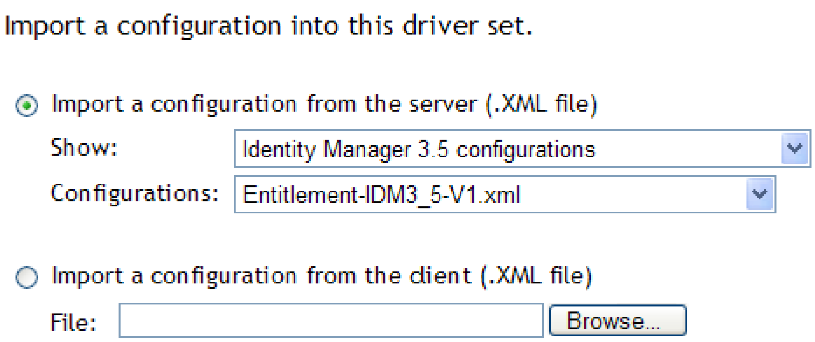
   You can also click *Identity Manager Utilities* > *Import Configurations*.

3 In the Create Driver Wizard page, select *In an Existing Driver Set*, then click *Next*.

4 In the *Import a configuration from the server (.XML file)* drop-down list, select how you want the driver configurations sorted:

   - All configurations
   - Identity Manager 3.5 configurations
   - Identity Manager 3.0 configurations
   - Configurations not associated with an IDM version

5 Select *Entitlement-IDM3_5-V1.xml*, then click *Next*.



Import a configuration into this driver set.

● Import a configuration from the server (.XML file)

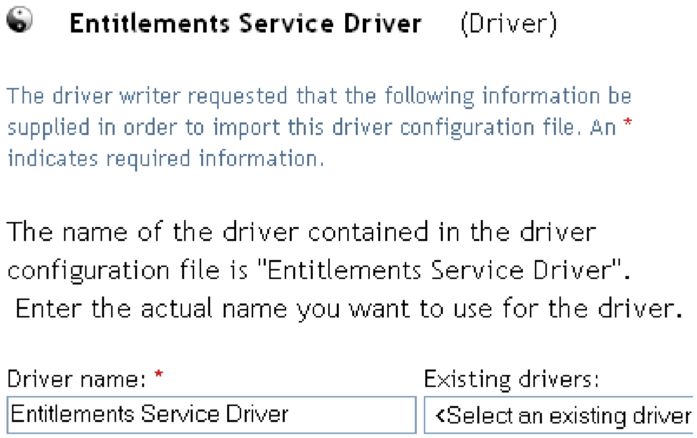
Show: Identity Manager 3.5 configurations

Configurations: Entitlement-IDM3_5-V1.xml

○ Import a configuration from the client (.XML file)

File:

6 Specify the name for Entitlements Service Driver object (or accept the default name), then click *Next*.



Entitlements Service Driver (Driver)

The driver writer requested that the following information be supplied in order to import this driver configuration file. An * indicates required information.

The name of the driver contained in the driver configuration file is "Entitlements Service Driver". Enter the actual name you want to use for the driver.

Driver name: * Entitlements Service Driver

Existing drivers: <Select an existing driver

The correct driver configuration file is chosen automatically.

7 Novell recommend that you define security equivalences and exclude administrative roles. Add user Admin to both of these selections, then click *Next*.

8 Review the summary, then click *Finish*.

The driver shim for the Entitlements Driver is installed by default when you install Identity Manager. The Entitlements Driver configuration file is installed by default when you install the Identity Manager plug-ins on your iManager server.

After completing the Wizard, you can access the plug-ins for Entitlements and begin creating Role-Based Entitlement Policies for this driver set.

IMPORTANT: If the driver set that hosts the Entitlement Services driver is assigned to more than one server, the Entitlement Services driver must be enabled on only one of those servers at a time. No other configuration is supported.

Although you can add more than one server to the driver set containing the Entitlement Services driver in iManager, the Role-Based Entitlements plug-in in iManager checks to see if the driver set is assigned to multiple servers and displays a configuration error message if it is. Even though other methods (LDAP calls, etc.) won't give you such configuration error messages, the only supported configuration is to associate the Entitlements Services driver to one server.

6.7 Creating Entitlement Policies

- Section 6.7.1, “Defining Membership for an Entitlement Policy,” on page 202
- Section 6.7.2, “Choosing Entitlements for an Entitlement Policy,” on page 204

To create an Entitlement Policy, you can use the wizard provided.

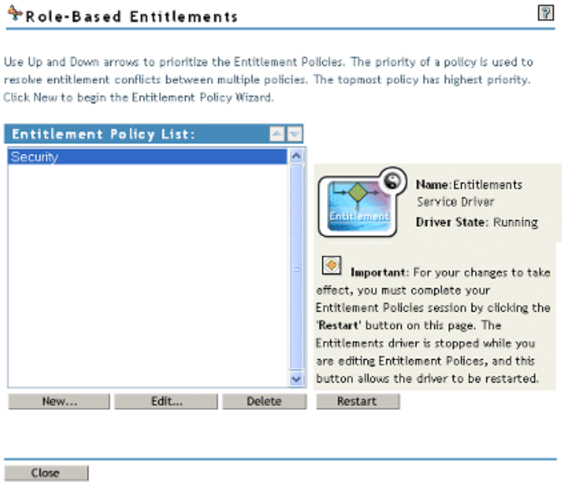
1 Make sure you have set up the Entitlements Service Driver and created the driver configurations that are necessary.

2 In iManager, click *Role-Based Entitlements* > *Role-Based Entitlements*.

3 Select a driver set.

Entitlement policies are per driver set.

The list of existing Entitlement policies opens, similar to the page in the following figure. If you are using Role-Based Entitlements for the first time, no policies are listed.



Role-Based Entitlements

Use Up and Down arrows to prioritize the Entitlement Policies. The priority of a policy is used to resolve entitlement conflicts between multiple policies. The topmost policy has highest priority. Click New to begin the Entitlement Policy Wizard.

Entitlement Policy List:

Security

Name: Entitlements Service Driver

Driver State: Running

Important: For your changes to take effect, you must complete your Entitlement Policies session by clicking the '**Restart**' button on this page. The Entitlements driver is stopped while you are editing Entitlement Polices, and this button allows the driver to be restarted.

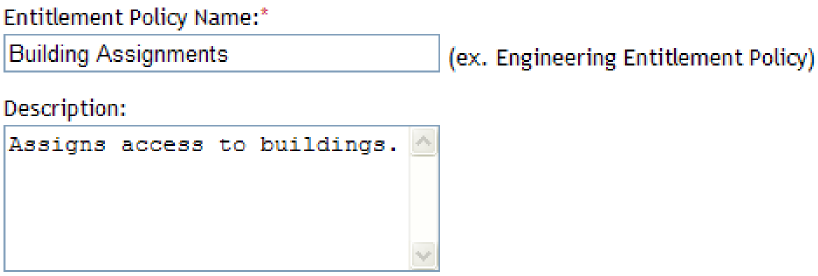
4 Click *New*.

The *Entitlement Policy* Wizard opens.

NOTE: Creating a new entitlement policy stops the Entitlements Service driver. You need to click *Restart* when you are finished creating policies.

5 Follow Step 5a through Step 5f in the wizard to create a new policy. Refer to the online help for information about each step in the wizard.

5a Specify a name and description for the policy, then click *Next*.



Entitlement Policy Name:*

Building Assignments (ex. Engineering Entitlement Policy)

Description:

Assigns access to buildings.

5b Define the dynamic membership filter by defining the search parameters.

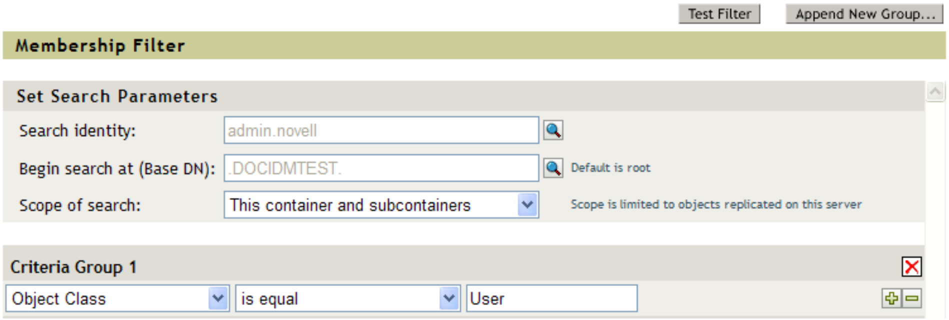
5b1 Specify a uses that has rights to perform the search.

5b2 Specify the location where to start the search.

5b3 Specify the scope of the search.

5b4 Define the filter criteria, then click *Next*.

The criteria determine which users are members of the entitlement policy.



Membership Filter

Set Search Parameters

Search identity: admin.novell

Begin search at (Base DN): .DOCIDMTEST.

Default is root

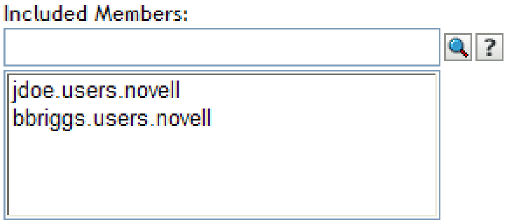
Scope of search: This container and subcontainers

Scope is limited to objects replicated on this server

Criteria Group 1

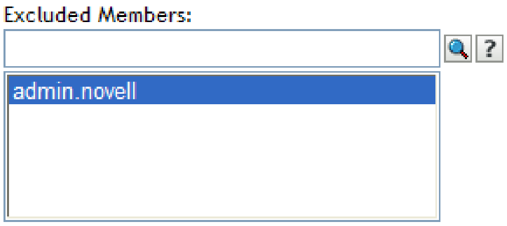
Object Class | is equal | User

5c Define static members by including and excluding members in the search criteria, then click *Next*.



Included Members:

jdoe.users.novell
bbriggs.users.novell



Excluded Members:

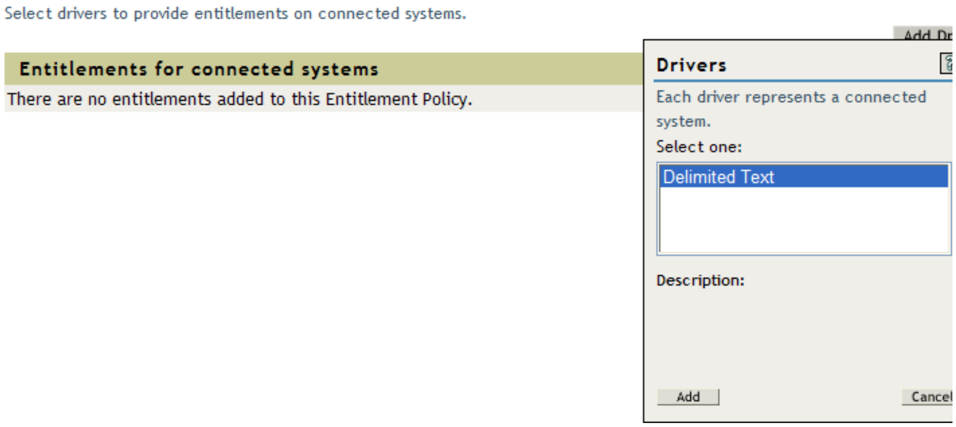
admin.novell

5d Define the entitlement and value for the policy.

5d1 Click *Add Driver* to select an Identity Manager driver and entitlements for inclusion.

You created entitlements in Section 6.4, "Writing Entitlements in XML through iManager," on page 180.

5d2 Browse to and select the driver that has an entitlement, then click *Add*.


Select drivers to provide entitlements on connected systems.

Entitlements for connected systems

There are no entitlements added to this Entitlement Policy.

Drivers

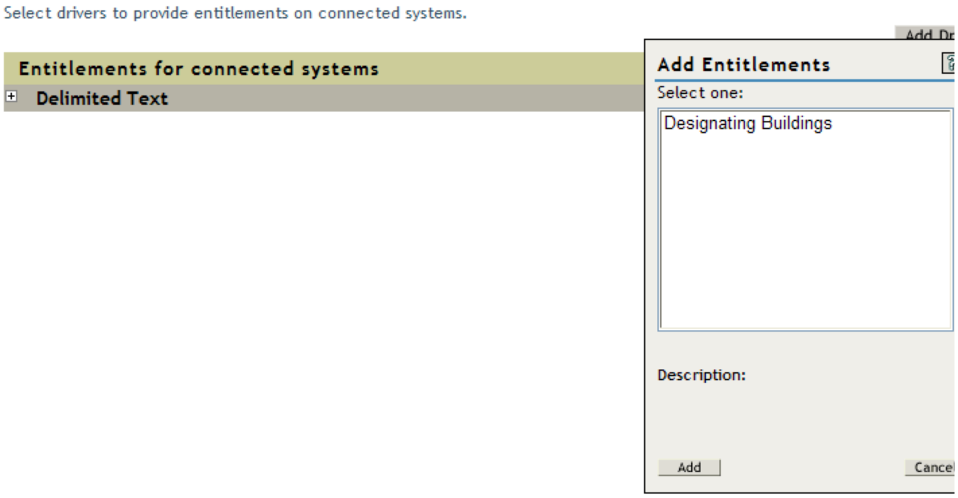
Each driver represents a connected system.

Select one:

Delimited Text

Description:

5d3 Select the entitlement, then click *Add*.


Select drivers to provide entitlements on connected systems.

Entitlements for connected systems

Delimited Text

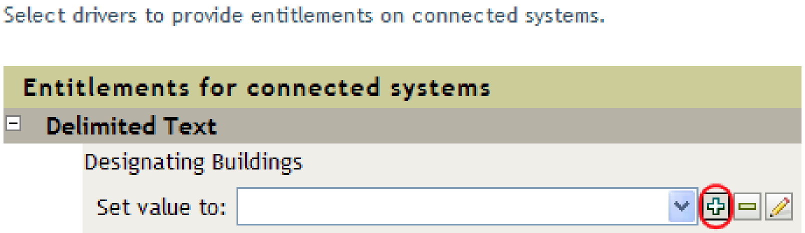
Add Entitlements

Select one:

Designating Buildings

Description:

5d4 Click the plus icon to define the value for the entitlement.


Select drivers to provide entitlements on connected systems.

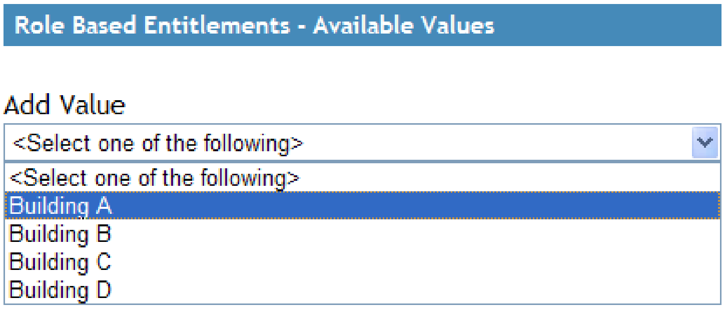
Entitlements for connected systems

Delimited Text

Designating Buildings

Set value to:

5d5 Select the desired value, then click *OK*.



Role Based Entitlements - Available Values

Add Value

<Select one of the following>

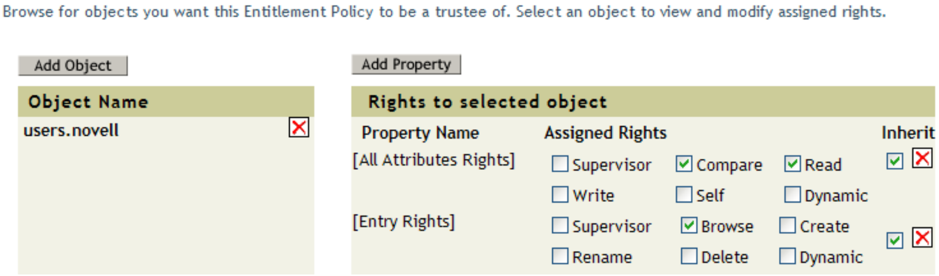
- <Select one of the following>
- Building A
- Building B
- Building C
- Building D

5d6 Click *Next*.

5e Browse for objects for which you want this entitlement policy to be a trustee.

5e1 Click *Add Object* to browse for the object, then click *OK*.

5e2 Select the object, then click *Add Property*.



Browse for objects you want this Entitlement Policy to be a trustee of. Select an object to view and modify assigned rights.

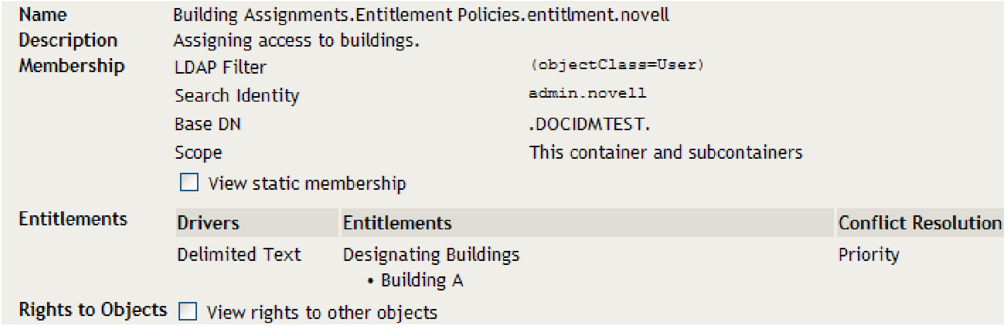
Object Name	
users.novell	☒

Rights to selected object

Property Name	Assigned Rights			Inherit	
[All Attributes Rights]	□ Supervisor	☑ Compare	☑ Read	☑	☒
	□ Write	□ Self	□ Dynamic		
[Entry Rights]	□ Supervisor	☑ Browse	□ Create	☑	☒
	□ Rename	□ Delete	□ Dynamic		

5e3 Define the desired properties, then click *Next*.

5f Read the summary to ensure that the entitlement policy does what you want it to. If it does, click *Finish*; if not, click *Back*.



Name	Building Assignments.Entitlement Policies.entitlment.novell		
Description	Assigning access to buildings.		
Membership	LDAP Filter	(objectClass=User)	
	Search Identity	admin.novell	
	Base DN	.DOCIDMTEST.	
	Scope	This container and subcontainers	
	□ View static membership		
Entitlements	Drivers	Entitlements	Conflict Resolution
	Delimited Text	Designating Buildings • Building A	Priority
Rights to Objects	□ View rights to other objects		

6 Click *Restart* to complete the session.

Entitlement policy creation turns off the Entitlements Service driver.

6.7.1 Defining Membership for an Entitlement Policy

Like an Identity Manager driver, each Entitlement policy can manage only objects that are in a master or read/write replica on the server to which is it assigned. Each Entitlement policy is associated with a single Driver Set object, which is assigned to a particular server.

Only User objects (and other object types derived from the class of User) can be members of an Entitlement policy. To get to the Membership page in an Entitlement policy, select *Role-Based Entitlements* > *Role-Based Entitlements*, then highlight the Entitlement policy you want to edit from the Entitlement Policy List and select *Edit*. In the Internet Explorer browser, select the *Membership* tab; in the Firefox browser, select Edit Dynamic Members from the pull-down menu.

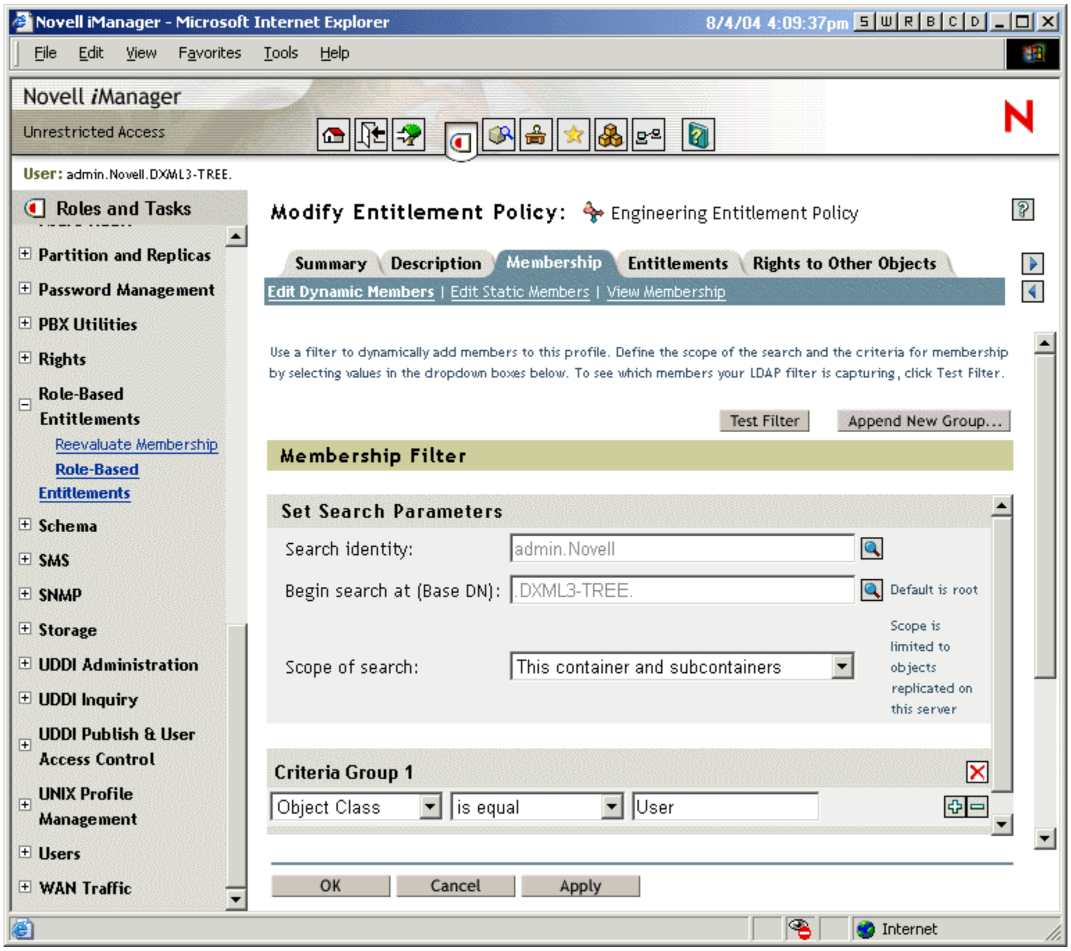
An Entitlement policy is a dynamic group object. You can define membership for an Entitlement policy by using two methods, dynamic and static. You can use both methods in the same Entitlement policy.

- **Dynamic:** You can define criteria for membership based on values of attributes of the object, such as whether the job title includes the word “Manager.” The criteria you specify are converted into an LDAP filter.

  Users who meet the criteria are automatically part of the Entitlement policy, without requiring you to specifically add each user to the policy. The dynamic membership is the same as a Dynamic Group object.

  If an object changes so that it no longer meets the criteria for dynamic membership, the entitlements are automatically revoked.

Figure 6-2 *Editing Dynamic and Static Members*



Modify Entitlement Policy: Engineering Entitlement Policy

Use a filter to dynamically add members to this profile. Define the scope of the search and the criteria for membership by selecting values in the dropdown boxes below. To see which members your LDAP filter is capturing, click Test Filter.

Membership Filter

Set Search Parameters

Search identity: admin.Novell

Begin search at (Base DN): .DXML3-TREE.

Default is root

Scope of search: This container and subcontainers

Scope is limited to objects replicated on this server

Criteria Group 1

Object Class | is equal | User

- **Static:** In addition to creating criteria for dynamic membership (an LDAP filter), you can include or exclude specific users.

  You can add statically members who don't meet the criteria of the filter. You can exclude members who meet the filter's criteria but should not be included in the Entitlement Policy.

NOTE: If you run the *Role-Based Entitlements* > *Reevaluate Membership* option and the Entitlement Services driver is stopped, you must first restart the driver before the reevaluation process can begin.

6.7.2 Choosing Entitlements for an Entitlement Policy

- "Accounts on Connected Systems" on page 204
- "Membership in E-Mail Distribution Lists and NOS Lists" on page 205
- "Attribute Values on Connected Systems" on page 206

Entitlements enable you to grant or revoke access to services on connected systems and rights in Identity Vault.

Drivers that you install with entitlements enabled come with a list of entitlements that can be assigned using an Entitlement policy. You can create your own entitlements that can be used in an Entitlement policy. The entitlements that the driver can provide are child objects of the driver, which is created by the driver developer to represent the capability of the driver and connected system.

Trustee rights to objects in the Identity Vault are immediately granted to members of the Entitlement policy. By default, entitlements in connected systems are granted to each member of the Entitlement policy the next time an attribute used for Entitlement policy membership is modified for that user, or when a user is moved to a different container or renamed.

Entitlements on connected systems can be any of the following:

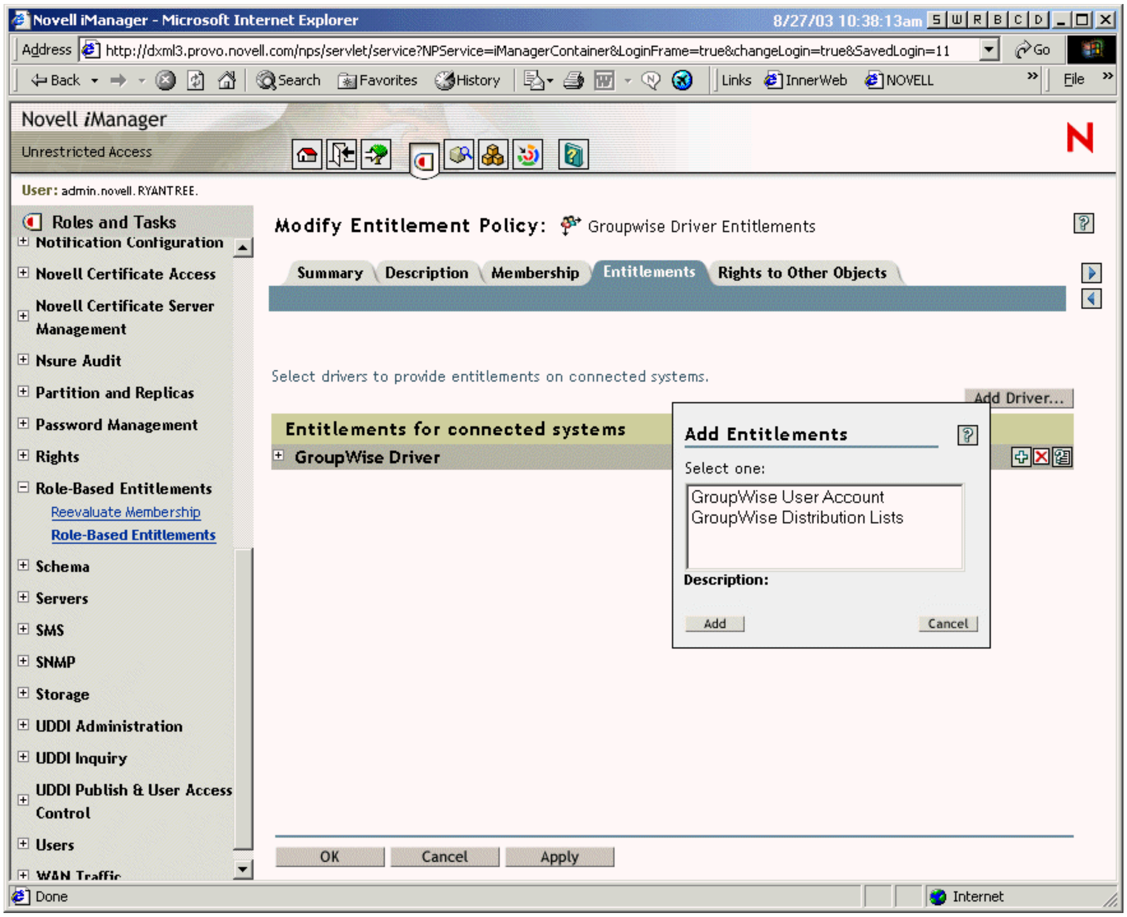
- Accounts
- Membership in e-mail distribution lists
- Group membership in NOS lists
- Attributes for the corresponding objects in connected systems, populated with values you specify
- Other entitlements that you customize

Accounts on Connected Systems

To add entitlements to an Entitlement Policy, go to the Entitlements page and select a driver. A pop-up window displays the entitlements that the driver offers.

For example, in the following figure, you can see two kinds of entitlements being offered by a GroupWise driver, and the first one in the list is a GroupWise User Account.

Figure 6-3 *Interface for Defining Entitlements*



Novell iManager - Microsoft Internet Explorer

Address http://dxml3.provo.novell.com/nps/servlet/service?NPService=iManagerContainer&LoginFrame=true&changeLogin=true&SavedLogin=11

User: admin.novell.RYANTREE.

Modify Entitlement Policy: Groupwise Driver Entitlements

Select drivers to provide entitlements on connected systems.

Entitlements for connected systems

GroupWise Driver

Add Entitlements

Select one:

GroupWise User Account
GroupWise Distribution Lists

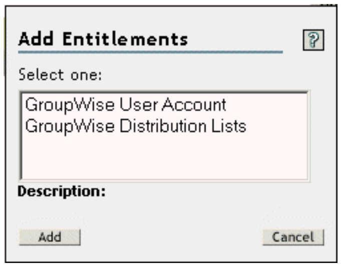
Description:

Membership in E-Mail Distribution Lists and NOS Lists

To assign membership in groups on connected systems, you choose the membership entitlement from the list of entitlements offered by a driver.

The following figure shows an example, with GroupWise Distribution Lists shown second in the list.

Figure 6-4 *Selecting GroupWise Distribution Lists*



Add Entitlements

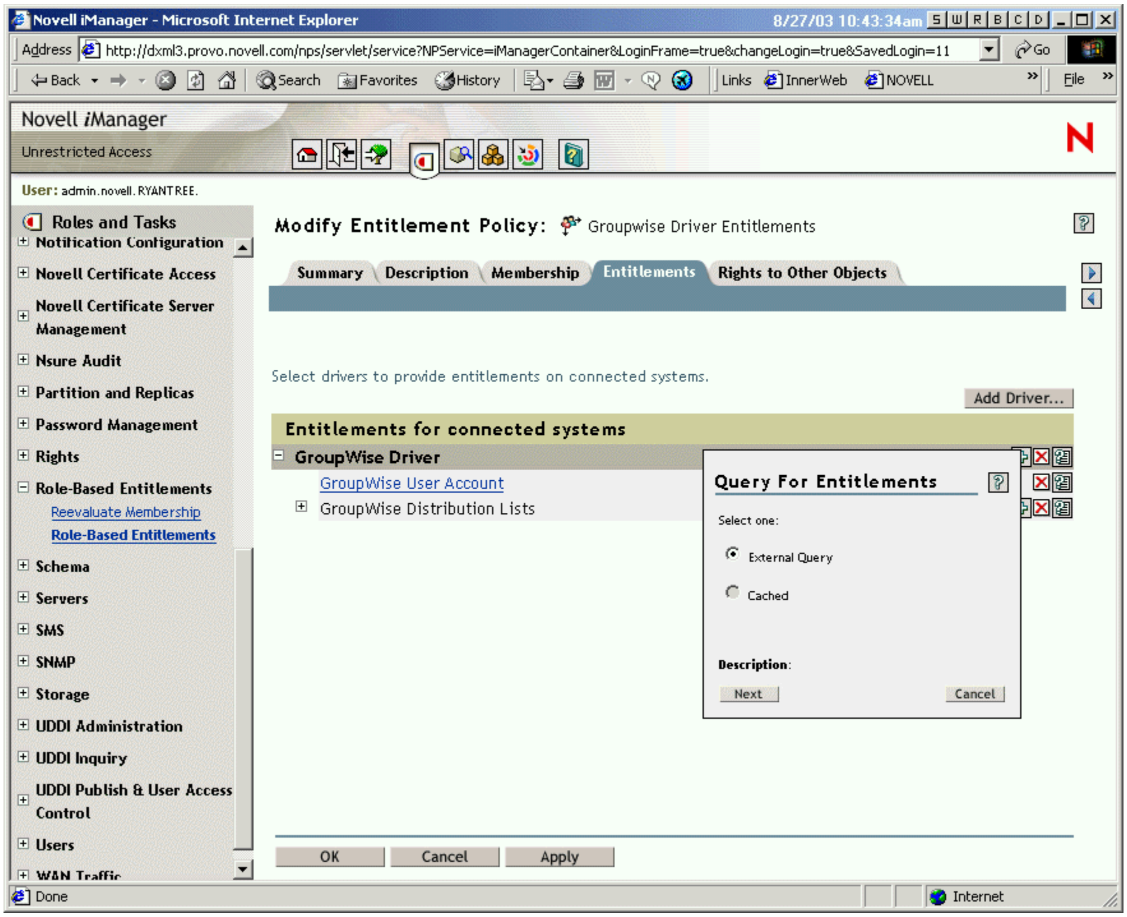
Select one:

GroupWise User Account
GroupWise Distribution Lists

Description:

If you choose *GroupWise Distribution Lists* in this example, a query pop-up is displayed, like the example in the following figure.

Figure 6-5 Query for Entitlements



Select drivers to provide entitlements on connected systems.

Select one:

● External Query

○ Cached

The Entitlement Policy interface lets you query for the list of e-mail distribution lists or NOS lists. After a query has been performed, you can choose to view the cached list.

The drivers are configured to return the complete list, so you can choose from the lists that exist on the connected system.

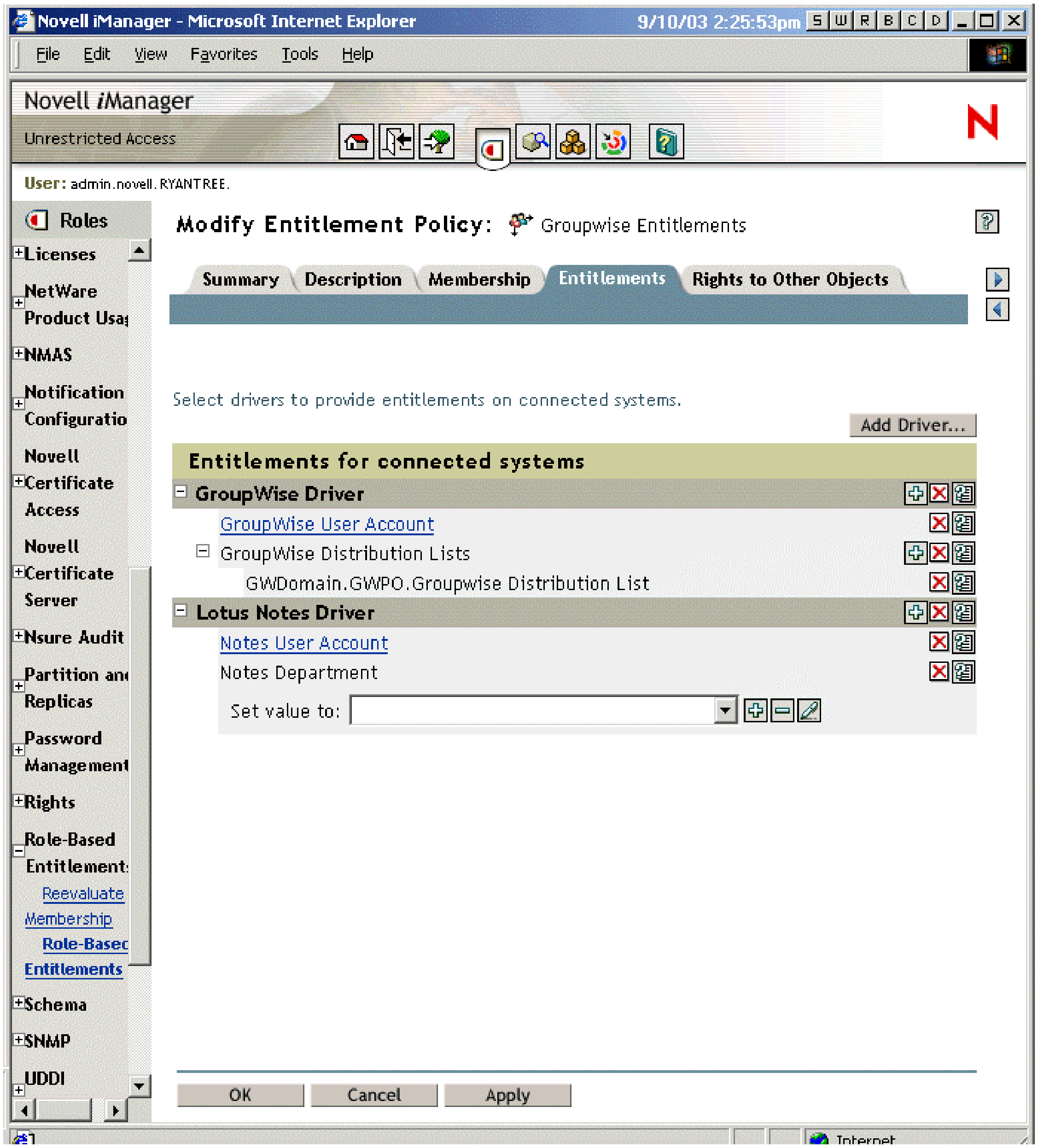
NOTE: A driver could be customized to limit the list to group names you specify, rather than a query that returns the complete list.

Attribute Values on Connected Systems

You can assign attribute values for user accounts on connected systems. The interface lets you type in the value you want the user accounts to have.

The following figure shows an example of adding an attribute value for a Notes attribute, Department.

Figure 6-6 *Adding an Attribute Value*



Modify Entitlement Policy: Groupwise Entitlements

Select drivers to provide entitlements on connected systems.

Entitlements for connected systems

- GroupWise Driver
  - GroupWise User Account
  - GroupWise Distribution Lists
    - GWDomain.GWPO.Groupwise Distribution List
- Lotus Notes Driver
  - Notes User Account
  - Notes Department
    - Set value to:

6.8 Conflict Resolution between Role-Based Entitlement Policies

- Section 6.8.1, "Conflict Overview," on page 208
- Section 6.8.2, "Changing the Conflict Resolution Method for an Individual Entitlement," on page 209
- Section 6.8.3, "Prioritizing Entitlement Policies," on page 210

6.8.1 Conflict Overview

When you are creating Entitlement Policies, it's possible that the policies that affect a particular user might conflict in assigning entitlements to that user.

Here's how those conflicts are resolved. For some entitlements, you can change the conflict resolution.

- **Entitlements that don't have values are additive.** In most cases an Account entitlement doesn't have values. If a user is granted an account on a connected system by any Entitlement Policy, the user receives an account on that system. It does not matter whether another Entitlement Policy conflicts; the result is additive.

  This is always true; the method of conflict resolution for granting accounts cannot be changed.

  One metaphor for entitlements that don't have values is a light switch; it's either on or off; granted or not granted.

  For example, if the Manager Entitlement Policy grants Jean Chandler an Exchange account, but Jean Chandler is excluded from the Mail Room Employees Entitlement Policy that also grants Exchange accounts, Jean still gets an Exchange account.

- **Entitlements that have values are additive by default, but you can choose to resolve by priority.** Entitlements, such as group membership, have a list of group names for the values, or an attribute with a value. By default, these kinds of entitlements are also additive.

  You can change the conflict resolution for these kinds of entitlements, if desired.

  - **conflict-resolution="union" —** A value of "union" means the entitlements are additive. A user is granted all the entitlements that he or she is assigned by membership in any policy. The differing entitlement values are simply added together and the user gets them all.

    For example, if Jameel is a member of the Trade Show Contractors Policy that grants membership in a GroupWise e-mail distribution list named Trade Show Mailing List, and he is excluded from membership in the Trade Show Managers Policy that also assigns the e-mail distribution list named Trade Show Mailing List, he still receives membership in the e-mail distribution list.

    As another example, if Consuela is granted membership in the Active Directory group named Mailroom Staff by the Mailroom policy, and also granted membership in the Active Directory group named Emergency Response by the Emergency Volunteers policy, she is granted membership in both groups in Active Directory.

    With this setting, the order of an Entitlement policy in the list of policies is not important for the entitlement.

  - **conflict-resolution="priority" —** A value of "priority" means that if the values in two different policies conflict, or if one policy includes the user and another excludes the user, the entitlements granted to the user are only those in the Entitlement policy that is listed higher in the list of Entitlement policies.

    The previous examples would have a different result with this setting.

    In the example above for Jameel, if the GroupWise e-mail distribution list entitlement had a value of "priority," and the Trade Show Managers Policy was higher in the list than the Trade Show Contractors Policy, Jameel would not be granted membership in the Trade Show Mailing List.

In the example above for Consuela, if the Active Directory NOS group membership entitlement had a value of “priority,” and the Mailroom Policy was higher in the list than the Emergency Volunteers Policy, Consuela would be granted membership only in the Mailroom Staff group. She would not be granted membership in the Emergency Response group because the conflict resolution is by priority, not additive.

This functionality is useful if, for example, you configure your environment to use Role-Based Entitlements to place users in a hierarchical structure on another system. You would want the user to be placed in either one place or another, not in two places at the same time.

Keep in mind that the setting is independent for each entitlement offered by each driver.

As a general rule, if you use the “priority” setting, you should place administrator or manager policies higher in the list than policies for end users or individual contributors. You should put groups with narrower membership higher than groups with broader membership.

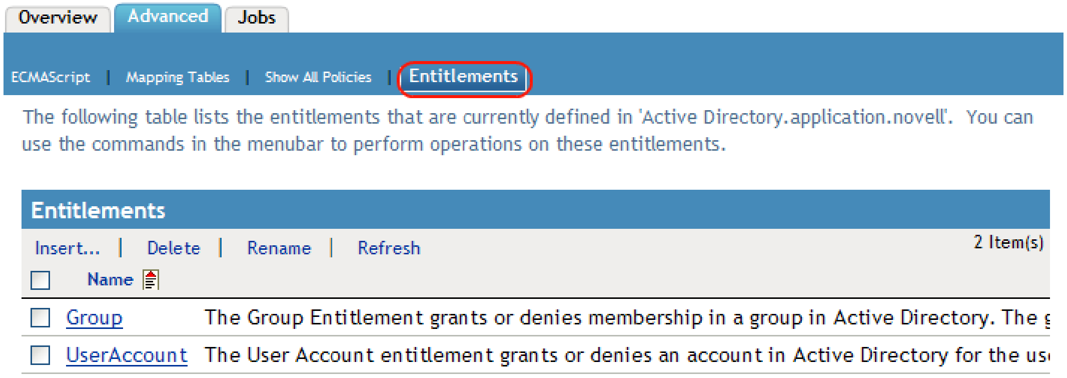
6.8.2 Changing the Conflict Resolution Method for an Individual Entitlement

1 In iManager, click *Identity Manager > Identity Manager Overview*, then select a driver set.

2 Click the Driver status button, then select *Stop driver.*

3 Click the driver icon for the driver that offers the entitlement you want to change.

4 On the Driver Overview page click the *Advanced* tab, then click *Entitlements*.



The following table lists the entitlements that are currently defined in 'Active Directory.application.novell'. You can use the commands in the menubar to perform operations on these entitlements.

Entitlements

☐	Name	
☐	Group	The Group Entitlement grants or denies membership in a group in Active Directory. The g
☐	UserAccount	The User Account entitlement grants or denies an account in Active Directory for the us

5 Click on the entitlement name to edit the entitlement in the XML viewer.

6 Select the check box for *Enable XML editing*.

7 In the XML, find the definition of the entitlement you want to change.

Here's an example of the line you should look for:

```
<entitlement conflict-resolution="union" description="Grants membership to
GroupWise Distribution lists" display-name="GroupWise Distribution Lists"
name="gwDistLists">
```

8 Change the conflict-resolution value. The two possible values are the following:

```
conflict-resolution="union"
```

```
conflict-resolution="priority"
```

For information about these values, see "Conflict Resolution between Role-Based Entitlement Policies" on page 207.

9 Click *OK* to save the changes.

10 Click the *Overview* tab to access the driver icon.

11 Click *Restart* to restart the driver.

12 Click *Identity Manager Overview*, to browse to and restart the Entitlement Services driver.

6.8.3 Prioritizing Entitlement Policies

By default, the order of the list of Entitlement Policies does not matter. This is because the driver configurations shipped with Identity Manager have `conflict-resolution="union"` as the method of conflict resolution for each entitlement.

If you change any of the entitlements to `conflict-resolution="priority,"` then the order of the list of Entitlement Policies matters, but only for those entitlements you changed. For information about these values, see "Conflict Resolution between Role-Based Entitlement Policies" on page 207.

You change the order of the Entitlement policies by using the arrow buttons next to the list of Entitlement Policies. The policy first in the list is the highest priority.

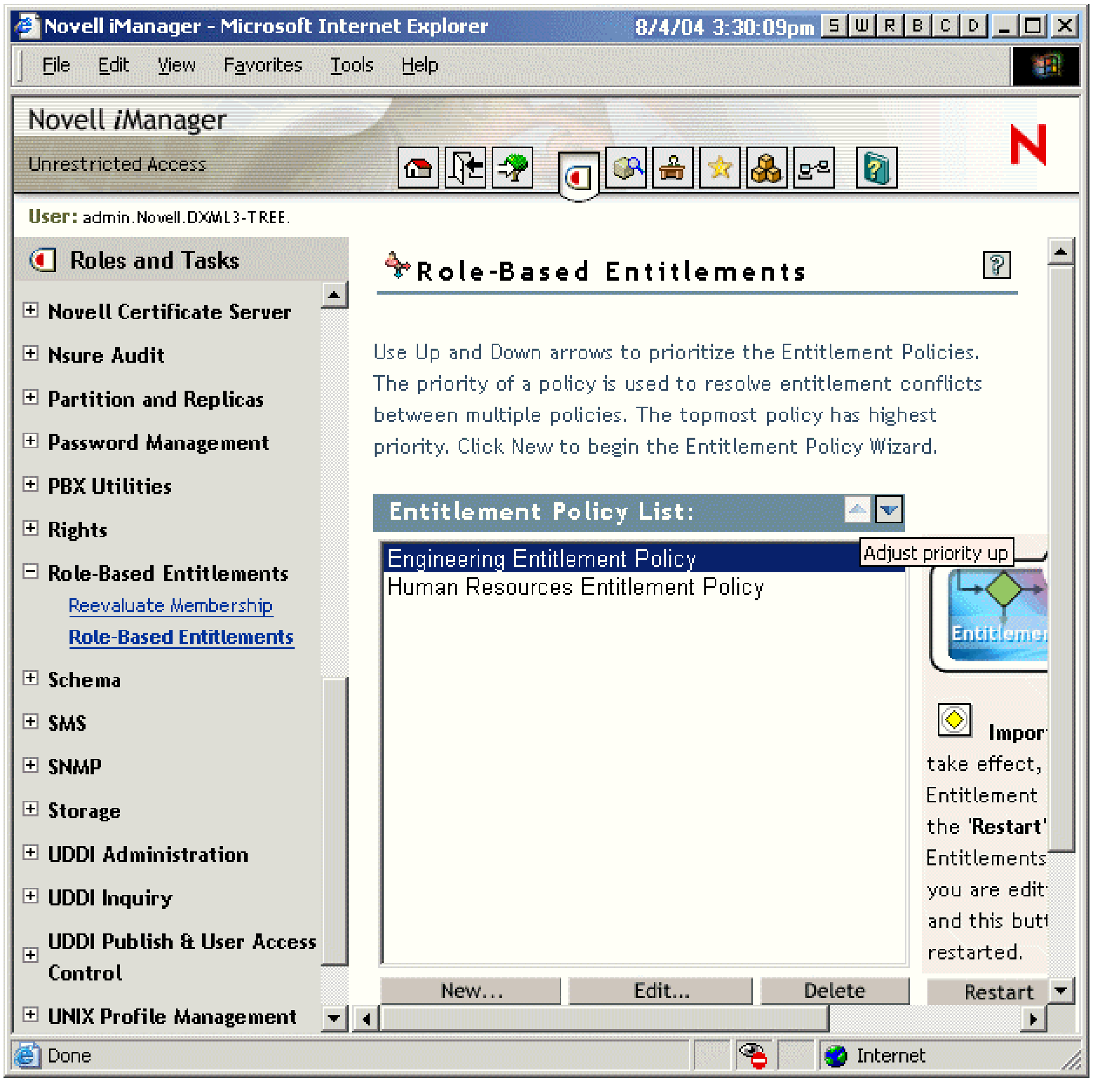
1 In iManager, click *Role-Based Entitlements* > *Role-Based Entitlements*.

2 Search for and select a driver set.

A page appears with a list of the Entitlement Policies.

3 Change the priority of the Entitlement policies by using the arrow buttons to move the policies up and down in the list.

Moving an Entitlement policy higher in the list gives it a higher priority.



Use Up and Down arrows to prioritize the Entitlement Policies. The priority of a policy is used to resolve entitlement conflicts between multiple policies. The topmost policy has highest priority. Click New to begin the Entitlement Policy Wizard.

Engineering Entitlement Policy
Human Resources Entitlement Policy

Impor
take effect,
Entitlement
the 'Restart'
Entitlements
you are edit
and this butt
restarted.

4 Click *Close* to restart the driver.

Changes in priority don't take effect until the driver is restarted.

6.9 Troubleshooting Role-Based Entitlements

When troubleshooting, keep in mind these issues:

- When you make any changes to policies by clicking *New*, *Edit*, or *Remove* on the page where the policies are listed, the *Entitlements Service Driver* is stopped. The driver is not restarted unless you click *Restart* on that page.

  This feature prevents the driver from granting or revoking entitlements in your production environment while your changes to policies are incomplete.

- Similarly, the Entitlements Service Driver won't start if more than one person appears to be editing Entitlement Policies at the same time.
- Because one Entitlements Service Driver is used per driver set, an Entitlement policy can manage only users that are in a read/write or master replica on the server that is associated with that driver set.

6.10 Entitlement Elements that Apply To Role-Based Entitlements and Workflow-Based Provisioning Entitlements

The information below applies to all entitlements rather than to a specific implementation.

6.10.1 Controlling the Meaning of Granting or Revoking Entitlements

You can control the consequences of granting or revoking an entitlement. Each driver provides a list of supported choices that control the meaning of "grant" or "revoke."

For example, when adding a GroupWise account, you could specify that grant actually means to grant the user an account in a disabled state, so that the administrator must intervene before the user can access the account. Or, you could choose to enable the account, which is the default.

By default, the driver configurations use the option that is most likely to preserve data. For example, the default meaning of remove for a GroupWise account is set to "disable," to avoid unintentionally losing accounts if a mistake is made when the administrator is making changes to policies. As another example, the Identity Manager driver configurations don't revoke entitlements that have values from a user account in another system. If a user is granted membership in an e-mail distribution list, and if later the user no longer meets the criteria for the Entitlement policy, he or she is simply dropped from the policy membership. Accounts are disabled, but group membership and attribute values are not removed. An Identity Manager expert can customize the driver configurations if you want a different result.

The interpretation of revoking an entitlement is especially important because Role-Based Entitlements functionality gives you the ability to make sweeping changes in an organization's entitlements in a production environment, without testing the results in a lab.

You can change the settings for interpreting grant or revoke by editing the Global Configuration Variables on a preconfigured driver. If you are creating your own custom configuration, you could add GCVs to interpret granting and revoking entitlements.

6.10.2 Preventing Data Loss

Role-Based Entitlements are designed to allow you to make sweeping changes to entitlements, such as accounts, based on membership in the policy. This means, however, that mistakes made in changing policies are a concern. The driver configurations that ship with Identity Manager use the most benign settings. You should understand how to use GCVs to avoid unintentional data loss.

For example, we recommend that you never use delete as the value for the GCV that interprets revoking an account entitlement.

As another measure to protect your data when you edit or create a new entitlement policy, the driver is turned off so that changes are not made while your editing of policies is incomplete. You can then manually restart the driver when you are finished, using the *Restart* button in the Entitlement Policies interface. Similarly, if another user appears to be editing Entitlement policies, and you try to restart the Entitlements Service driver using the *Restart* button, you are prompted not to restart the driver until the other user is finished making changes.

6.10.3 Password Synchronization and Entitlements

Password Synchronization is managed the same way for drivers that are using Role-Based Entitlements as it is for other drivers, as described in "Password Synchronization across Connected Systems" on page 91.

Scheduling Jobs

7

Designer and iManager have a job scheduling utility to schedule events. Through this utility, the system can be set to disable an account on a specific day, or to initiate a workflow to request an extension for a person's access to a corporate resource, such as:

- Create a Job object from an installed job definition.
- Define when a job is to run, which servers the job is to run on, the scope of the job in terms of eDirectory objects, and the a job reports intermediate and final results.
- Set values for the job's parameters, its description and display name.
- Enable/disable a job, manually start a job, stop a job that is running, and display a list of running jobs.

- Section 7.1, "Scheduling Jobs in Designer," on page 215
- Section 7.2, "Scheduling Jobs in iManager," on page 230

7.1 Scheduling Jobs in Designer

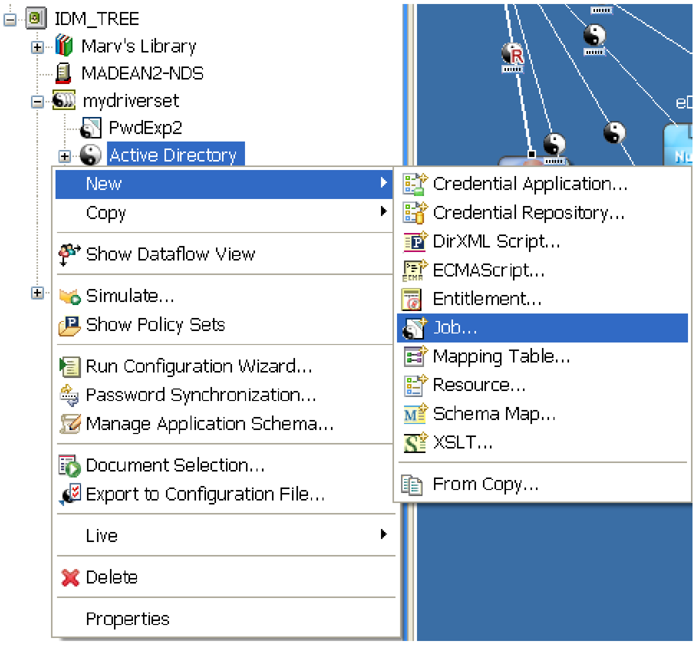
Designer's job scheduler contains most of the same functionality as the job scheduler found in iManager.

- Section 7.1.1, "Creating a Job," on page 215
- Section 7.1.2, "Editing a Job," on page 217

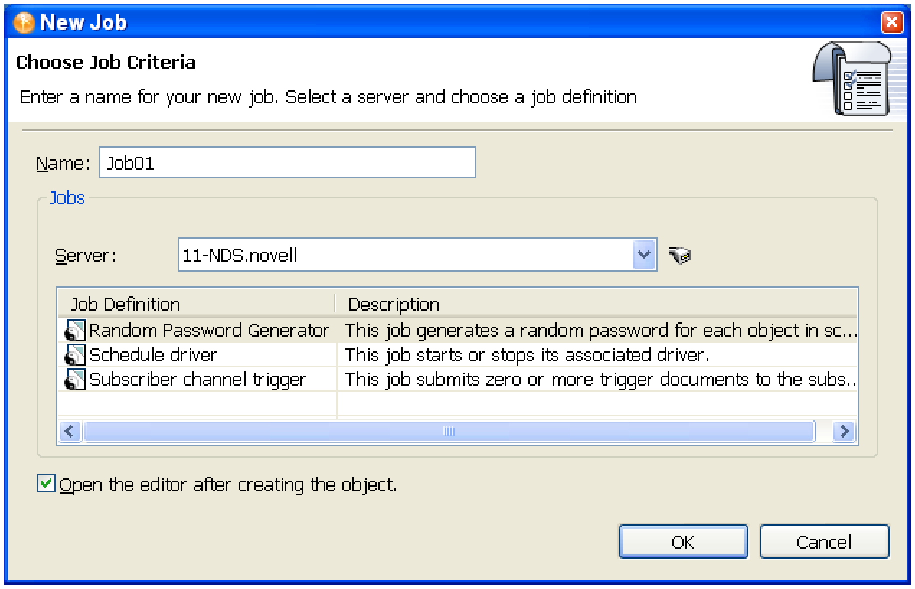
7.1.1 Creating a Job

To create a new job in Designer:

1 In the Outline view, right-click a driver and select *New* > *Job*.



2 In the New Job window, give the job a descriptive name, or use the name provided.



Choose Job Criteria

Enter a name for your new job. Select a server and choose a job definition

Name: Job01

Jobs

Server: 11-NDS.novell

Job Definition	Description
Random Password Generator	This job generates a random password for each object in sc...
Schedule driver	This job starts or stops its associated driver.
Subscriber channel trigger	This job submits zero or more trigger documents to the subs...

☑ Open the editor after creating the object.

3 In the *Server* field, select the server to perform the job execution. You can click the drop-down item to select a different server than the one listed.

4 If the administrator has created and installed any custom job definitions on the selected server, click the *Update Job Definitions from Server* icon to the right of the server entry to read in a live list of available jobs from the server. Because Designer is an offline modeling tool, only the Identity Manager 3.5 job definitions show in the server's *Job Definition* list by default.

5 Choose a definition for the job. The New Job Wizard comes with three job definitions; there might also be additional custom jobs listed.

   - **Random Password Generator:** Generates a random password for each object in the job's scope. The password is generated by NMAS™ to match the Password Policy object that the job references. These Password Policy objects are not usually the same as those used for eDirectory™ user password policies.

     The job submits the generated passwords one at a time to the driver's Subscriber channel. The Subscriber channel policies must take action on the passwords.

   - **Schedule Driver:** Starts or stops the associated driver. You can also toggle a driver to start the driver if it is stopped or to stop the driver if it is running.

   - **Subscriber Channel Trigger:** Submits zero or more trigger documents to the Subscriber channel. The submission can either be a document per object if a scope is defined, or it can be a single trigger event if no scope is defined.

     Trigger event documents identify the job and the scope object. A trigger event can bypass the cache and "go to the head of the queue" if desired. Trigger jobs allow you to use driver policies that you can customize for your personal requirements.

6 Decide if you want to edit the job. The *Open the editor after creating the object* option opens the newly created job in the IDM Job Editor window after it first saves the job object. If you do not want to open the editor at this time, deselect the editor.

7 Click *OK*.

8 The File Conflict window gives you the option to save the job object and continue. If you want to create the job object and continue with the IDM Job Editor, click *Yes*. Otherwise, click *No*.

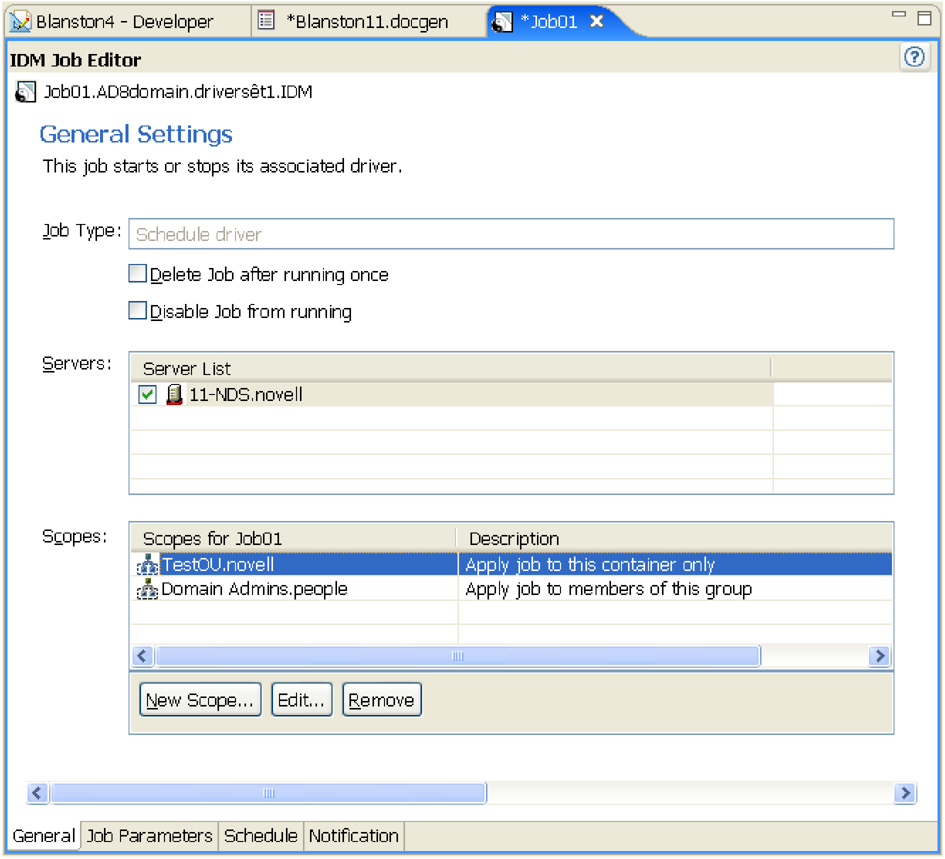
9 Continue with "Job Editor Selections Under the General Tab" on page 218.

7.1.2 Editing a Job

- "Job Editor Selections Under the General Tab" on page 218
- "Job Editor Selections Under the Job Parameter Tab" on page 221
- "Job Editor Selections Under the Scheduler Tab" on page 224
- "Job Editor Selections Under the Notification Tab" on page 226
- "Deploying a Job with Scope Objects" on page 228

After you create a job, you need to add the necessary information to make the job useful. To edit a job, double-click a newly created job in the Outline view to bring up the job in the IDM Job Editor view.

Figure 7-1 *The IDM Job Editor View*



Job01.AD8domain.driversêt1.IDM

General Settings

This job starts or stops its associated driver.

Job Type: Schedule driver

□ Delete Job after running once

□ Disable Job from running

Servers:

Server List
☑ 11-NDS.novell

Scopes:

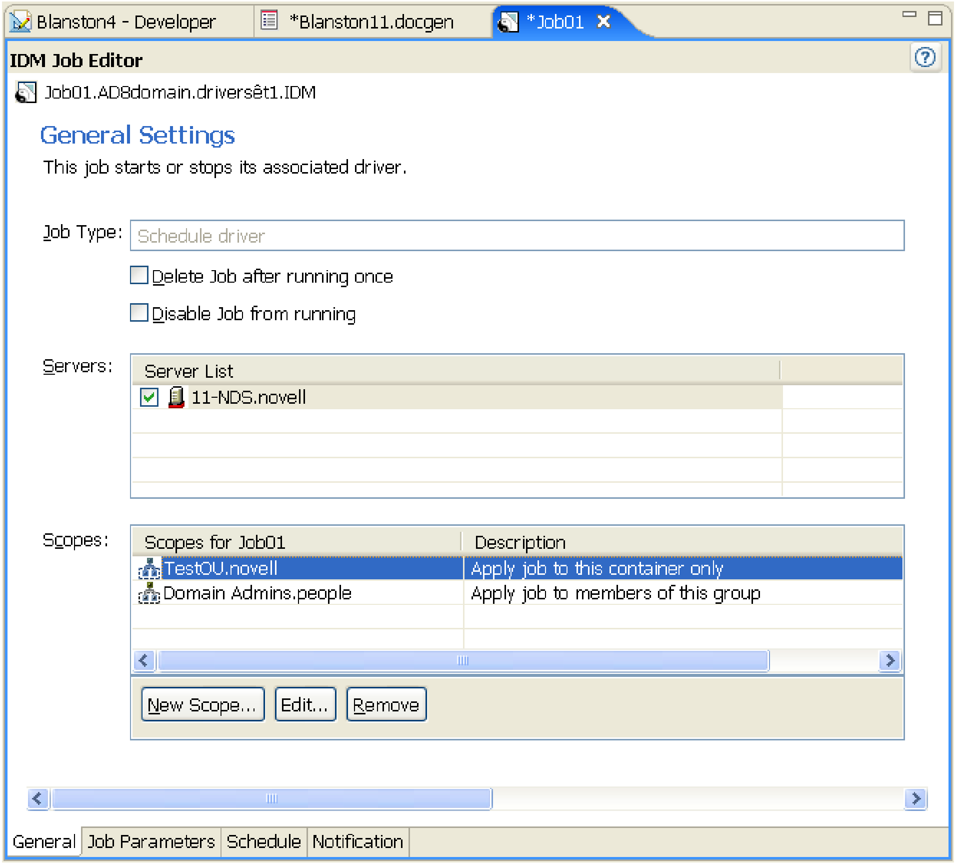
Scopes for Job01	Description
TestOU.novell	Apply job to this container only
Domain Admins.people	Apply job to members of this group

The IDM Job Editor has four tabs at the bottom of its view:

Job Editor Selections Under the General Tab

A line at the top of the General tab shows the Java class name of the job. This is followed by the job type, which shows the type of job you selected. Under the Job Type heading, you can enable or disable the job, or delete the job after it runs.

Figure 7-2 *Items Under the General Tab*



General Settings

This job starts or stops its associated driver.

Job Type: Schedule driver

☐ Delete Job after running once

☐ Disable Job from running

Servers:

Server List
☑ 11-NDS.novell

Scopes:

Scopes for Job01	Description
TestOU.novell	Apply job to this container only
Domain Admins.people	Apply job to members of this group

1 To delete the job after it runs, select *Delete job after running once*.

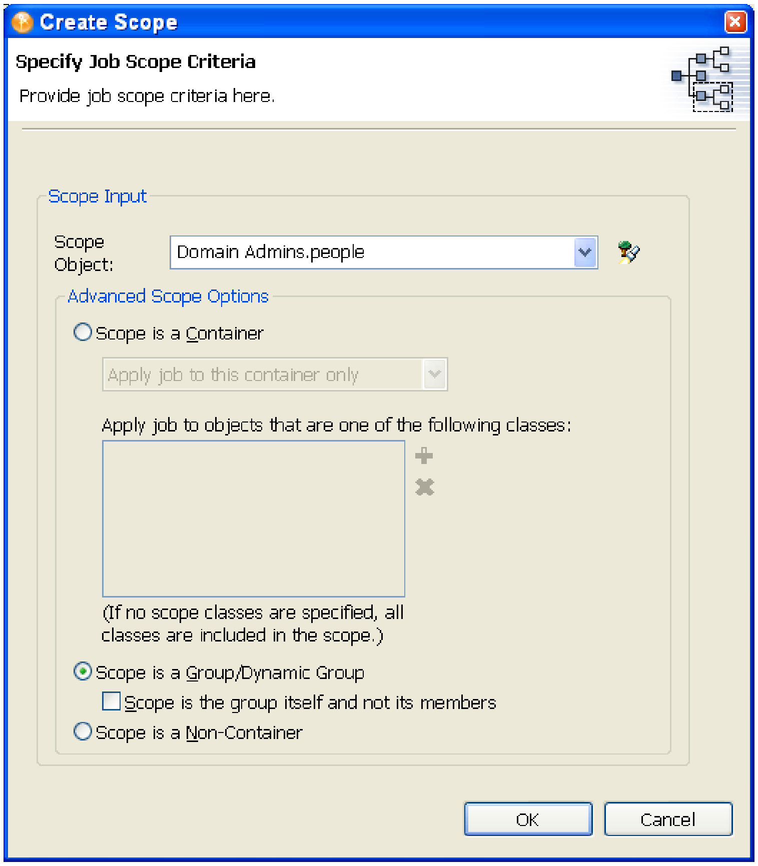
2 To disable the job from running, select *Disable job from running*.

3 In the *Server List* column, select the server or servers where this job should run.

A filtered list of servers is available to help you assign this job. A custom job may be installed on one server but not on another. In this case, the server without this custom job is filtered out of the Server List.

A job can be assigned to multiple servers as long as it has been installed on each server. Designer only allows this association if the jobs are properly installed and packaged so that the Metadirectory engine can see them.

4 To add a scope to the *Scopes* column, click *New Scope*.



Create Scope

Specify Job Scope Criteria

Provide job scope criteria here.

Scope Input

Scope Object: Domain Admins.people

Advanced Scope Options

○ Scope is a Container

Apply job to this container only

Apply job to objects that are one of the following classes:

(If no scope classes are specified, all classes are included in the scope.)

● Scope is a Group/Dynamic Group

□ Scope is the group itself and not its members

○ Scope is a Non-Container

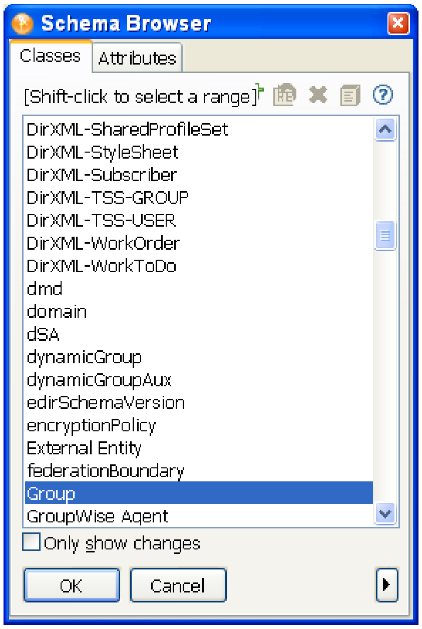
5 To select a scope object, type the Distinguished name of the object or use the *Browse* icon to browse to and select the object. Click *OK* to add the scope object.

Scopes allow you to define the objects that this job applies to. An object in eDirectory can be a container, a dynamic group, a group, or a leaf object. If you select a group object, you can apply the job to the group's members, or only to the group. If you select a container object, you can apply the job to all descendants in that container, to all of the children in the container, or to the container only.

6 If the object is a container, select *Scope is a Container*. Then select how you want to apply the job:

- Apply job to this container only
- Apply job to children of this container
- Apply job to all descendants of this container

7 (Optional) If you select *Apply job to children of this container* or *Apply job to all descendants of this container*, you can specify the classes you want to scope. Click the *Plus* icon to bring up the Schema Browser window to select the classes you want to scope. Select the class schema, then click *OK*.



The classes are added to the *Classes* box. To remove a class, select it and click the Minus icon.

8 If the object is a group or a dynamic group, select *Scope is a Group/Dynamic Group*. You can then select the *Scope is the group itself and not its members* option if the scope is for the group.

9 If the object is a non-container, select *Scope is a Non-Container*.

10 After the scope criteria are selected, click *OK* to return to the General Settings page.

11 If you need to edit a scope, select the scope name, then click *Edit*.

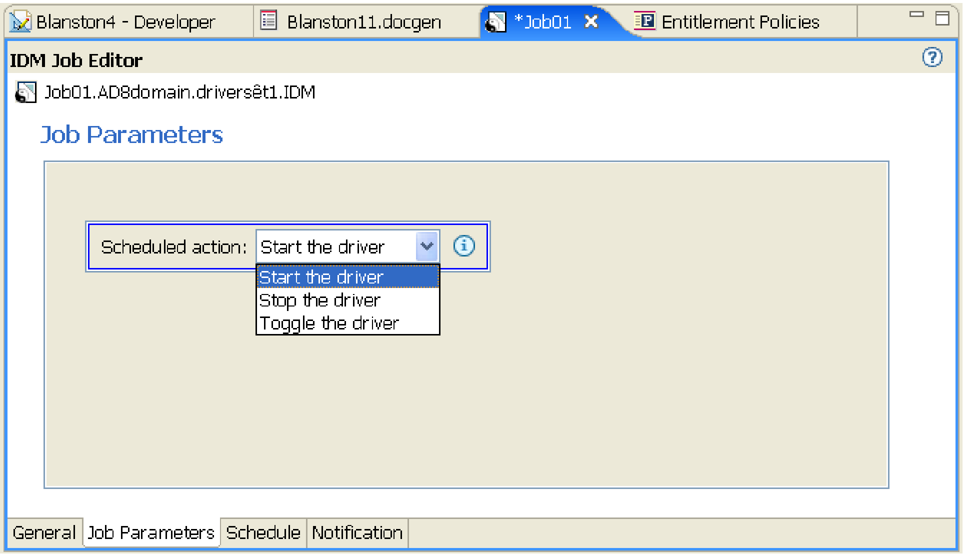
12 To remove a scope, select the scope name, then click *Remove*.

Job Editor Selections Under the Job Parameter Tab

The Job Parameter page allows you to add additional parameters to the job and to view the parameters as they are presently set up. What you can do depends on the type of job you selected.

Parameters for the Schedule Driver Job

Figure 7-3 *The Job Parameter Page for a Schedule Driver Job*

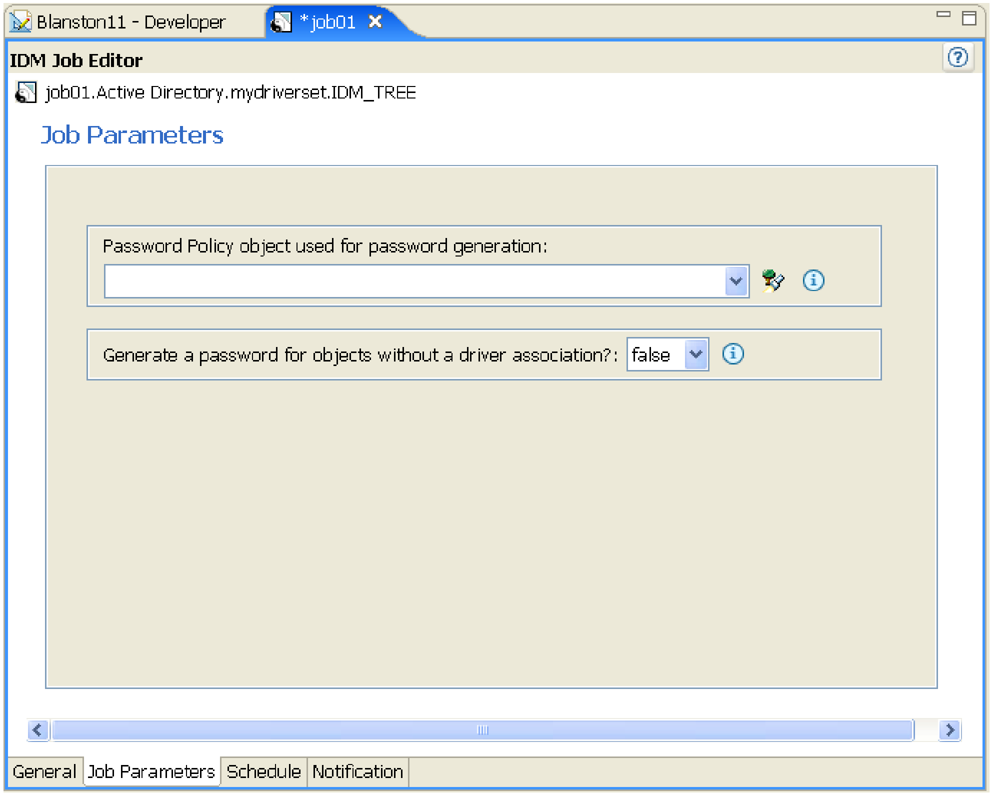


Scheduled action: Start the driver

1. If you want the job to start the driver, select *Start the driver*.
2. If you want the job to stop the driver, select *Stop the driver*.
3. If you want the job to stop the driver if it's started or to start the driver if it's stopped, select *Toggle the driver*.

Parameters for the Generate Random Passwords Job

Figure 7-4 *The Job Parameter Page for the Generate Random Password Job*



Password Policy object used for password generation:

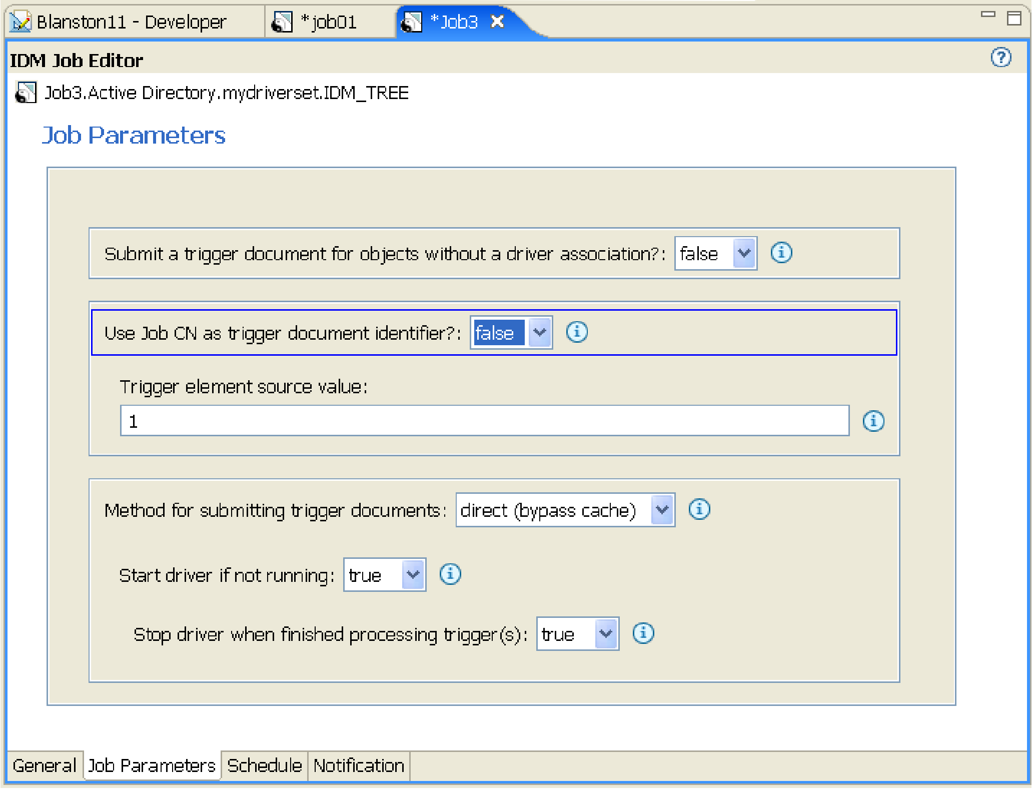
Generate a password for objects without a driver association?: false

1 Type the Password policy object's Distinguished name, or use the *Browse* icon to select the Password policy you want to use for password generation.

2 If you want to generate passwords for scoped objects without a driver association, select *True*. Otherwise, select *False*.

Parameters for the Subscriber Channel Trigger Job

Figure 7-5 *The Job Parameter Page for the Subscriber Channel Trigger Job*



Submit a trigger document for objects without a driver association?: false

Use Job CN as trigger document identifier?: false

Trigger element source value:

1

Method for submitting trigger documents: direct (bypass cache)

Start driver if not running: true

Stop driver when finished processing trigger(s): true

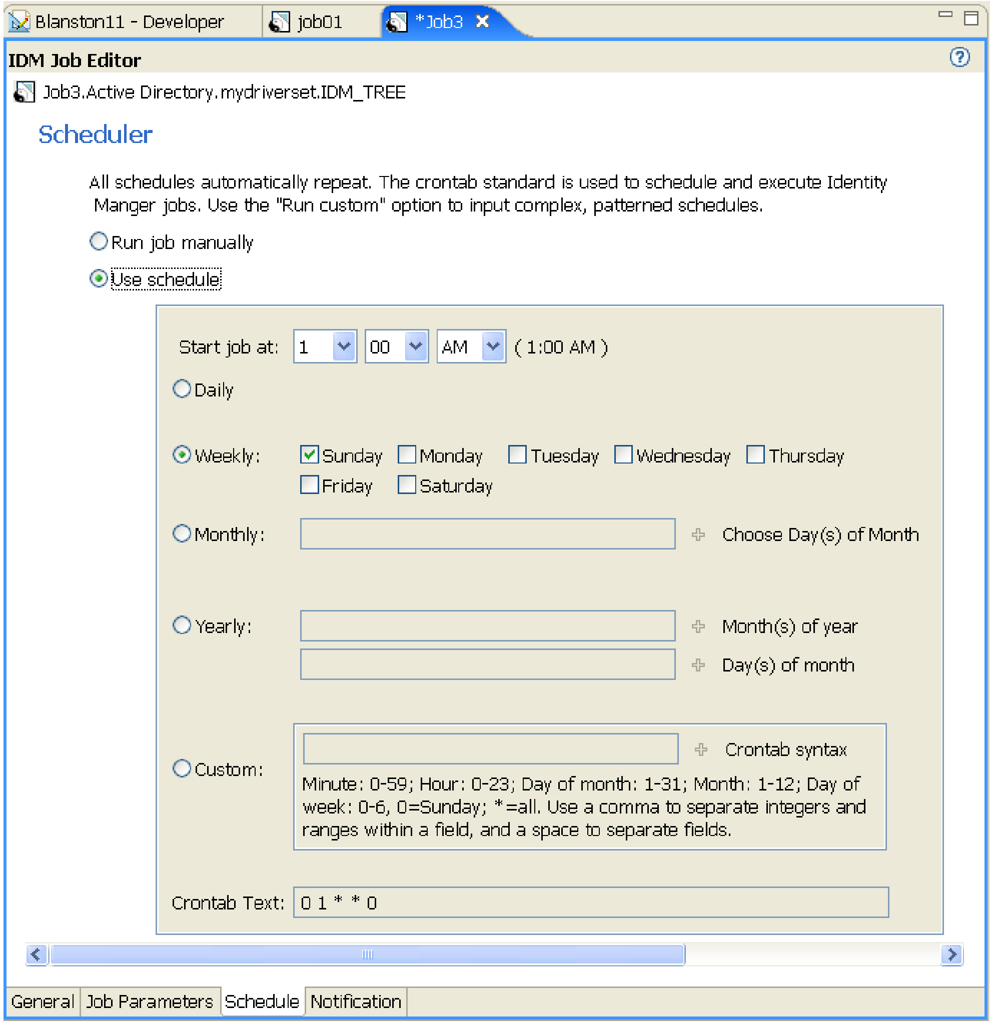
1. If you want to submit a trigger document for scoped objects that do not have a driver association, select *True*. Otherwise, keep the default of *False*.
2. If you want to use the job's Common Name (CN) as a document identifier trigger, keep the default of *True*. Otherwise, select *False*.
3. (Optional) If you select *False*, specify the string that the job can use as the value for the trigger element's Source attribute.
4. Select a method for submitting the trigger documents. If you want to queue the job the trigger is from, keep the default of *Queue (use cache)*. Otherwise, select *Direct (bypass cache)*.
5. (Optional) If you select *Direct (bypass cache)*, you are presented with the *Start driver if not running* option. If you want to start the driver if it is not running, keep the default of *True*. Otherwise, select *False*.
6. (Optional) If you select *True* on the *Start driver if not running* option, you are presented with the *Stop driver when finished processing triggers* option with the default of *True*. Use the default to stop the driver once it finishes processing the trigger job, or select *False* to keep the driver running.

Customized job definitions have their own parameter sets.

Job Editor Selections Under the Scheduler Tab

The Scheduler tab allows you to set up when you want to run the job.

Figure 7-6 *The Job Options for the Scheduler Tab*



Job3.Active Directory.mydriverset.IDM_TREE

Scheduler

All schedules automatically repeat. The crontab standard is used to schedule and execute Identity Manger jobs. Use the "Run custom" option to input complex, patterned schedules.

○ Run job manually

● Use schedule

Start job at: 1 00 AM (1:00 AM)

○ Daily

● Weekly: ☑ Sunday □ Monday □ Tuesday □ Wednesday □ Thursday □ Friday □ Saturday

○ Monthly: Choose Day(s) of Month

○ Yearly: Month(s) of year
Day(s) of month

○ Custom: Crontab syntax

Minute: 0-59; Hour: 0-23; Day of month: 1-31; Month: 1-12; Day of week: 0-6, 0=Sunday; *=all. Use a comma to separate integers and ranges within a field, and a space to separate fields.

Crontab Text: 0 1 * * 0

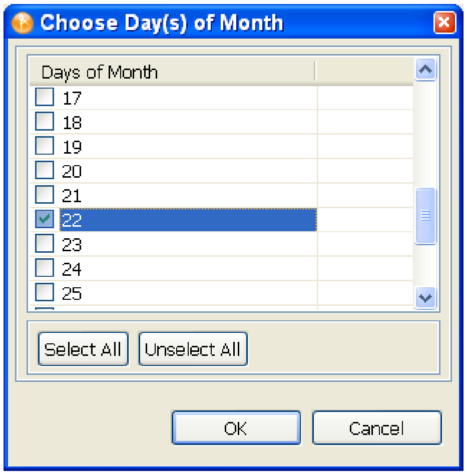
1 Select the *Use schedule* option to set the date and time, and whether to run the job daily, weekly, monthly, yearly.

Select the *Run job manually* option to run the job when you choose to.

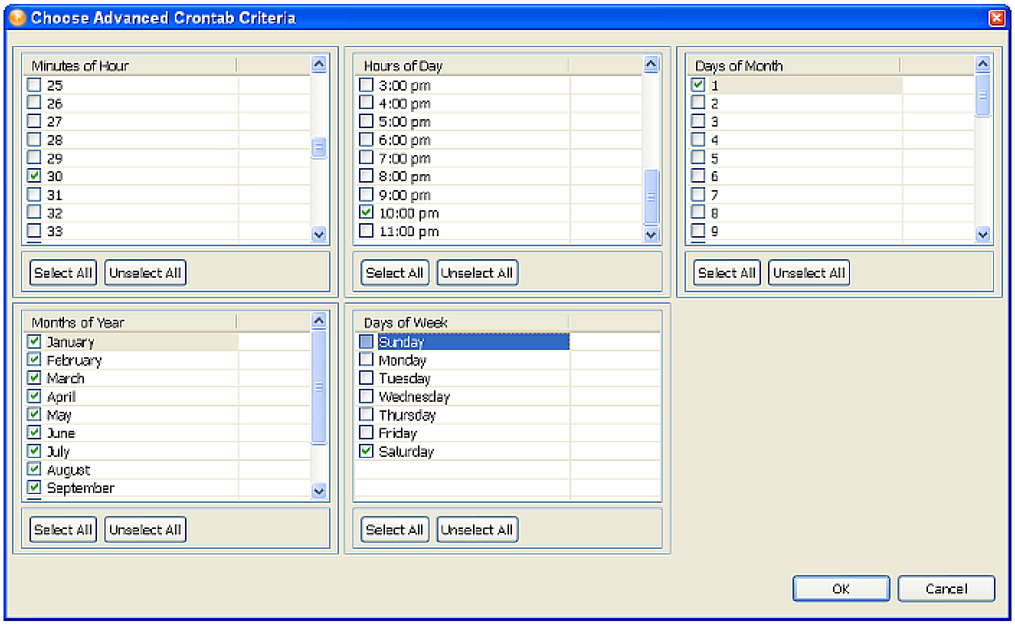
2 With *Use schedule* selected, set the time when you want the job to start running. Use the drop-down menus to select the hours, minutes, and *AM* or *PM*. The default is 1:00 AM.

3 If you want to run the job repeatedly, use the *Daily*, *Weekly*, *Monthly*, *Yearly*, or *Custom* fields to select when you want it to run.

For example, if you want the job to run weekly, select *Weekly*, then select the day you want it to run. If you want the job to run once a month, select *Monthly*, then click the *Plus* icon to select the day of the month.



4 (Optional) Select *Custom* to choose minutes, hours, days, months, and days of the week from the Choose Advanced Crontab Criteria page.



5 The Choose Advanced Crontab Criteria page default has everything selected. Click *Unselect All*, choose the time and days you want to run the job, then click *OK* to return to the Scheduler page.

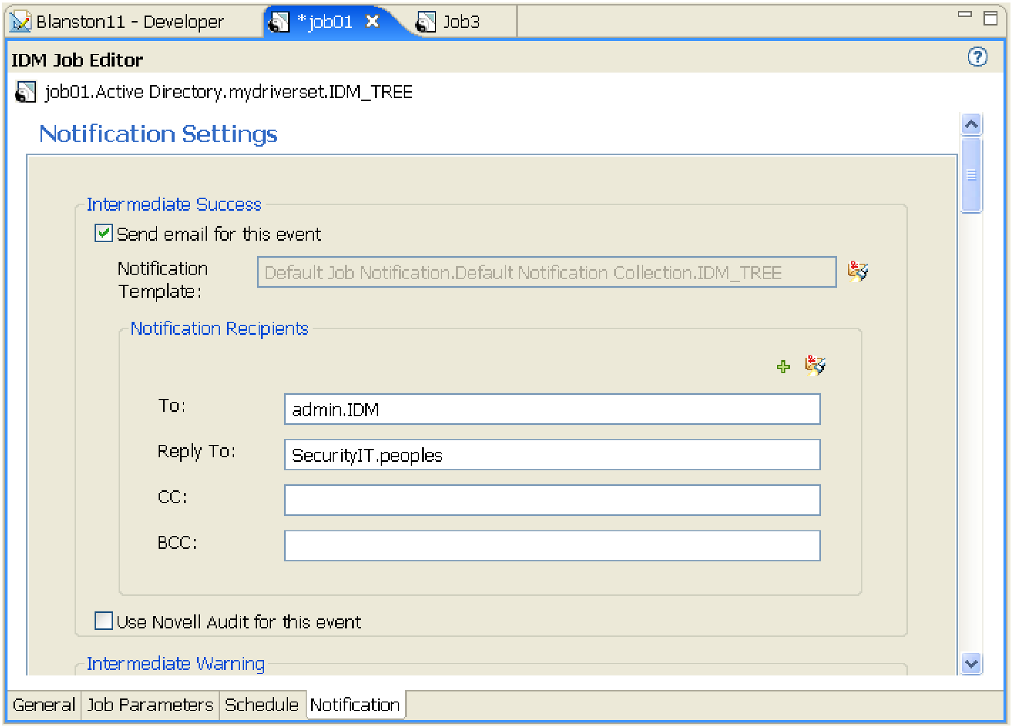
The information displayed in the *Crontab Text* field displays any settings you make on the Scheduler page. For example, if you click *Monthly* and select two days, those two days are displayed in the *Crontab Text* field.

Job Editor Selections Under the Notification Tab

The Notification Settings page allows you to define what you want to do with the job results. It is divided into two parts, *Intermediate* and *Final*, with the *Success*, *Warning*, *Error*, and *Aborted* results for each part.

The Notification Settings page allows you to set how you want to be notified for each result. Actions include sending an audit result or sending an e-mail when the result completes.

Figure 7-7 *The Job Options Under the Notification Tab*



☑ Send email for this event

Notification Template: Default Job Notification.Default Notification Collection.IDM_TREE

To: admin.IDM

Reply To: SecurityIT.peoples

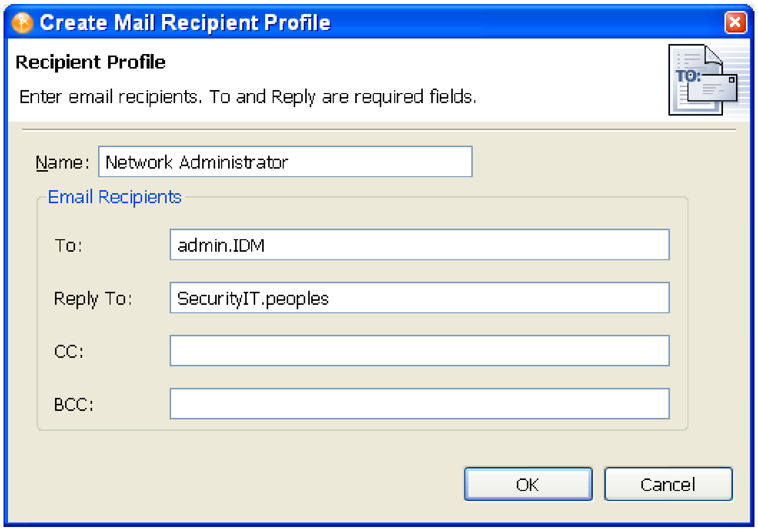
CC:

BCC:

□ Use Novell Audit for this event

1 If you select *Send email for this event*, Designer supplies the *Default Job Notification.Default Notification.security* template in the *Notification Template* field. To change templates, click the *Model Browser* icon to select another template.

2 Under Notification Recipients, select who you want to send the results to by typing the user's or group's fully Distinguished name. You can use the *Plus* icon to create a mail profile.



Enter email recipients. To and Reply are required fields.

Name: Network Administrator

To: admin.IDM

Reply To: SecurityIT.peoples

CC:

BCC:

The *To* and *Reply* fields are required for a profile.

3. When you have filled in the information, click *OK*.
4. If you want the results to go to Novell® Audit, select *Use Novell Audit for this event*.
5. Use Steps 1 through Step 4 for each of the options:
   - *Intermediate Success*
   - *Intermediate Warning*
   - *Intermediate Error*
   - *Intermediate Abort*
   - *Final Success*
   - *Final Warning*
   - *Final Error*
   - *Final Abort*

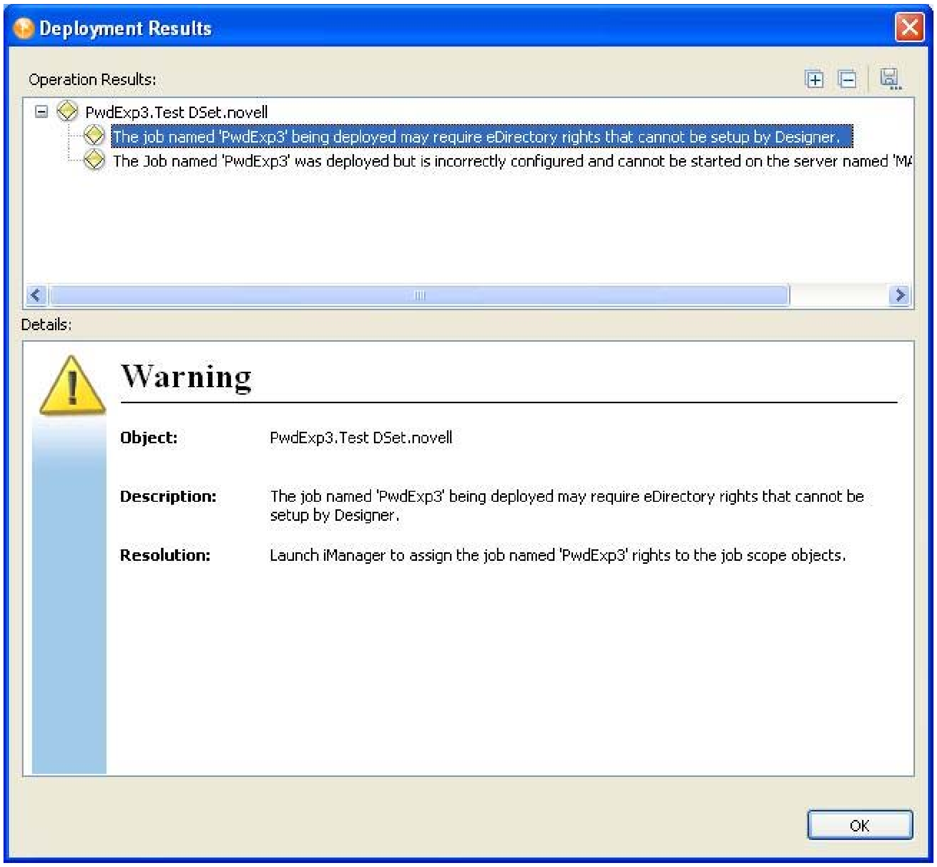
   If you do not select an option, no action is taken for the result.

Deploying a Job with Scope Objects

Jobs might need access to eDirectory data and certain Identity Manager actions, such as starting and stopping drivers. Such access is subject to eDirectory rights assignments and is controlled by the rights that are granted to the DirXMLJob object. Identity Manager actions are controlled by special attributes, but normal eDirectory rights are needed for data reads and writes.

When you deploy a job object that has scope objects, there might be eDirectory rights assignments that Designer cannot properly set up. The rights needed to complete the task depend on the scope objects that are assigned to the job object.

Figure 7-8 Warning Messages When Deploying a Job with Scope Objects



Deployment Results

Operation Results:

PwdExp3.Test DSet.novell

The job named 'PwdExp3' being deployed may require eDirectory rights that cannot be setup by Designer.

The Job named 'PwdExp3' was deployed but is incorrectly configured and cannot be started on the server named 'MA

Details:

Warning

Object: PwdExp3.Test DSet.novell

Description: The job named 'PwdExp3' being deployed may require eDirectory rights that cannot be setup by Designer.

Resolution: Launch iManager to assign the job named 'PwdExp3' rights to the job scope objects.

If you see this warning when deploying job objects, use the iManager utility to assign eDirectory rights to the job object so it can properly access the job scope objects and complete its task.

You can assign the eDirectory rights either from the *View Objects* option or from the *Identity Manager Overview* option. To assign eDirectory rights to the job object, do the following:

1. In Novell iManager, click the *View Objects* icon.
2. Select the job object from the *DriverSet* in the *Tree* tab.
3. In the View pane, select *(current level)*.
4. From the *Actions* list, select *Modify Inherited Rights Filter.*
5. In the Modify Inherited Rights Filter window, click *Add Property*.
6. From the *Add property* list, select the related property, and click *OK*.
7. Change the rights as required in the Modify Inherited Rights Filter window, and click *OK*.
8. From the View Effective Rights window, select the property from the *Property name* list.

Another procedure to assign the eDirectory rights is as below:

1. In Novell iManager, click the *Identity Manager Administration* icon.
2. In the Administration list, click *Identity Manager Overview*.
3. Click the Active Directory Parent or Active Directory Child for which the job is created.

4 Click the *Jobs* tab, select the job to be assigned with rights, then click *Get Status*.

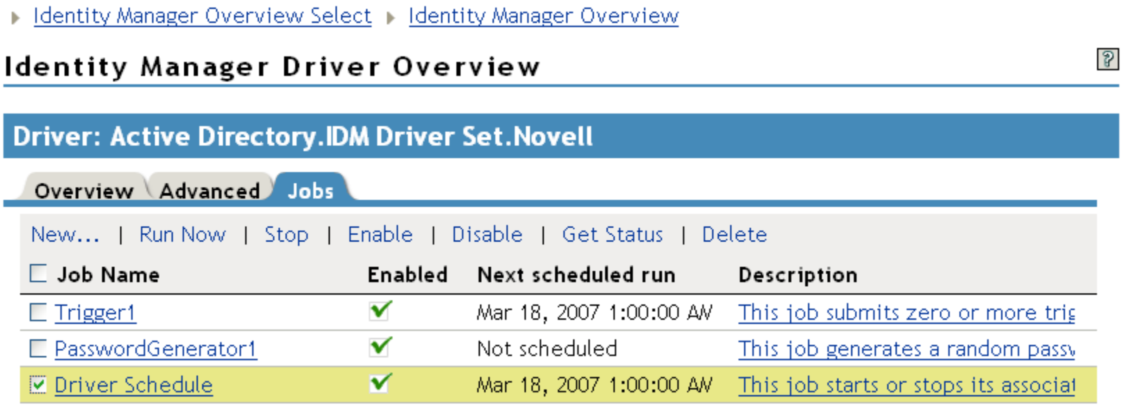
5 In the Job Status window, click the *Grant rights* link, then click *Close*.

7.2 Scheduling Jobs in iManager

iManager's job scheduler contains the similar functionality as the job scheduler found in Designer. However, iManager's job scheduler can start and stop a job, as well as get a job's status.

The Jobs page is accessed from the Identity Manager Driver Overview page and contains a table showing the existing job objects for the selected driver, which is listed with the name of the job object. Below is an example of scheduled jobs that are created for the Job driver.

Figure 7-9 *The Job's Tab from the Identity Manager Driver Overview*



Identity Manager Overview Select ▸ Identity Manager Overview

Identity Manager Driver Overview

Driver: Active Directory.IDM Driver Set.Novell

Overview
Advanced
Jobs
New... | Run Now | Stop | Enable | Disable | Get Status | Delete

□ Job Name	Enabled	Next scheduled run	Description
□ Trigger1	✓	Mar 18, 2007 1:00:00 AM	This job submits zero or more trig
□ PasswordGenerator1	✓	Not scheduled	This job generates a random passw
☑ Driver Schedule	✓	Mar 18, 2007 1:00:00 AM	This job starts or stops its associat

7.2.1 Jobs Column Headings

The headings include the job's name, whether the job is enabled or disabled, when it is scheduled to run, and the job description. Click the job's name to bring up the Job Properties page. Click the enable/disable icon under the Enabled column to enable or disable the job. The Next Scheduled Run heading shows if the job is scheduled to run, and if it is, the date when the job is scheduled. Click the job's description to see a pop-up window listing the job's full description.

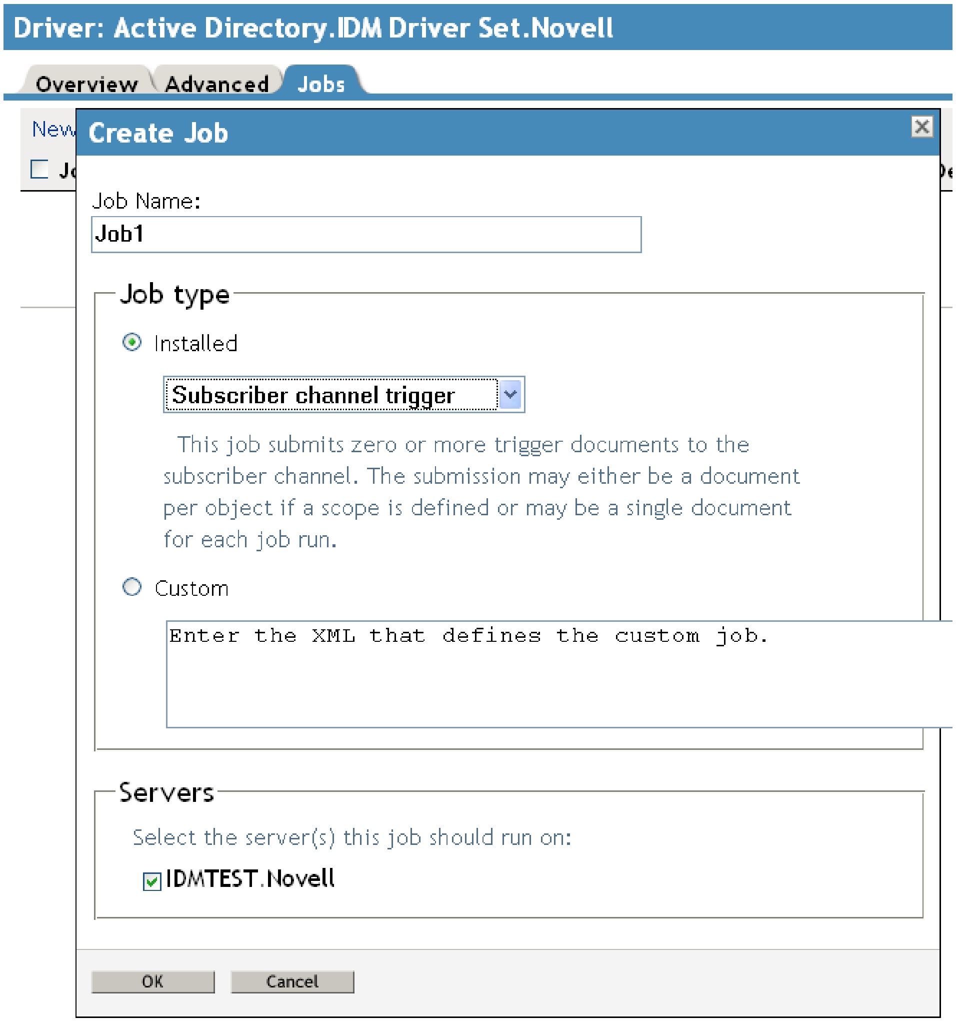
Use the commands in the menu bar to perform the following operations.

- "New" on page 230
- "Run Now" on page 232
- "Stop" on page 232
- "Enable" on page 232
- "Disable" on page 232
- "Get Status" on page 232
- "Delete" on page 232

New

To create a new job in iManager:

1 Click New to bring up the Create Job page.



Driver: Active Directory.IDM Driver Set.Novell

Create Job

Job Name: Job1

Job type

● Installed

Subscriber channel trigger

This job submits zero or more trigger documents to the subscriber channel. The submission may either be a document per object if a scope is defined or may be a single document for each job run.

○ Custom

Enter the XML that defines the custom job.

Servers

Select the server(s) this job should run on:

☑ IDMTEST.Novell

2 In the Job Name entry, give the job a descriptive name.

3 Under *Job Type*, select an *Installed* or a *Custom* job. The default is *Installed*.

3a Under *Installed*, choose a definition for the job. The Create Job page displays all jobs that have been installed on the servers. Identity Manager 3.5 comes with three job definitions:

- **Random Password Generator:** Generates a random password for each object in the job's scope. The password is generated by NMAS to match the Password Policy object that the job references. These Password Policy objects are not usually the same as those used for eDirectory user password policies.

  The job submits the generated passwords one at a time to the driver's Subscriber channel. The Subscriber channel policies must take action on the passwords.

- **Schedule Driver:** Starts or stops the associated driver. You can also toggle a driver to start the driver if it is stopped or to stop the driver if it is running.

- **Subscriber Channel Trigger:** Submits zero or more trigger documents to the Subscriber channel. The submission can either be a document per object if a scope is defined, or it can be a single trigger event if no scope is defined.

  Trigger event documents identify the job and the scope object. A trigger event can bypass the cache and "go to the head of the queue" if desired. Trigger jobs allow you to use driver policies that you can customize for your personal requirements.

3b If you select Custom, enter the XML code that defines the custom job.

4 In the *Server* field, select the server or servers to perform the job execution.

5 Click *OK*. The new job opens Job Property page.

6 Continue with Section 7.2.2, "Setting the Job's Parameters," on page 232.

Run Now

Select a job by clicking the box to the left of the job, then click Run Now.

Stop

Select a job by clicking the box to the left of the job, then click Stop.

Enable

Select a job by clicking the box to the left of the job, then click *Enable*. The *Disabled* icon in the *Enabled* column changes to a check mark when the job is enabled.

Disable

Select a job by clicking the box to the left of the job, then click *Disable*. The *Enabled* icon in the *Enabled* column changes to the *Disabled* icon.

Get Status

Select a job by clicking the box to the left of the job, then click *Get Status*.

Delete

To delete a job, click the box to the left of the job name, then click *Delete*. You see a message stating that the selected job will be deleted from the directory. Click *OK* to delete the job, or click *Cancel* to stop the operation. You can click multiple boxes to delete multiple jobs, or click the upper left box to delete all of the jobs listed.

7.2.2 Setting the Job's Parameters

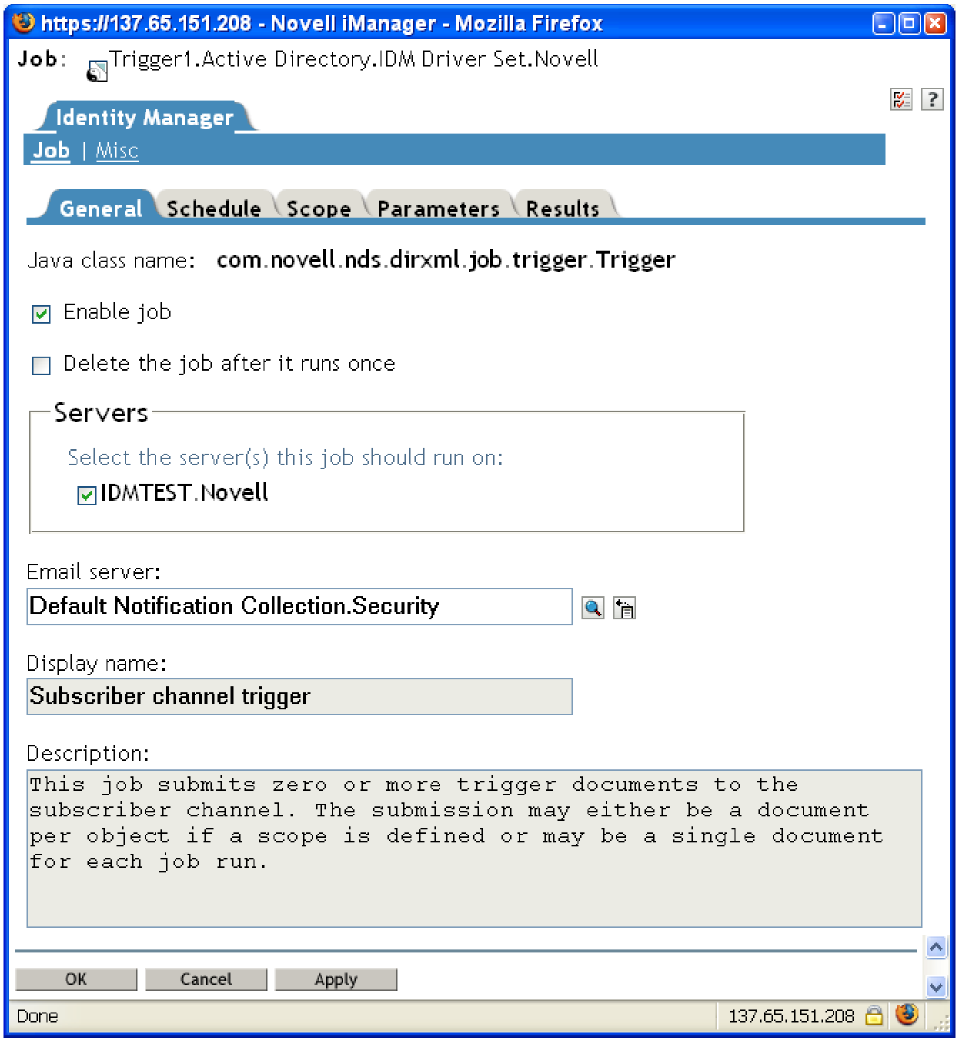
Click a job to bring up the Job Property page and set up how you want the job to run.

- "Selections Under the General Tab" on page 233
- "Selections Under the Schedule Tab" on page 234
- "Selections Under the Scope Tab" on page 236
- "Selections Under the Parameters Tab" on page 238
- "Selections Under the Results Tab" on page 239

Selections Under the General Tab

A line at the top of the General tab shows the Java class name of the job.

Figure 7-10 *Items Under the General Tab*



Job: Trigger1.Active Directory.IDM Driver Set.Novell

Java class name: com.novell.nds.dirxml.job.trigger.Trigger

☑ Enable job

☐ Delete the job after it runs once

Servers

Select the server(s) this job should run on:

☑ IDMTEST.Novell

Email server: Default Notification Collection.Security

Display name: Subscriber channel trigger

Description:

```
This job submits zero or more trigger documents to the
subscriber channel. The submission may either be a document
per object if a scope is defined or may be a single document
for each job run.
```

1 To enable the job, select *Enable job*.

2 To delete the job after it runs, select *Delete job after it runs once*.

3 In the *Server* column, select the server or servers where this job should run.

A filtered list of servers is available to help you assign this job. A custom job may be installed on one server but not on another. In this case, the server without this custom job is filtered out of the Server List.

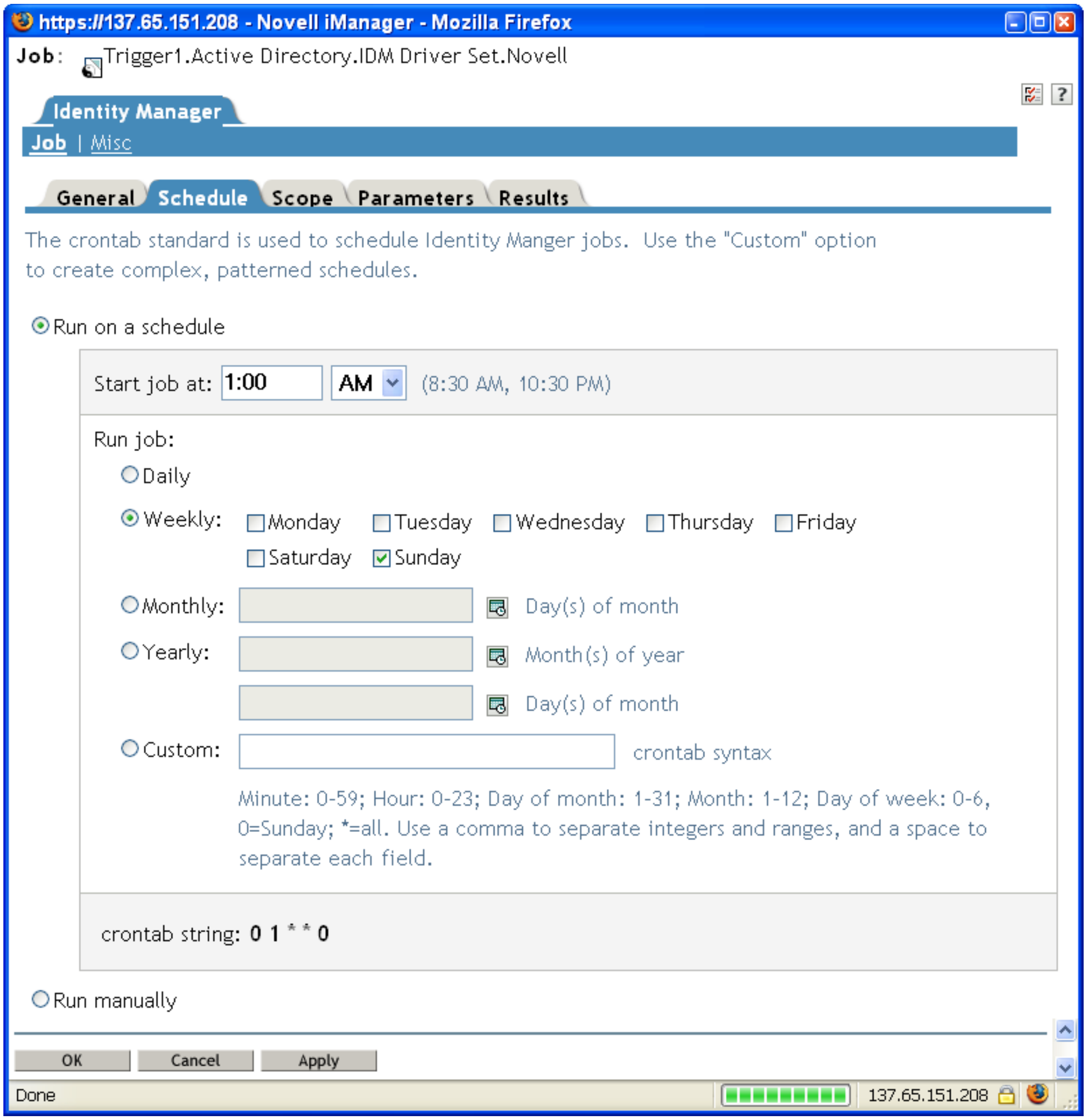
A job can be assigned to multiple servers as long as it has been installed on each server. iManager only allows this association if the jobs are properly installed and packaged so that the Metadirectory engine can see them.

4. For the *Email Server* entry, browse to the default notification collection template. Select the browse button, select the Security container, then the Default Notification Collection object. The fully distinguished path is Default Notification Collection.Security. Or you can use the history icon to select from previously selected items.
5. Check the other job settings:
   - The *Display Name* entry shows which type of job has been selected.
   - The *Description* field displays the description that has been written for the job type.

Selections Under the Schedule Tab

The Schedule tab allows you to set up when you want to run the job.

Figure 7-11 The Job Options for the Scheduler Tab



Job: Trigger1.Active Directory.IDM Driver Set.Novell

The crontab standard is used to schedule Identity Manger jobs. Use the "Custom" option to create complex, patterned schedules.

● Run on a schedule

Start job at: 1:00 AM (8:30 AM, 10:30 PM)

Run job:

○ Daily

● Weekly: □ Monday □ Tuesday □ Wednesday □ Thursday □ Friday □ Saturday ☑ Sunday

○ Monthly: Day(s) of month

○ Yearly: Month(s) of year

Day(s) of month

○ Custom: crontab syntax

Minute: 0-59; Hour: 0-23; Day of month: 1-31; Month: 1-12; Day of week: 0-6, 0=Sunday; *=all. Use a comma to separate integers and ranges, and a space to separate each field.

crontab string: 0 1 * * 0

○ Run manually

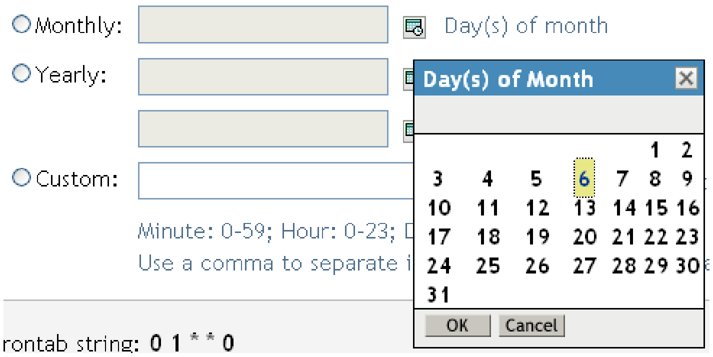
1 Select the *Run on a schedule* option to set the date and time, and whether to run the job daily, weekly, monthly, yearly.

Or select the *Run manually* option to run the job when you choose to.

2 With *Run on a schedule* selected, type the hours and minutes when you want the job to start running. Use the drop-down menus to select *AM* or *PM*. The default is 1:00 AM.

3 If you want to run the job repeatedly, use the *Daily*, *Weekly*, *Monthly*, *Yearly*, or *Custom* fields to select when you want it to run.

For example, if you want the job to run weekly, select *Weekly*, then select the day you want it to run. If you want the job to run once a month, select *Monthly*, then click the *Calendar* icon to select the day of the month.



rontab string: 0 1 * * 0

4 If you select *Custom*, you can choose minutes, hours, days, months and days of the week by using crontab syntax:

- Minute: 0-59; Hour: 0-23; Day of month: 1-31; Month: 1-12; Day of week: 0-6, 0=Sunday; *=all.
- Use a comma to separate integers and ranges, and a space to separate each field.

The information displayed in the *crontab string* field displays any settings you make in the Custom entry. For example, if you type `30,2,1,12,5` and press *Enter*, those numbers are displayed in the *crontab string* field.

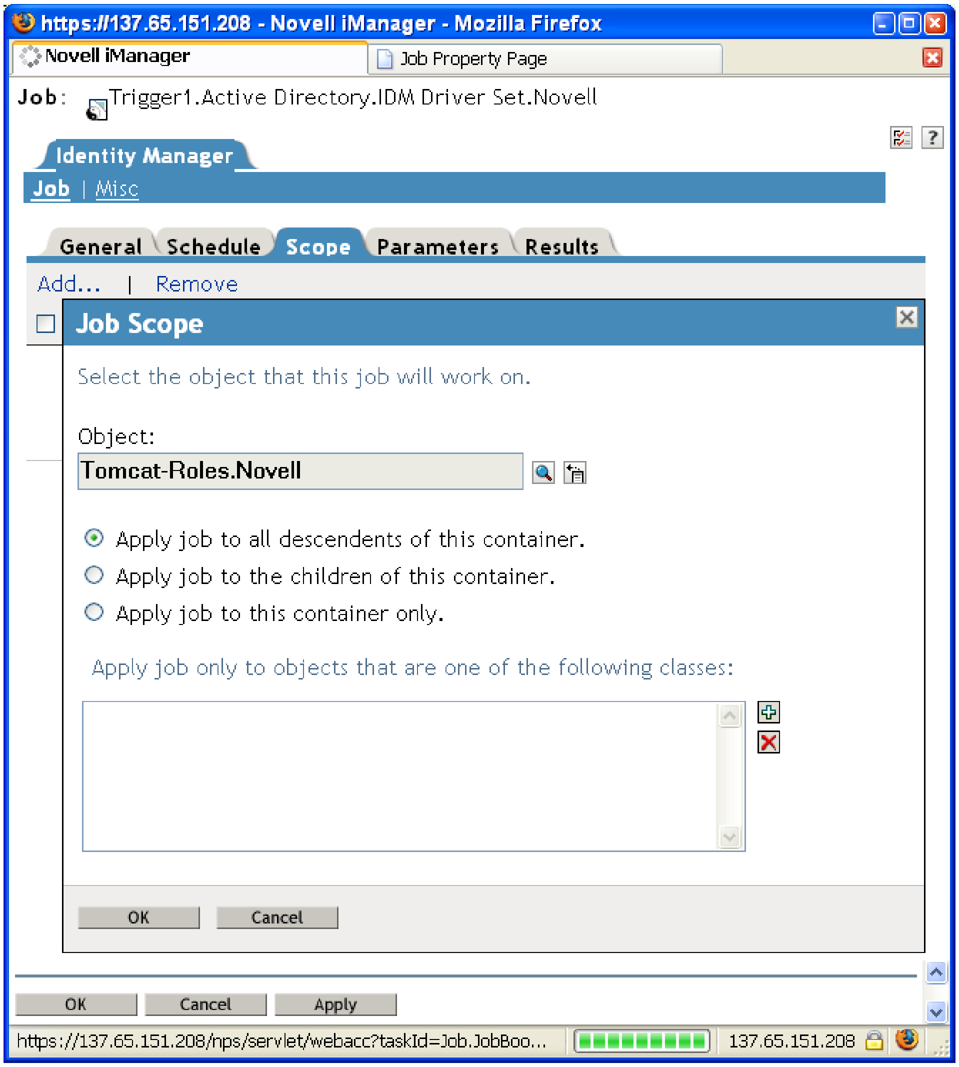
5 Once you have selected the day or days you want to schedule the job, click *OK*.

Selections Under the Scope Tab

Scopes allow you to define the objects that this job applies to. An object in eDirectory can be a container, a dynamic group, a group, or a leaf object. If you select a group object, you can apply the job to the group's members, or only to the group. If you select a container object, you can apply the job to all descendants in that container, to all of the children in the container, or to the container only.

1 Click *Add* to add a scope to the job object.

2 On the Job Scope page, use the *Browse* icon to browse to the object. Click *OK* to add the scope object.



Job: Trigger1.Active Directory.IDM Driver Set.Novell

Job Scope

Select the object that this job will work on.

Object: Tomcat-Roles.Novell

● Apply job to all descendents of this container.

○ Apply job to the children of this container.

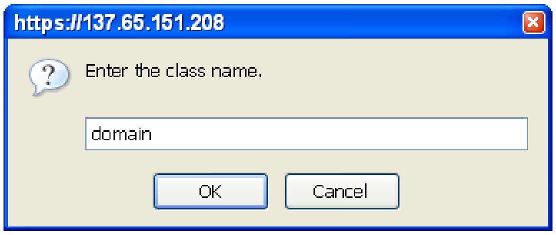
○ Apply job to this container only.

Apply job only to objects that are one of the following classes:

3 If the object is a container, select how you want to apply the job:

- *Apply job to all descendents of this container*
- *Apply job to the children of this container*
- *Apply job to this container only*

4 (Optional) If you select *Apply job to the children of this container* or *Apply job to all descendents of this container*, you can specify the classes you want to scope. Click the *Plus* icon to bring up the Enter the Class Name page and specify the classes you want to scope. Then click *OK*.



Enter the class name.

domain

The classes are added to the *Apply job only to objects that are one of the following classes* box. To remove a class, select it and click the *Minus* icon.

5 If the object is a group or a dynamic group, select the *Apply job to the members of this group* or *Apply job to this group only* if the scope is for the group.

6 After the scope criteria are selected, click *OK* to add the scope job to the Job Scope page.

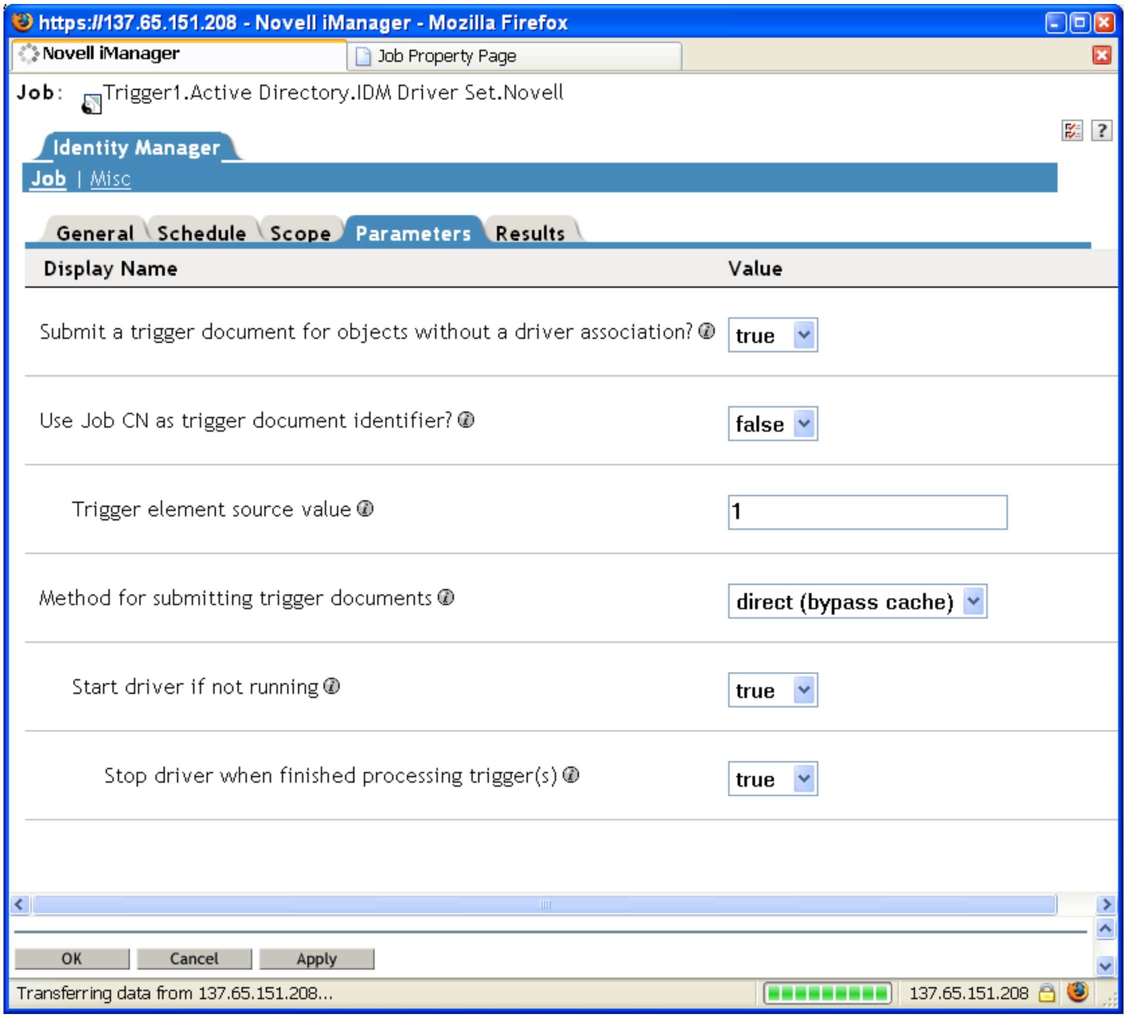
7 If you need to edit a scope, click the scope name.

8 To remove a scope, select the scope name, then click *Remove*.

Selections Under the Parameters Tab

The Parameter page allows you to set parameters used by the job when the job is doing its work. What you can do depends on the type of job you selected. The following procedure uses a Subscriber Channel trigger for its example.

Figure 7-12 *The Job Parameter Page for the Subscriber Channel Trigger Job*



Job: Trigger1.Active Directory.IDM Driver Set.Novell

Display Name	Value
Submit a trigger document for objects without a driver association?	true
Use Job CN as trigger document identifier?	false
Trigger element source value	1
Method for submitting trigger documents	direct (bypass cache)
Start driver if not running	true
Stop driver when finished processing trigger(s)	true

1 If you want to submit a trigger document for scoped objects that do not have a driver association, select *True*. Otherwise, keep the default of *False*.

2. If you want to use the job's Common Name (CN) as a document identifier trigger, keep the default of *True*. Otherwise, select *False*.
3. (Optional) If you select *False*, specify the string that the job can use as the value for the trigger element's Source attribute.
4. Select a method for submitting the trigger documents. If you want to queue the job the trigger is from, keep the default of *Queue (use cache)*. Otherwise, select *Direct (bypass cache)*.
5. (Optional) If you select *Direct (bypass cache)*, you are presented with the *Start driver if not running* option. If you want to start the driver if it is not running, keep the default of *True*. Otherwise, select *False*.
6. (Optional) If you select *True* for the *Start driver if not running* option, you are presented with the *Stop driver when finished processing triggers* option with the default of *True*. Use the default to stop the driver after it finishes processing the trigger job, or select *False* to keep the driver running.

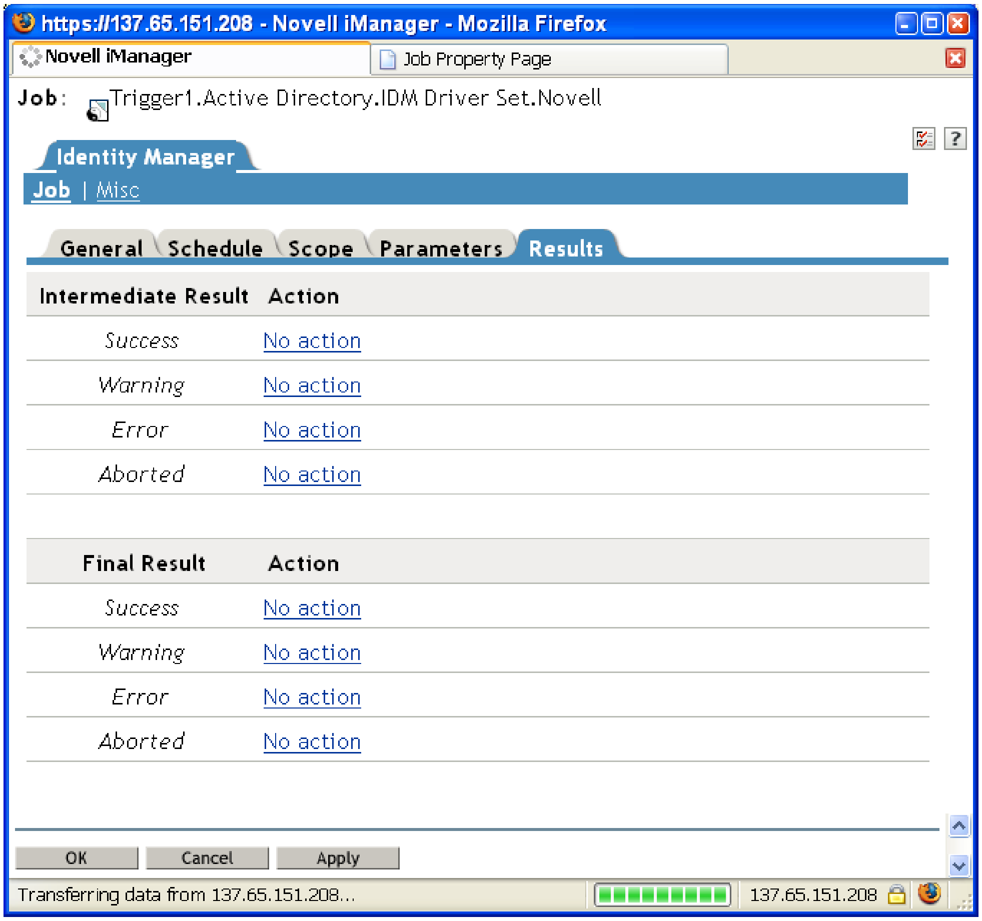
All job definitions have their own parameter sets.

Selections Under the Results Tab

The *Results* tab allows you to define what you want to do with the job results. The Results page is divided into two parts: *Intermediate Result* and *Final Result*, with the following results allowed: *Success*, *Warning*, *Error*, and *Aborted*.

To the right of the Results column is the *Action* column. Clicking the *Action* column allows you to set how you want to be notified for each result. Actions include sending an audit result or sending an e-mail when the result completes. If you do not select an option, no action is taken for the result.

Figure 7-13 Setting Up Who Gets Notified from the Results Tab

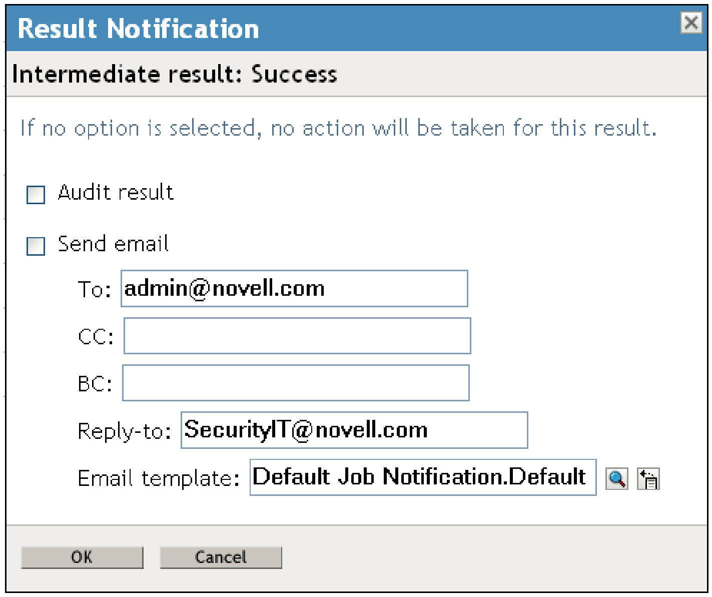


Job: Trigger1.Active Directory.IDM Driver Set.Novell

Intermediate Result	Action
Success	No action
Warning	No action
Error	No action
Aborted	No action

Final Result	Action
Success	No action
Warning	No action
Error	No action
Aborted	No action

1 Click an *Action* entry to bring up the Result Notification page.



Intermediate result: Success

If no option is selected, no action will be taken for this result.

□ Audit result

□ Send email

To: admin@novell.com

CC:

BC:

Reply-to: SecurityIT@novell.com

Email template: Default Job Notification.Default

2 Under *Send e-mail*, select who you want to send the results to by typing in the user's or group's e-mail name.

The *To*, *Reply-to*, and *Email template* fields are required for a profile. The fully distinguished name for the Email template is *Default Job Notification.Default Notification Collection.Security*.

3 When you have filled in the information, click *OK*.

4 If you want the results to go to Novell Audit, select *Audit result*.

5 Use Step 1 through Step 4 for each of the options:

- *Intermediate Success*
- *Intermediate Warning*
- *Intermediate Error*
- *Intermediate Abort*
- *Final Success*
- *Final Warning*
- *Final Error*
- *Final Abort*

If you do not select an option, no action is taken for the result.

8 Client Login Extension for Novell Identity Manager 3.5.1

The Client Login Extension for Novell® Identity Manager 3.5.1 facilitates password self-service by adding a link to the Novell and Microsoft* GINA login clients. When users click the *Forgot Password* link in their login client, the Client Login Extension launches a restricted browser to access the Identity Manager User Application Password Self-Service feature. This feature assists in reducing help desk calls from people who forgot their passwords.

Running the Client Login Extension Configuration utility for Novell Identity Manager 3.5 configures the Client Login Extension MSI file, which you then install on client workstations running the Novell Client™ software or the Microsoft GINA. The Client Login Extension works on Windows* XP and Windows 2000 workstations.

The Client Login Extension MSI files come in a number of different languages. You must configure the Client Login Extension file for each language, including English, before it can be used.

The Client Login Extension Configuration utility allows the system administrator to specify the following configuration information for the Client Login Extension MSI file:

- You can set the URL for password self services.
- For the Microsoft GINA client, you can include text (such as “Forgotten Password”) for the link to the password self-services.

NOTE: The Client Login Extension for Novell Identity Manager 3.5 works with the native Microsoft GINA and the Novell Client 4.91 SP3 or later. It does not work with any application that alters the Microsoft GINA, except the Novell Client 4.91 SP3 or later. The Client Login Extension has been tested, and is to be used, on licensed Novell Identity Manager 3.5 systems.

The following instructions step you through installing and using the Client Login Extension Configuration utility to configure the Client Login Extension MSI files. Instructions for using the Client Login Extension MSI files are also included.

8.1 Preparing to Run the Client Login Extension for Novell Identity Manager 3.5

Before running the Client Login Extension, you must have a working Identity Manager 3.5 system and have the User Application configured correctly to enable the Password Self-Service feature. (For information on installing Identity Manager 3.5 and the User Application, see the *Identity Manager 3.5.1 Installation Guide*.)

In order to have the Password Self-Service feature enabled, you need to perform at least the following:

- Enable Universal Password
- Create a password policy or select an existing password policy
- Enable and configure the Forgotten Password option
- Assign the password policy to the appropriate users, groups, or container
- Enable SSL

You initially set up the Password Self-Service feature through iManager by using the *Passwords > Password Policies > Forgotten Password* and *Policy Assignment* options. For more information on the Password Self-Service feature, see Chapter 3, “Managing Passwords by Using Password Policies” and Chapter 4, “Password Self-Service” of the *Novell Password Management Administration Guide* (http://www.novell.com/documentation/password_management31/index.html).

Use the Identity Manager User Application to complete the password configuration. For information about configuring Password Self-Service through the Identity Manager User Application, see Chapter 5.3, “Password Management Configuration” in the *Identity Manager 3.5 User Application: Administration Guide* (http://www.novell.com/documentation/idm35/index.html).

You also need to turn on SSL in JBoss*. See Chapter 2.2.2, “Self-Signed Certificates,” and 2.2.3, “Turning on SSL in JBoss” in the *Identity Manager 3.5 User Application: Administration Guide* (http://www.novell.com/documentation/idm35/index.html). The Client Login Extension does not work without SSL.

You must enable the URL rewriting when the User Application is deployed on the IBM WebSphere Application Server.

1. Log in to the Administration Console.
2. Go to the *Server > Application Servers > Select your server* (for example server1), then select *Session Management* on the Configuration tab.
3. Select the checkbox against the *Enable URL Rewriting*.
4. Click *Apply,* then click *Save*.
5. Restart the WebSphere Application Server.

With the Forgotten Password feature enabled and the password policy assigned, you now have a valid HTML link for the restricted browser to use. This link needs to be configured for HTTPS, for example `https://hostname:8443/IDM/jsps/pwdmgt/ForgotPassword.jsf`. Use this URL when running the Client Login Extension Configuration utility.

8.2 Installing the Client Login Extension Configuration Utility for Novell Identity Manager 3.5

You need to download the `idmcle.exe` file from the Novell Downloads page (http://download.novell.com/index.jsp). Under Product or Technology, select *Identity Manager* and click *search*, or type *Client Login Extension* under Keyword and click *search*. Select *Client Login Extension for Novell Identity Manager 3.5* to download, then follow the steps to download the `idmcle.exe` file.

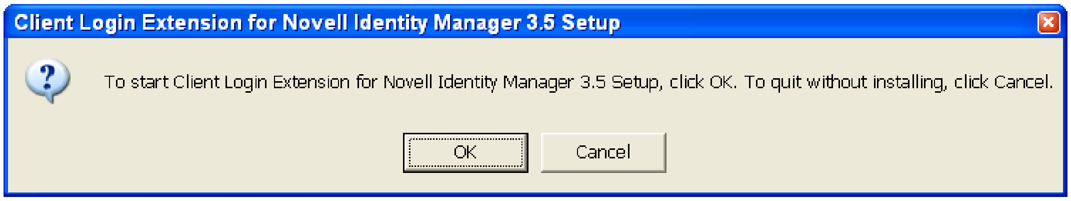
After the file is downloaded, double-click the `idmcle.exe` file to install the files for the Client Login Extension Configuration utility and the Client Login Extension MSI files. The process also installs the .NET platform if it is not already installed.

The `idmcle.exe` file creates an `idmcle` folder containing the following files:

- `ClientLoginExtensionConfigurationUtilitySetup.msi`
- `Config.ini`
- `dotnetfx.exe`
- `Settings.ini`
- `Setup.exe`

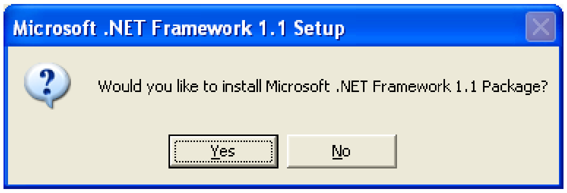
To install the Client Login Extension:

1 From the `idmcle` folder, run `Setup.exe` to launch the Client Login Extension Configuration Utility installer.



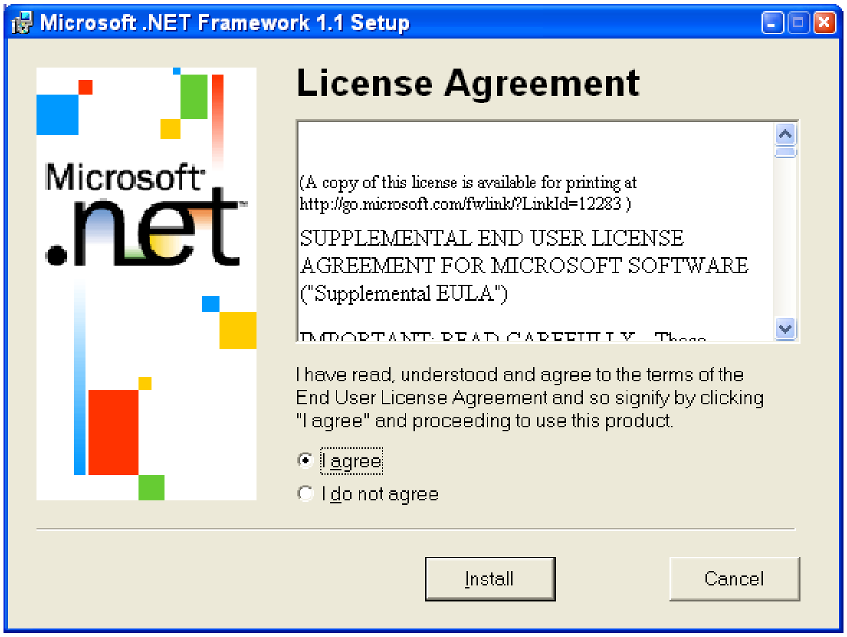
To start Client Login Extension for Novell Identity Manager 3.5 Setup, click OK. To quit without installing, click Cancel.

2 Click *OK* to continue with the installation. The installer checks to see if the .NET platform is installed on this machine. If .NET is installed, skip to Step 5. If it is not, you see the following dialog box:



Would you like to install Microsoft .NET Framework 1.1 Package?

3 Click *Yes* to install the Microsoft .NET Framework.



License Agreement

(A copy of this license is available for printing at http://go.microsoft.com/fwlink/?LinkId=12283)

SUPPLEMENTAL END USER LICENSE AGREEMENT FOR MICROSOFT SOFTWARE ("Supplemental EULA")

IMPORTANT: READ CAREFULLY - These

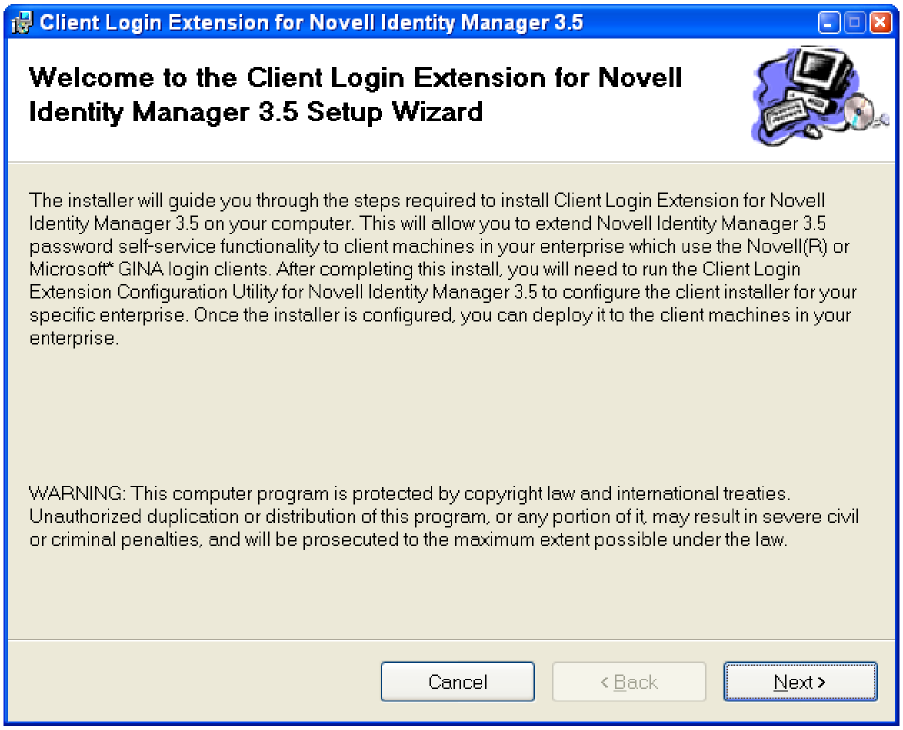
I have read, understood and agree to the terms of the End User License Agreement and so signify by clicking "I agree" and proceeding to use this product.

● I agree

○ I do not agree

4 Click *I agree*, then click *Install*.

After the .NET Framework is installed, the Client Login Extension for Novell Identity Manager 3.5 Setup Wizard launches.



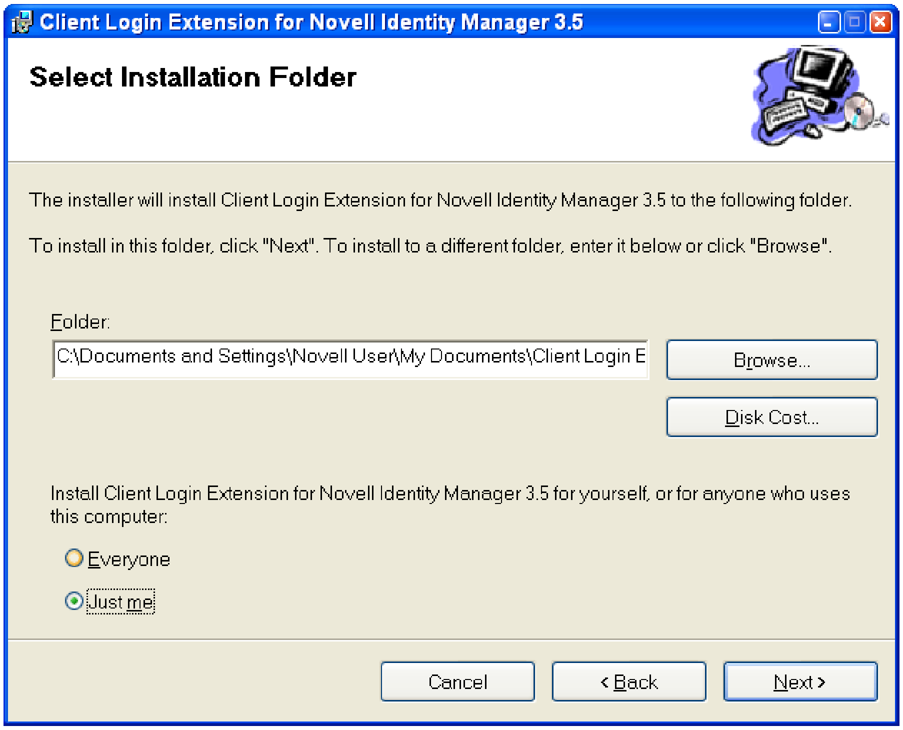
Welcome to the Client Login Extension for Novell Identity Manager 3.5 Setup Wizard

The installer will guide you through the steps required to install Client Login Extension for Novell Identity Manager 3.5 on your computer. This will allow you to extend Novell Identity Manager 3.5 password self-service functionality to client machines in your enterprise which use the Novell(R) or Microsoft* GINA login clients. After completing this install, you will need to run the Client Login Extension Configuration Utility for Novell Identity Manager 3.5 to configure the client installer for your specific enterprise. Once the installer is configured, you can deploy it to the client machines in your enterprise.

WARNING: This computer program is protected by copyright law and international treaties. Unauthorized duplication or distribution of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.

5 Read the information on the initial wizard page, then click *Next*.

6 On the License Agreement page, read the license agreement. If you agree, click *I Agree*, then click *Next*.



Client Login Extension for Novell Identity Manager 3.5

Select Installation Folder

The installer will install Client Login Extension for Novell Identity Manager 3.5 to the following folder.

To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".

Folder:
C:\Documents and Settings\Novell User\My Documents\Client Login E

Install Client Login Extension for Novell Identity Manager 3.5 for yourself, or for anyone who uses this computer:

○ Everyone

● Just me

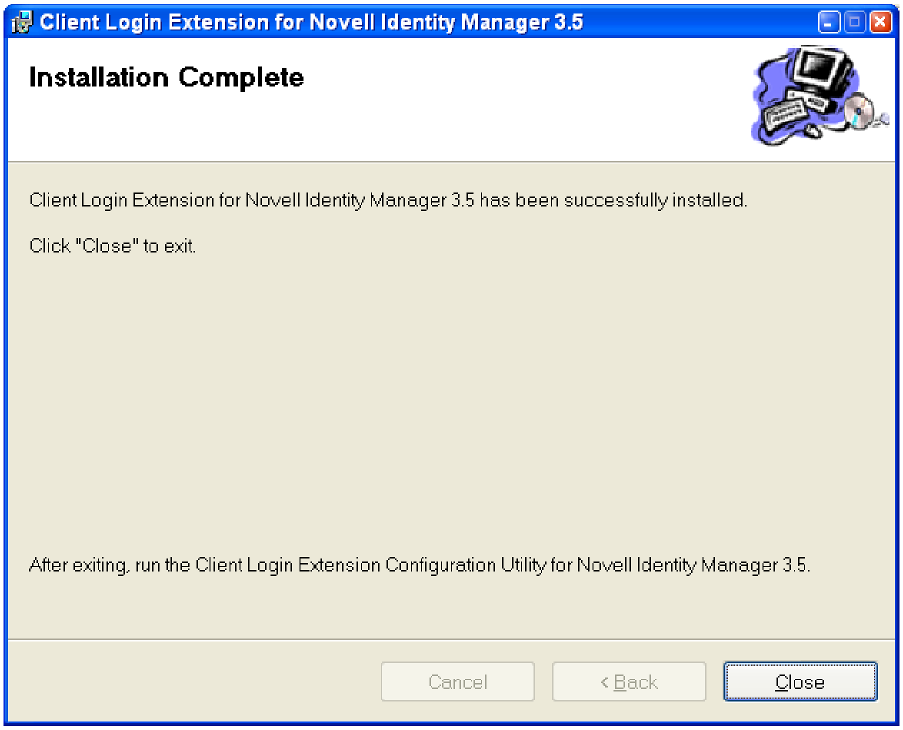
7 On the Select Installation Folder page, use the default directory selection, or click *Browse* to select a different directory.

The default directory is `C:\Documents and Settings\Username\My Documents\Client Login Extension Configuration Utility for Novell Identity Manager 3.5`.

You can also select to create shortcuts for the currently logged-in user, or for anyone who uses this computer. The default is *Just me*.

8 Click *Next*.

9 On the Confirm Installation page, click *Next* to install the Client Login Extension Configuration utility and the Client Login Extension files.



Client Login Extension for Novell Identity Manager 3.5 has been successfully installed.

Click "Close" to exit.

After exiting, run the Client Login Extension Configuration Utility for Novell Identity Manager 3.5.

10 When the installation completes, click *Close*, then click *OK*.

The install process creates two shortcuts to `ClientLoginExtensionConfigurationUtility.exe`, one for the desktop and one for the *Programs* menu. The process installs the following folders and files in the installation folder:

- `ClientLoginExtensionConfigurationUtility.exe`
- `Interop.WindowsInstaller.dll`
- `license.rtf`
- `Installer/`
  - `IdentityManagerClientLoginExtension_en.msi` (English--default)
  - `IdentityManagerClientLoginExtension_de.msi` (German)
  - `IdentityManagerClientLoginExtension_es.msi` (Spanish)
  - `IdentityManagerClientLoginExtension_fr.msi` (French)
  - `IdentityManagerClientLoginExtension_it.msi` (Italian)
  - `IdentityManagerClientLoginExtension_ja.msi` (Japanese)
  - `IdentityManagerClientLoginExtension_zh_CN.msi` (Chinese Mandarin)
  - `IdentityManagerClientLoginExtension_zh_TW.msi` (Chinese Traditional)

11 Continue with Section 8.3, “Using the Client Login Extension Configuration Utility to Configure the Client Login Extension MSI Files,” on page 249.

8.2.1 Uninstalling the Client Login Configuration Utility for Novell Identity Manager 3.5

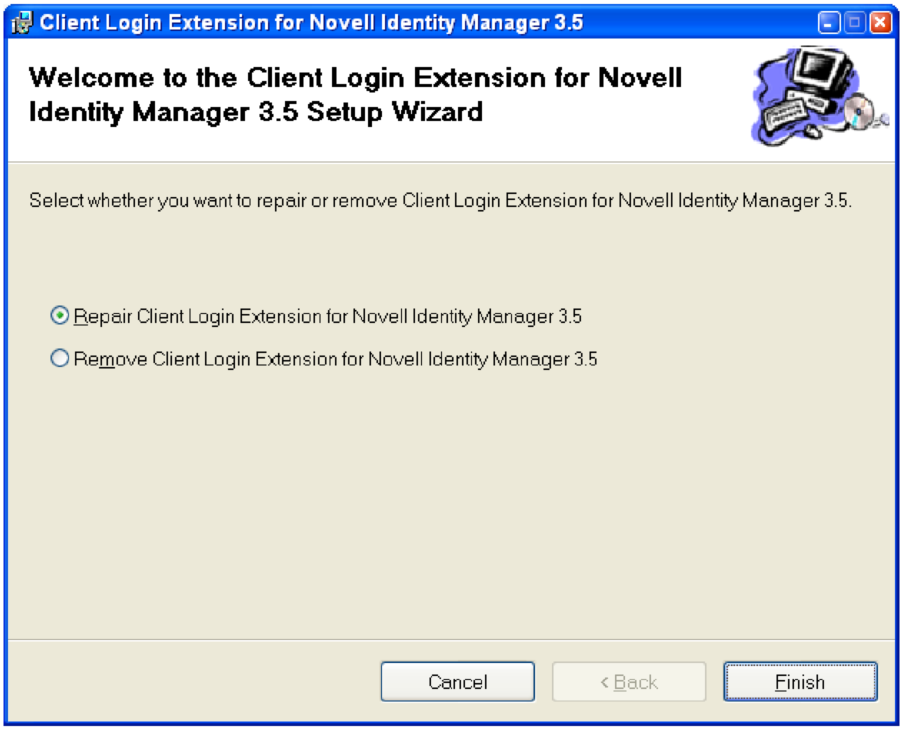
To uninstall the configuration utility:

Using Add or Remove Programs to Uninstall the Configuration Utility

1 Open the Add or Remove Programs dialog box in the Control Panel, select *Client Login Extension Configuration Utility for Novell Identity Manager 3.5*. Then click *Remove*.

Using the Setup Wizard to Uninstall the Configuration Utility

1 Re-run `Setup.exe` to re-launch the Client Login Extension for Novell Identity Manager 3.5 Setup Wizard.



Client Login Extension for Novell Identity Manager 3.5

Welcome to the Client Login Extension for Novell Identity Manager 3.5 Setup Wizard

Select whether you want to repair or remove Client Login Extension for Novell Identity Manager 3.5.

● Repair Client Login Extension for Novell Identity Manager 3.5

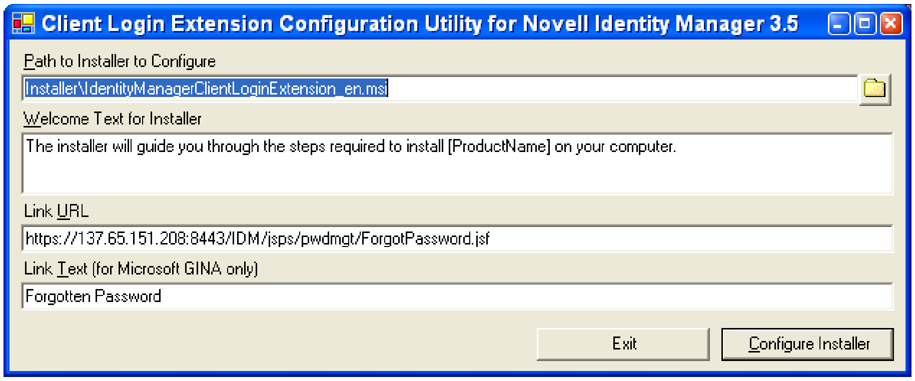
○ Remove Client Login Extension for Novell Identity Manager 3.5

2 Select the option *Remove Client Login Extension for Novell Identity Manager 3.5*, then click *Finish*.

8.3 Using the Client Login Extension Configuration Utility to Configure the Client Login Extension MSI Files

1 Click the *Client Login Extension Configuration utility for Novell Identity Manager 3.5* shortcut to launch the Client Login Extension Configuration utility.

Or you can double-click the `ClientLoginExtensionConfigurationUtility.exe` file to launch the utility.


Client Login Extension Configuration Utility for Novell Identity Manager 3.5

Path to Installer to Configure: Installer\IdentityManagerClientLoginExtension_en.msi

Welcome Text for Installer: The installer will guide you through the steps required to install [ProductName] on your computer.

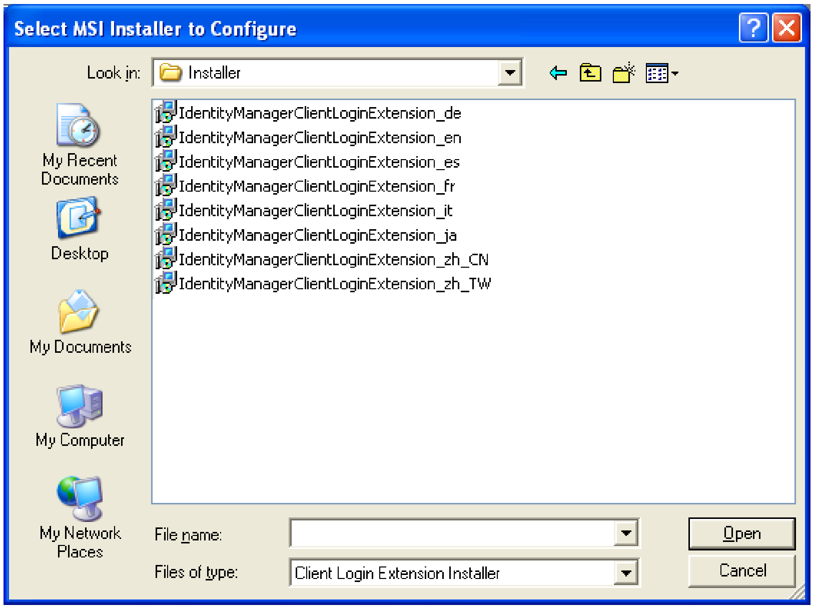
Link URL: https://137.65.151.208:8443/IDM/jsps/pwdmgt/ForgotPassword.jsf

Link Text (for Microsoft GINA only): Forgotten Password

The *Path to the Installer to Configure* option shows the path to the English version of the Client Login Extension installer file being configured. Whenever this text box contains a path to a valid MSI file, the utility automatically opens the file, populates the other controls with the information it contains, and enables the *Configure Installer* button.

2 (Optional) If you want to select another language, click the *Browse* button to select a Client Login Extension installer file in a different language.

By default, the *Browse* button opens into the `Installer` subfolder in the installation folder and displays all files that match the Client Login Extension Installer pattern.


Select MSI Installer to Configure

Look in: Installer

IdentityManagerClientLoginExtension_de
IdentityManagerClientLoginExtension_en
IdentityManagerClientLoginExtension_es
IdentityManagerClientLoginExtension_fr
IdentityManagerClientLoginExtension_it
IdentityManagerClientLoginExtension_ja
IdentityManagerClientLoginExtension_zh_CN
IdentityManagerClientLoginExtension_zh_TW

File name:

Files of type: Client Login Extension Installer

3 Modify the information in the Welcome text, or keep the information as it is presented.

The information in the text box is displayed in the Welcome screen of the Client Login Extension. The string *[ProductName]* displays as "*Client Login Extension for Novell Identity Manager 3.5*."

4 Specify the URL that the Client Login Extension restricted browser uses to connect to the User Application's Forgotten Password page. You can use either a DNS name or an IP address. An example URL using a DNS name that links to the Forgot Password page is:

```
https://hostname:8443/IDM/jsps/pwdmgt/ForgotPassword.jsf
```

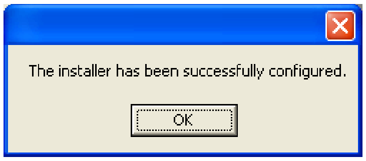
This can also be configured to access the external password WAR. To do this, see Chapter 5.8.4, "Accessing the External Password WAR," in the Identity Manager 3.5 Installation Guide (http://www.novell.com/documentation/idm35/index.html).

IMPORTANT: You must have a valid URL pointing to the User Application's Forgotten Password page; otherwise, the client connection fails and you might not be able to log in through the workstation. For more information, see Section 8.5.1, "Troubleshooting," on page 256.

5 If you have workstations that are using the Microsoft GINA, specify the text to be displayed on the link to the restricted browser that the Client Login Extension uses.

The default text is *Forgotten Password*. The text for the button in the Novell Client is provided by the Novell Client and cannot be changed here.

6 After all of the information is in place, click *Configure Installer* to write the new configuration settings to the selected Client Login Extension file.



The installer has been successfully configured.

7 Click *OK* to close the confirmation message.

8 The Client Login Extension Configuration utility remains open, allowing you to configure another Client Login Extension MSI file in a different language. To do so, click the *Browse* button to the right of the *Path to the Installer to Configure* option, select another language, and configure another `.msi` file by following Step 2 on page 250 through Step 7 on page 251.

Localized Client Login Extension MSI files for the more common languages are delivered with the configuration utility in the `Installers` folder. You must configure each localized installer individually.

To localize the Client Login Extension MSI files for languages other than those delivered with the Client Login Extension, see Section 8.3.1, "Localizing Client Login Extension Files for Other Languages," on page 251.

9 To close the Client Login Extension Configuration utility for Novell Identity Manager 3.5 window, click *Exit*.

NOTE: You cannot run the Client Login Extension MSI file while it is open in the Client Login Extension Configuration utility.

8.3.1 Localizing Client Login Extension Files for Other Languages

To localize the Client Login Extension for languages other than those delivered with the Client Login Extension Configuration utility, you can use Orca to directly edit the content of the MSI database (`IdentityManagerClientLoginExtension.msi`).

Orca (Orca.exe) (http://msdn2.microsoft.com/en-us/library/aa370557.aspx) is a database table editor for creating and editing Windows Installer packages. It is available in the Windows SDK Components for Windows Installer Developers (http://msdn2.microsoft.com/en-us/library/aa370834.aspx).

The text to be localized for `IdentityManagerClientLoginExtension.msi` is located in the following table.

Table 8-1 *Text You Need to Localize*

Table	Column	Comments
Control	Text	
Dialog	Title	
Directory	DefaultDir	Put text after "|".
Launch Condition	Description	
Property	Value	Only ProductName, Manufacturer, ARPCONTACT, and VSDVERSIONMSG.
RadioButton	Text	
Registry	Value	set LogFile, LinkURL, LinkText, PasswordComplexityText, and LoginExtDesc to defaults for the configuration utility.
Shortcut	Name	Put text after "|".
Shortcut	Description	If not Null.
UIText	Text	

WARNING: Translate only the user interface text. For example, do not translate text that is surrounded by square brackets ([xxxx]), or is in mixed case (XxxXxxXxx). Modifying these property names and identifiers breaks the installer.

Use the following procedure to localize the Client Login Extension MSI file to a new language:

1. Copy `IdentityManagerClientLoginExtension.msi` to `IdentityManagerClientLoginExtension_`*`xx`*`.msi`, where *xx* identifies the new language (locale).
2. Open `IdentityManagerClientLoginExtension_`*`xx`*`.msi` in `Orca.exe`, edit the tables and columns as they are listed in Table 8-1 on page 252 to insert the localized text, then save and close the file.
3. Open `IdentityManagerClientLoginExtension_`*`xx`*`.msi` with the Client Login Extension Configuration utility (`ClientLoginExtensionConfigurationUtility.exe`), review the default values, make any modifications if needed, then click *Configure Installer*.

NOTE: Step 3 is required, even if the default values that you set in the Registry table do not need modification. The Client Login Extension Configuration utility makes additional changes that enable the Client Login Extension MSI file.

8.4 Installing the Client Login Extension MSI File

With the Client Login Extension MSI file configured, you can distribute the `IdentityManagerClientLoginExtension_xx.msi` file (or its distribution name) to users or to a distribution mechanism. The *xx* identifies the language (locale). The `IdentityManagerClientLoginExtension_xx.msi` file is located in the `Installer` folder.

You must also install the Microsoft .NET Framework on every workstation that you plan to run the Client Login Extension MSI file. For version consistency, use the `dotnetfx.exe` file that comes with the Client Login Extension feature and is found in the `idmcle` folder.

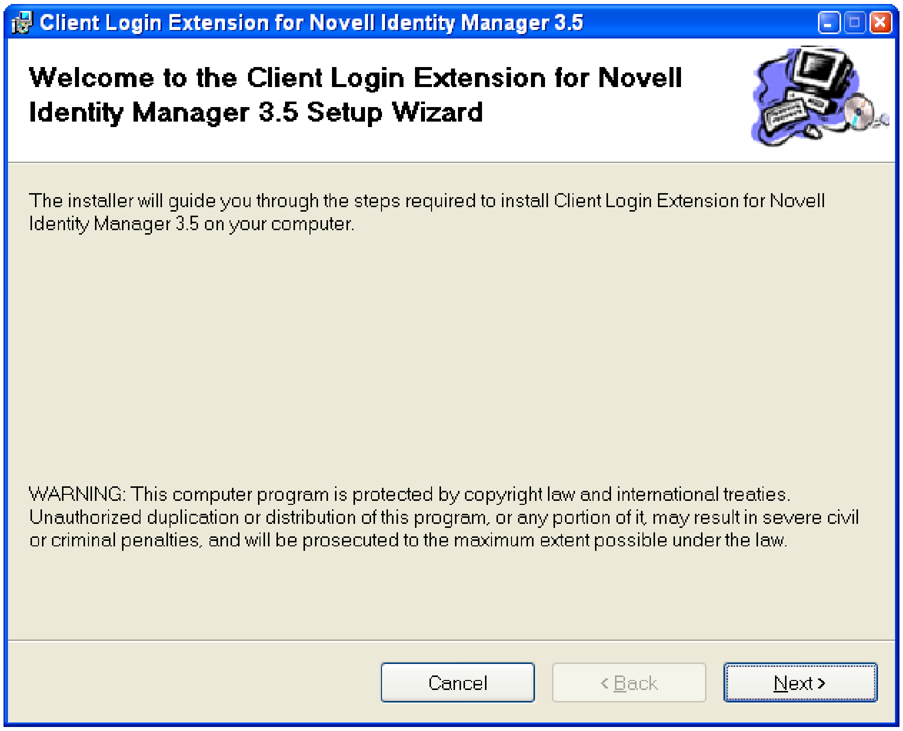
NOTE: The Client Login Extension only works with the native Microsoft GINA and the Novell Client 4.91 SP3. It does not work with any application that alters the Microsoft GINA, except the Novell Client 4.91 SP3. The Client Login Extension works on Windows XP and Windows 2000 workstations.

---

1 Run the `dotnetfx.exe` file to install the Microsoft .NET Framework on each workstation that you plan to run the Client Login Extension MSI file. See Step 2 under Section 8.2, "Installing the Client Login Extension Configuration Utility for Novell Identity Manager 3.5," on page 245.

2 After the Microsoft .NET Framework is installed, double-click the `IdentityManagerClientLoginExtension_xx.msi` file to launch the Client Login Extension welcome page.

The welcome message is the text you provided in the Client Login Extension Configuration utility.



Client Login Extension for Novell Identity Manager 3.5

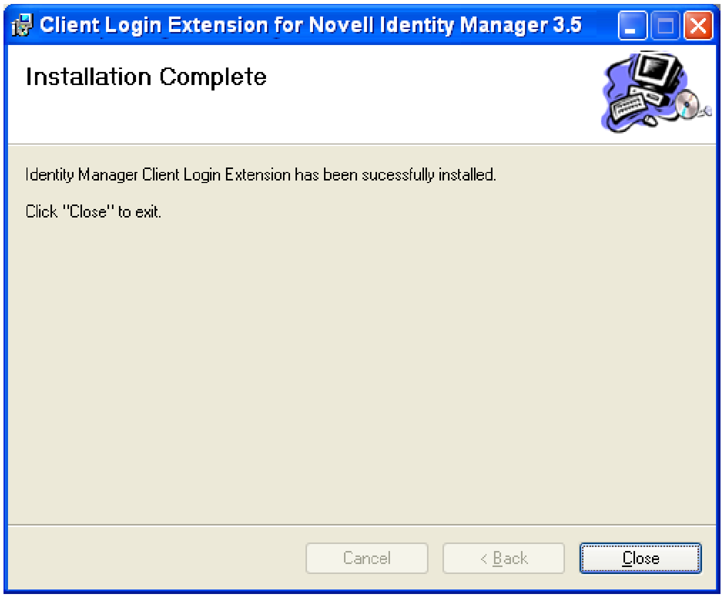
Welcome to the Client Login Extension for Novell Identity Manager 3.5 Setup Wizard

The installer will guide you through the steps required to install Client Login Extension for Novell Identity Manager 3.5 on your computer.

WARNING: This computer program is protected by copyright law and international treaties. Unauthorized duplication or distribution of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.

3 Click *Next* to start the installation.

4 After the Client Login Extension is installed, click *Close*.



Installation Complete

Identity Manager Client Login Extension has been sucessfully installed.

Click "Close" to exit.

5 (Optional) To uninstall the Client Login Extension, open the Add or Remove Programs dialog box in the Control Panel, select *Client Login Extension for Novell Identity Manager 3.5*, then click *Remove*.

8.4.1 Using the Client Login Extension Installer Command Line Options

The Client Login Extension MSI file is a standard MSI installer. It can be used with any of the standard `Msiexec.exe` command line options, which you can find at msdn (http://msdn2.microsoft.com/en-us/library/aa367988.aspx). Some examples are shown below.

To install the Client Login Extension MSI file with no user interface, enter the following at the command line:

```
msiexec /i IdentityManagerClientLoginExtension_en.msi /q
```

or

```
IdentityManagerClientLoginExtension_en.msi /q
```

To install with no user interface except for a modal dialog box displayed at the end, enter:

```
msiexec /i IdentityManagerClientLoginExtension_en.msi /qn+
```

or

```
IdentityManagerClientLoginExtension_en.msi /qn+
```

To uninstall with no user interface, enter:

```
msiexec /x IdentityManagerClientLoginExtension_en.msi /q
```

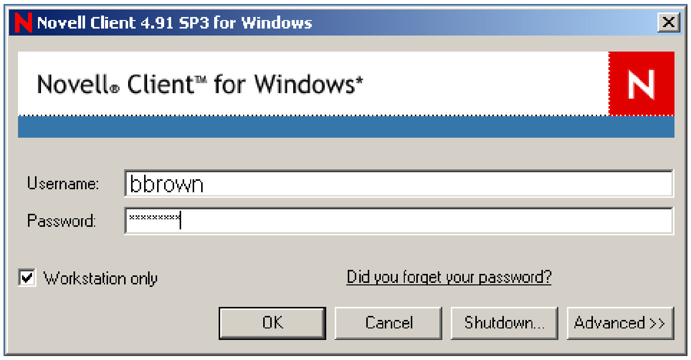
To uninstall with no user interface except for a modal dialog box displayed at the end, enter:

```
msiexec /x IdentityManagerClientLoginExtension_en.msi /qn+
```

8.5 Using the Forgotten Password Feature

After you run the Client Login Extension MSI file on workstations running the Novell Client 4.91 SP3 or the Microsoft GINA, and you have specified a valid HTTPS link, the Password Self-Service feature is now ready to use. (See Section 8.1, “Preparing to Run the Client Login Extension for Novell Identity Manager 3.5,” on page 244 to ensure that you have all the information in place for Password Self-Service to work.)

1 If you forget your password, click the *Did you forget your password?* link in the Novell Client.


Novell Client 4.91 SP3 for Windows
Novell® Client™ for Windows*
N
Did you forget your password?
OK
Cancel
Shutdown...
Advanced >>

Username: bbrown

Password: *********

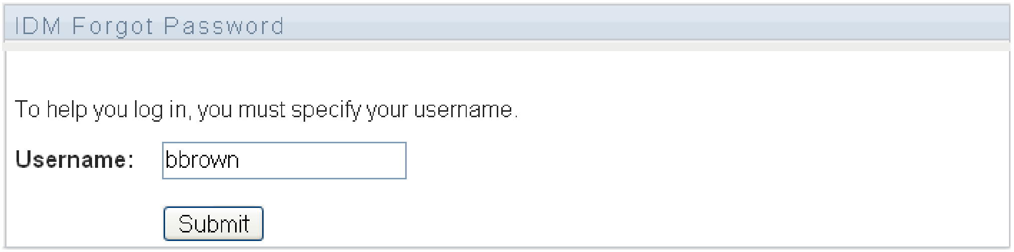
☑ Workstation only

The Microsoft GINA link says *Forgotten Password*, or whatever text you provided when configuring the Client Login Extension.

Clicking the *Did you forget your password?* link launches the restricted browser that can only go to the URL designated in the Client Login Extension Configuration utility. The restricted browser performs the following:

- Verifies that the protocol is HTTPS
- Validates the hostname
- Verifies that the target Web site is operating in Internet Explorer’s restricted sites zone
- Disables hotkeys
- Disables tabs
- Disables right-clicking
- Disables ActiveX*
- Disables scripts
- Runs on its own process, separate from the Winlogon process

2 After the restricted browser connects to the Forgotten Password page, you see the IDM Forgot Password dialog box. Type your login name and click *Submit*.

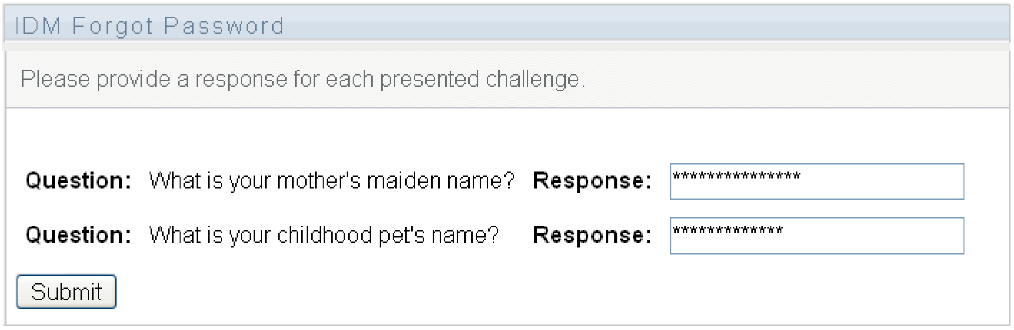


To help you log in, you must specify your username.

Username: bbrown

What you see in the IDM Forgot Password dialog box depends on how the system administrator has set up the Forgotten Password option. You can see a hint, have your hint e-mailed to you, have your password e-mailed to you, or you can be allowed to change your password. You can also be given challenge questions.

For this example, the user is given a challenge question and a hint.



IDM Forgot Password

Please provide a response for each presented challenge.

Question: What is your mother's maiden name? **Response:** ****************

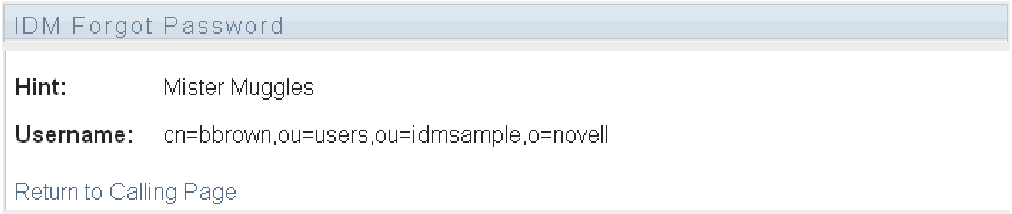
Question: What is your childhood pet's name? **Response:** **************

3 Type your response to the questions and click *Submit*.

The number of response questions and what they say is configurable by the system administrator.

If you do not answer the questions correctly, you see a *Challenge Response failed* message and you are presented with the questions again.

4 After the response questions are answered correctly, you are presented with the password hint, depending on how the system administrator has configured password self-service.



IDM Forgot Password

Hint: Mister Muggles

Username: cn=bbrown,ou=users,ou=idmsample,o=novell

5 Close the browser (because the browser is restricted, the *Return to Call Page* link won't work.) Use the hint to remember your password. If you still cannot remember your password, contact your system administrator.

8.5.1 Troubleshooting

Keep in mind the following information as you use the Forgotten Password feature:

- If your system administrator allows you to change your password through this process, it can take up to 15 minutes or longer before all changes are in place throughout the network. Be patient before contacting your system administrator.
- For those using the Novell client, if you are already logged in to the network and right-click the red N in the taskbar, select *NetWare Login*, then select the *Did you forget your password?* link, you will not launch the restricted browser. The Client Login Extension only applies when you have not logged in.

- If the server running the Identity Manager User Application is down and you select the *Did you forget your password?* link, you receive the message “*An error has occurred*” in red on the restricted browser’s initial page. Contact your system administrator.
- If the server running the Identity Manager external WAR is down and you select the *Did you forget your password?* link, you receive the message “*Page Not Found*” on the restricted browser’s initial page. Contact your system administrator.
- If the URL to the IDM Forgot Password page is mis-configured and you select the *Did you forget your password?* link, you receive the message “*Page Not Found*” on the restricted browser’s initial page. Contact your system administrator.

Security: Best Practices 9

9.1 Using SSL

Enable SSL for all transports, where it is available. Enable SSL for communication between the Metadirectory engine and Remote Loader (see Section 3.2, “Providing for Secure Data Transfers,” on page 55), and between the Metadirectory engine or Remote Loader and the connected systems.

If you don't enable SSL, you are sending sensitive information such as passwords in the clear.

9.2 Securing Access

Make sure that you secure access to Identity Vaults and to Identity Manager objects.

Physical Security: Protect access to the physical location of the servers where an Identity Vault is installed.

File System Access: The security of the file system for Identity Manager is critical to ensuring the security of the system as a whole. Verify that the directories containing eDirectory, the Metadirectory engine, and the Remote Loader are accessible only to the appropriate administrators.

There is an issue with the file system, when the Remote Loader is installed on a Windows 2000 server. For more information, see TID# 3243550, Securing a Remote Loader Install on a Microsoft Windows 2000 Server (http://www.novell.com/support/search.do?cmd=displayKC&docType=kc&externalId=3243550&sliceId=SAL_Public&dialogID=47824778&stateId=0%200%2047832401).

Access Rights: Identity Manager requires Administrative rights to create Identity Manager objects and configure drivers. Monitor and control who has rights to create or modify the following:

- An Identity Manager driver set
- An Identity Manager driver
- Driver configuration objects (filters, style sheets, policies), especially policies that are used for password retrieval or synchronization
- Password policy objects (and the iManager task for editing them), because they control which passwords are synchronized to each other, and which Password Self-Service options are used

9.2.1 Granting Task-Based Access to Drivers and Driver Sets

In addition to the eDirectory's standard object-based access controls, Identity Manager lets you assign trustee rights to perform only certain tasks on an Identity Manager driver, rather than just granting full Supervisor rights to the driver object. For example, you can assign trustee rights such that a user can only configure the driver object (create and modify object properties), while another user can only start and stop the driver.

Identity Manager provides the following driver object attributes that enable role-based access:

Attribute	Description
DirXML-AccessRun	Start and stop Identity Manager drivers and jobs
DirXML-AccessMigrate	Manage migrate operations into the Identity Vault
DirXML-AccessSubmitCommand	Manage the driver's pass-through commands
DirXML-AccessCheckObjectPassword	Manage the driver's "check object password" commands
DirXML-AccessConfigure	Manage the driver's and job configuration
DirXML-AccessManage	View and modify the driver's cache file contents

Setting trustee rights to these attributes grants access to the associated Identity Manager verbs and sub-verbs. Read access lets users view state (get verb state), and Write access lets users modify or change state (set verb state.) For example, granting Read access to a driver object's DirXML-AccessRun attribute lets the user get the driver state (started or stopped.) Granting Write access lets the user set the driver state (change from started to stopped, or vice-versa.)

The goal of providing this attribute-based access to driver tasks is to let you create well-defined administrative roles, perhaps using eDirectory's Administrative Role object, that let users perform certain management tasks without exposing all management functionality. Creating these roles can go beyond providing access to the DirXML-Access attributes described above and can include access rights to other attributes, as well as access to other Identity Manager objects. The following examples demonstrate the flexibility available for creating administrative roles.

Start/Stop Driver Admin This administrative role lets the assigned user Start and Stop all drivers in a given driver set. It requires the following access rights:

- Browse rights to the Driver Set object
- Read and Write access, with inheritance, to the DirXML-AccessRun attribute of the Driver Set object

Driver Admin This administrative role lets the assigned user manage a single Driver object. It requires the following access rights:

- Browse and Create rights to the Driver object
- Read and Write access to [All Attribute Rights] in the Driver object

NOTE: Make sure the rights are inherited so the Driver Admin can also manage the driver's policy objects.

Information about using iManager to grant eDirectory access rights is available in the iManager Administration Guide (http://www.novell.com/documentation/imanager26/imanager_admin_26/data/bu0906q.html).

9.3 Managing Passwords

When you choose to exchange information between connected systems, you should take precautions to make sure that the exchange is secure. This is especially true for passwords.

- The Password Hint attribute (nsimHint) is also publicly readable, to allow unauthenticated users who have forgotten a password to access their own hint. Password Hints can help reduce help desk calls.

  For security, Password Hints are checked to make sure that they do not contain the user's actual password. However, a user could still create a Password Hint that gives too much information about the password.

  To increase security when using Password Hints:

  - Allow access to the nsimHint attribute only on the LDAP server used for Password Self-Service.
  - Require that users answer Challenge Questions before receiving the Password Hint.
  - Remind users to create Password Hints that only they would understand. The Password Change Message in the password policy is one way to do that. See “Adding a Password Change Message” in the Password Management Administration Guide (http://www.novell.com/documentation/password_management/index.html).

  If you choose not to use Password Hint at all, make sure you don't use it in any of the password policies. To prevent Password Hints from being set, you can go a step further and remove the Hint Setup gadget completely, as described in “Disabling Password Hint by Removing the Hint Gadget” in the Password Management Administration Guide (http://www.novell.com/documentation/password_management/index.html).

- Challenge Questions are publicly readable, to allow unauthenticated users who have forgotten a password to authenticate another way. Requiring Challenge Questions increases the security of Forgotten Password Self-Service, because a user must prove his or her identity by giving the correct responses before receiving a forgotten password or a Password Hint, or resetting a password.

  The intruder lockout setting is enforced for Challenge Questions, so the number of incorrect attempts an intruder could make is limited.

  However, a user could create Challenge Questions that hold clues to the password. Remind users to create Challenge Questions and Responses that only they would understand. The Password Change Message in the password policy is one way to do that. See “Adding a Password Change Message” in the Password Management Administration Guide (http://www.novell.com/documentation/password_management/index.html).

- For security, the Forgotten Password actions of *E-mail password to user* and *Allow user to reset password* are available only if you require the user to answer Challenge Questions.

- A security enhancement was added to NMAS™ 2.3.4 regarding Universal Passwords changed by an administrator. It works basically the same way as the feature previously provided for NDS® Password.

  If an administrator changes a user's password, such as when creating a new user or in response to a help desk call, the password is automatically expired if you have enabled the setting to expire passwords in the password policy. The setting in the password policy is in Advanced Password Rules, named *Number of days before password expires (0-365)*. For this particular feature, the number of days is not important, but the setting must be enabled.

9.4 Creating Strong Password Policies

Password policy objects are publicly readable to allow applications to check whether passwords are compliant. This means that an unauthenticated user could query an Identity Vault and find out what password policies are in place. If the password policies require users to create strong passwords, this should not pose a risk, as noted in “Create Strong Password Policies” in the *Password Management Administration Guide* (http://www.novell.com/documentation/password_management/index.html).

Identity Manager Password Synchronization lets you simplify user passwords and reduce help desk costs. Bidirectional password synchronization lets you share passwords among eDirectory and connected systems in multiple ways, as described in the scenarios in Section 5.8, “Implementing Password Synchronization,” on page 124.

Using Universal Password and password policies allows you to enforce strong password syntax requirements for users. Use the Advanced Password Rules in password policies to define your organization’s best practices for passwords. The Advanced Password Rules features let you manage password syntax using either Novell syntax or the Microsoft Complexity Policy. For more information, see “Managing Passwords by Using Password Policies” in the *Novell Password Management 3.1 Administration Guide* (http://www.novell.com/documentation/password_management31/pwm_administration/data/ampxjj0.html).

For example, using Novell password syntax options, you can require user passwords to comply with rules such as the following:

- Requiring unique passwords.

  You can prevent users from reusing passwords, and control the number of passwords the system should store in the history list for comparison
- Requiring a minimum number of characters in password.

  Requiring longer passwords is one of the best ways to make passwords stronger.
- Requiring a minimum number of numerals in password.

  Requiring at least one numeric character in a password helps protect against “dictionary attacks,” in which intruders try to log in using words in the dictionary.
- Excluding passwords of your choice.

  You can exclude words that you consider to be security risks, such as the company name or location, or the words test or admin. Although the exclusion list is not meant to import an entire dictionary, the list of words you exclude can be quite long. Just keep in mind that a long list of exclusions makes login slower for your users. A better protection from dictionary attacks is probably to require numerals or special characters.

Keep in mind that you can create multiple password policies if you have different password requirements in different parts of the tree. You can assign a password policy to the whole tree, a partition root container, container, or even an individual user. (To simplify administration, we recommend you assign password policies as high up in the tree as possible.)

In addition, you can use intruder lockout. As always, this eDirectory feature lets you specify how many failed login attempts are allowed before an account is locked. This is a setting on the parent container instead of in the password policy. See “Managing User Accounts” in the *Novell eDirectory Administration Guide* (http://www.novell.com/documentation/edir88/index.html?page=/documentation/edir88/edir88/data/afxkmdi.html).

9.5 Securing Connected Systems

Keep in mind that the connected systems that you are synchronizing data to might store or transport that data in a compromising manner.

Secure the systems to which you exchange passwords. For example LDAP, NIS, and Windows each have security concerns that you must consider before enabling password synchronization with those systems.

Many software vendors provide specific security guidelines that you should follow for their products.

9.5.1 Password Generation

Identity Manager 3.5 includes a pre-defined password generation job for the Job Scheduler. The password generation job generates random passwords for a group of User objects in eDirectory, either periodically or on demand. This functionality is designed primarily to support products like Novell Certificate Login, but can also be used in other situations.

Invoking the password generation job initializes NMAS with the password policy, and the following occurs for each object in the specified job scope:

1. NMAS generates a random password consistent with the password policy specified in the job. Password policies are stored in `nspmPasswordPolicy` objects. Typically, each connected system has its own policy object. These policy objects can be stored in `DirXML-Driver` and `DirXML-DriverSet` objects.
2. Each generated password i submitted, one at a time, to the containing driver’s subscriber channel.

   If the object has a non-disabled association for the driver then a `<generated-password>` event is submitted to the subscriber channel event queue (cache) of the driver.

   If the object has no association for the driver and the option to submit events for non-associated objects is selected then a `<generated-password>` event is submitted to the subscriber channel event queue (cache) of the driver.
3. It is up to the subscriber channel policies to handle the generated passwords. Job Scheduler is only responsible for generating the passwords and handing them off to the subscriber channel.

9.6 Designer for Identity Manager

When using Designer for Identity Manager, consider the following issues:

- Monitor and control who has rights to create or modify an Identity Manager driver.

  Administrative rights are needed to create Identity Manager objects and configure drivers.
- Before giving a consultant an Identity Vault administrator password, limit the rights assigned to that administrator to areas of the tree that the consultant must access.
- Delete the project files (`.proj`) or save them to a company directory.

  Designer `.proj` files are to remain at the company's project site. A consultant does not take the files after completing a project.
- After project files, log files, and trace files are no longer needed, delete them.
- Before discarding or surplusing a laptop, verify that project files have been cleaned.
- Ensure that the connection from Designer to the Identity Vault server is physically secure.

  Otherwise, someone could monitor the wire and pull sensitive information.
- When you create documents by using Document Generator, be careful with those documents.

  These documents can contain passwords and sensitive data in clear text.
- If Designer needs to read or write to an eDirectory attribute, do not mark that attribute as encrypted.

  Designer is unable to read or write to encrypted attributes.
- Do not store passwords that are sensitive.

  Currently, Designer projects are not encrypted. Passwords are only encoded.Therefore, do not share Designer projects that have saved passwords.

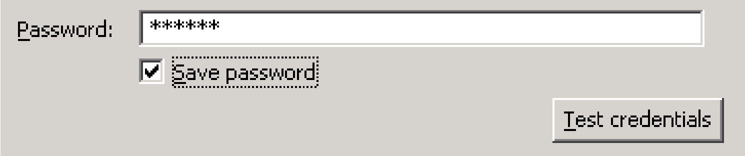
  To save a password for a session, but not save it to the project:

  1. In an expanded Outline view, right-click an Identity Vault.
  2. Select Properties.
  3. On the Configuration page, type a password, then click *OK*.

     You can enter a password once per session. After you close the project, the password is lost.

  To save a password to the hard drive, complete Steps 1-3, select *Save Password*, then click *OK*.

Figure 9-1 *Save Password*



Password: ******

☑ Save password

9.7 Industry Best Practices for Security

Follow industry best practices for security measures, such as blocking unused ports on the server.

9.8 Tracking Changes to Sensitive Information

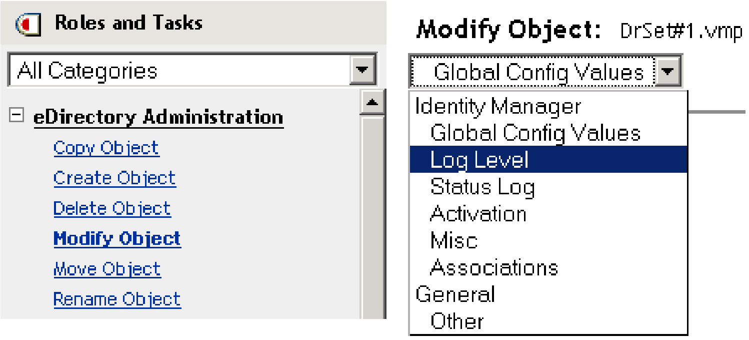
- Section 9.8.1, “Logging Events by Using iManager,” on page 265
- Section 9.8.2, “Logging Events by Using Designer,” on page 266

9.8.1 Logging Events by Using iManager

You can use Novell Audit to log events that you consider important for security.

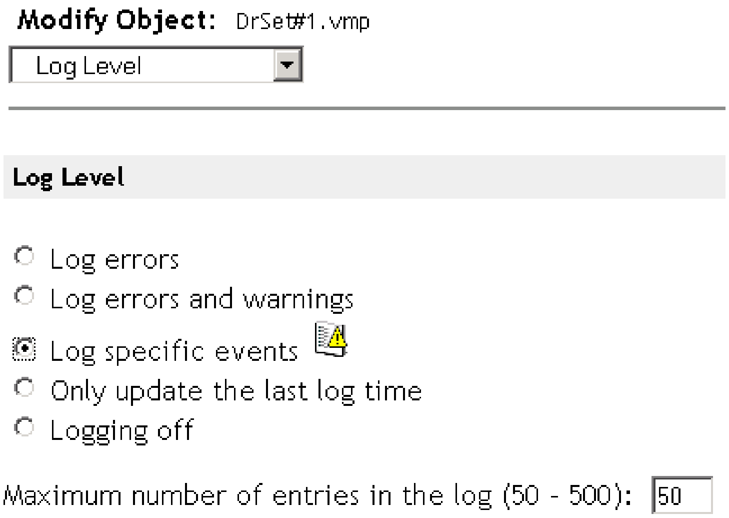
For example, you could log password changes for a particular Identity Manager driver (or driver set) by doing the following:

1 Select *eDirectory Administration > Modify Object > Log Level*.



Select from the drop-down list or select a tab, depending on your version of iManager.

2 Select *Log Specific Events*.



Modify Object: DrSet#1.vmp

Log Level

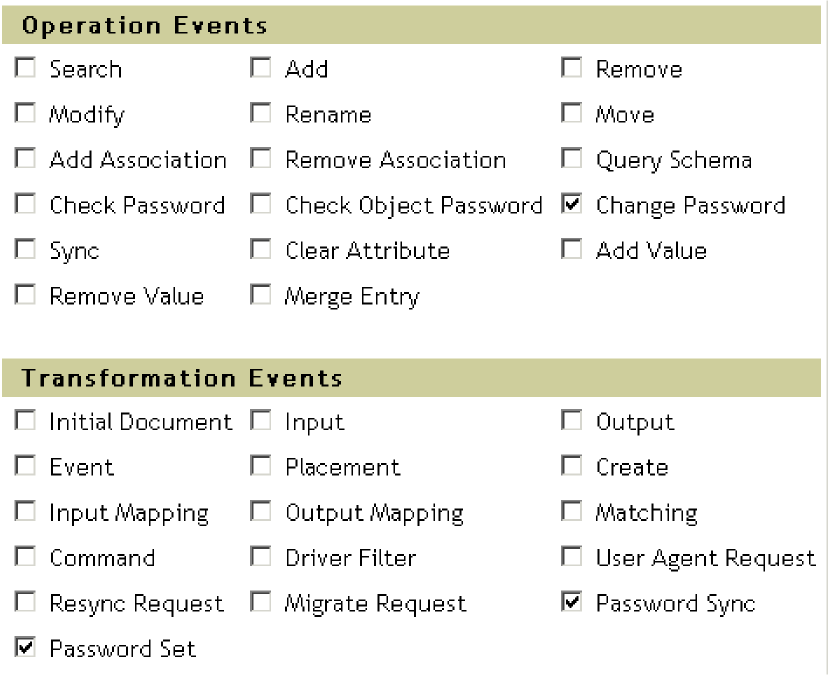
Log Level

○ Log errors
○ Log errors and warnings
● Log specific events
○ Only update the last log time
○ Logging off

Maximum number of entries in the log (50 - 500): 50

3 To select the specific events, click the log events icon .

4 On the Events page, select the following:



Operation Events

☐ Search ☐ Add ☐ Remove
☐ Modify ☐ Rename ☐ Move
☐ Add Association ☐ Remove Association ☐ Query Schema
☐ Check Password ☐ Check Object Password ☑ Change Password
☐ Sync ☐ Clear Attribute ☐ Add Value
☐ Remove Value ☐ Merge Entry

Transformation Events

☐ Initial Document ☐ Input ☐ Output
☐ Event ☐ Placement ☐ Create
☐ Input Mapping ☐ Output Mapping ☐ Matching
☐ Command ☐ Driver Filter ☐ User Agent Request
☐ Resync Request ☐ Migrate Request ☑ Password Sync
☑ Password Set

- In Operation Events, select *Change Password*.

  This item monitors direct changes to the NDS password.
- In Transformation Events, select *Password Se*t and *Password Sync*. These two items monitor events for the Universal Password and Distribution Password.

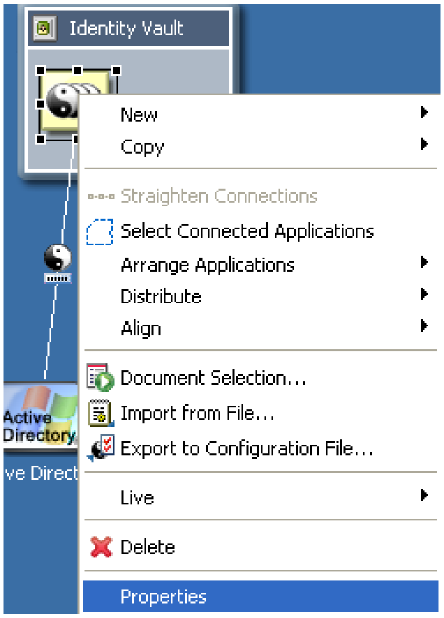
5 Click *OK* twice.

9.8.2 Logging Events by Using Designer

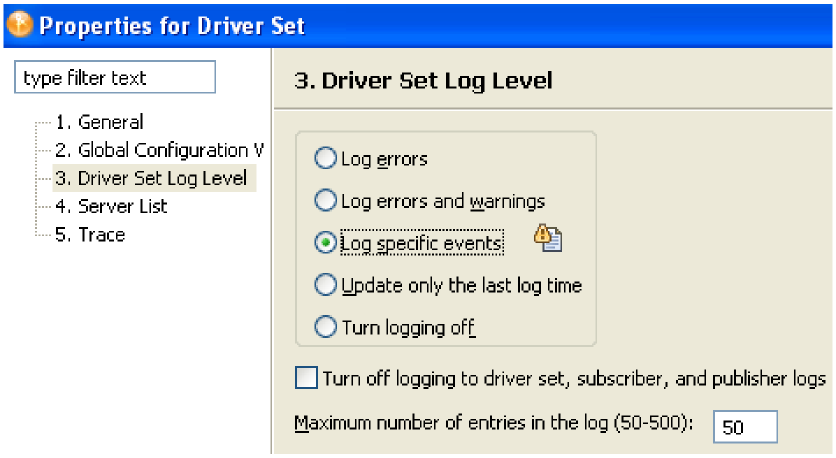
You can log events that apply to a driver set or to a driver.

Logging Events for a Driver Set

1 In Designer, right-click a driver set, then select *Properties*.



2 Select *Driver Set Log Level*, then select *Log Specific Events*.



Properties for Driver Set

type filter text

- 1. General
- 2. Global Configuration V
- 3. Driver Set Log Level
- 4. Server List
- 5. Trace

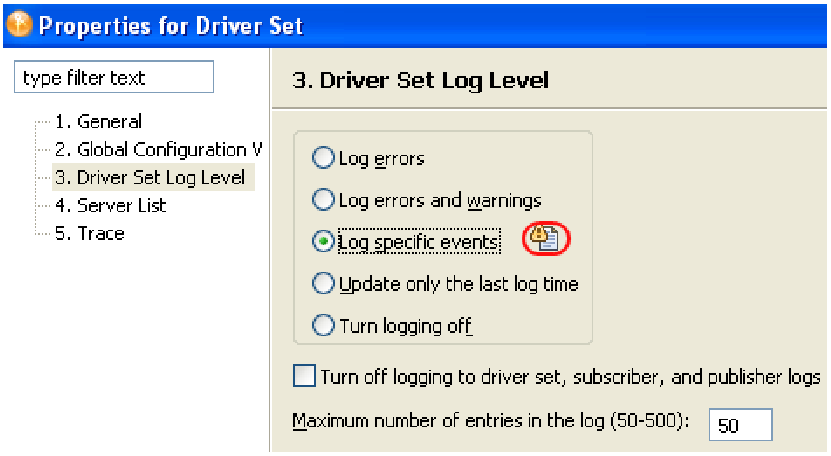
3. Driver Set Log Level

○ Log errors
○ Log errors and warnings
● Log specific events
○ Update only the last log time
○ Turn logging off

□ Turn off logging to driver set, subscriber, and publisher logs

Maximum number of entries in the log (50-500): 50

3 Click *Select Events to Log* icon.



Properties for Driver Set

type filter text

- 1. General
- 2. Global Configuration V
- 3. Driver Set Log Level
- 4. Server List
- 5. Trace

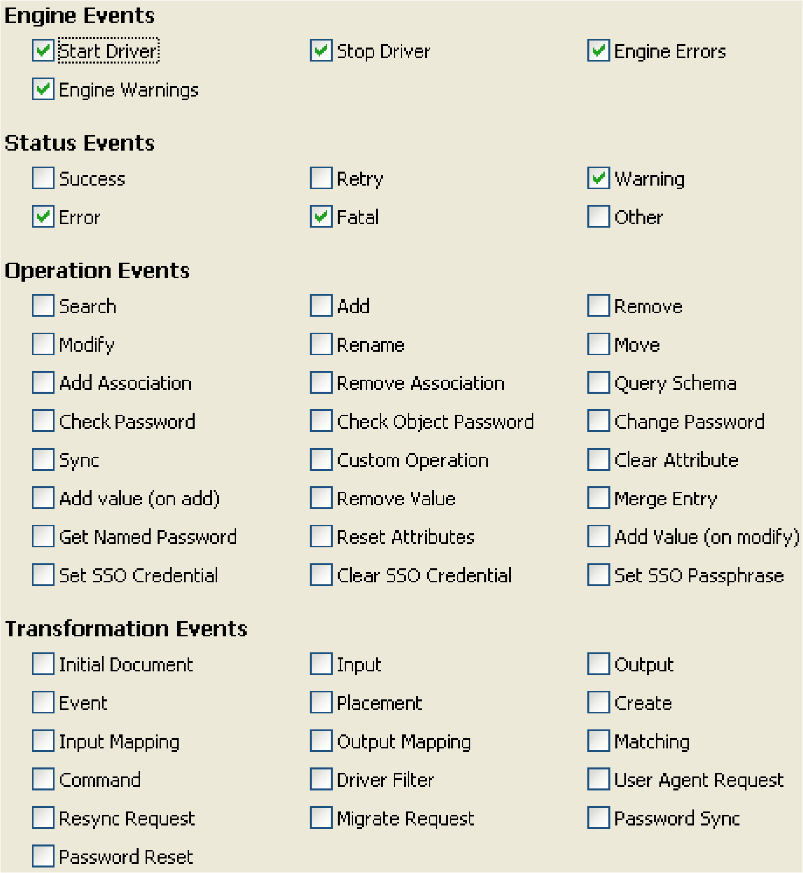
3. Driver Set Log Level

○ Log errors
○ Log errors and warnings
● Log specific events
○ Update only the last log time
○ Turn logging off

□ Turn off logging to driver set, subscriber, and publisher logs

Maximum number of entries in the log (50-500): 50

4 Select events to log, then click *OK*.



Engine Events

☑ Start Driver ☑ Stop Driver ☑ Engine Errors

☑ Engine Warnings

Status Events

□ Success □ Retry ☑ Warning

☑ Error ☑ Fatal □ Other

Operation Events

□ Search □ Add □ Remove

□ Modify □ Rename □ Move

□ Add Association □ Remove Association □ Query Schema

□ Check Password □ Check Object Password □ Change Password

□ Sync □ Custom Operation □ Clear Attribute

□ Add value (on add) □ Remove Value □ Merge Entry

□ Get Named Password □ Reset Attributes □ Add Value (on modify)

□ Set SSO Credential □ Clear SSO Credential □ Set SSO Passphrase

Transformation Events

□ Initial Document □ Input □ Output

□ Event □ Placement □ Create

□ Input Mapping □ Output Mapping □ Matching

□ Command □ Driver Filter □ User Agent Request

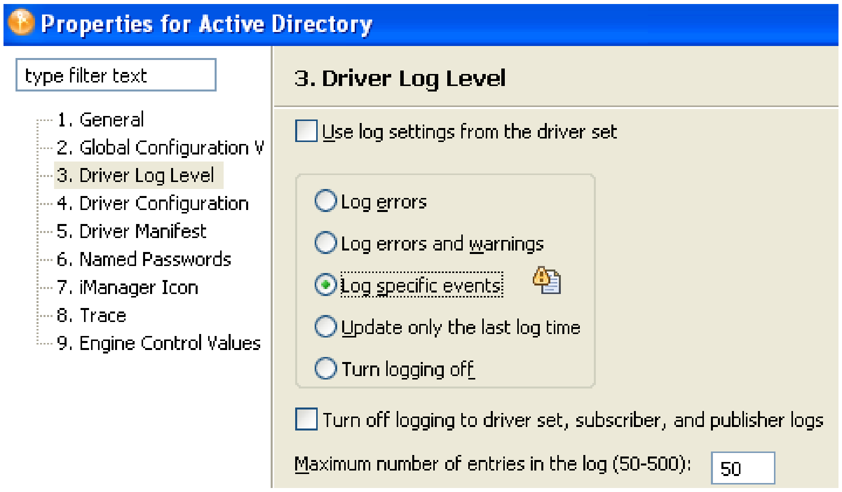
□ Resync Request □ Migrate Request □ Password Sync

□ Password Reset

Logging Events for a Driver

1 In Designer, right-click a driver, then select *Properties*.

2 Select *Driver Log Level*, then select *Log Specific Events*.



3. Driver Log Level

□ Use log settings from the driver set

○ Log errors

○ Log errors and warnings

● Log specific events

○ Update only the last log time

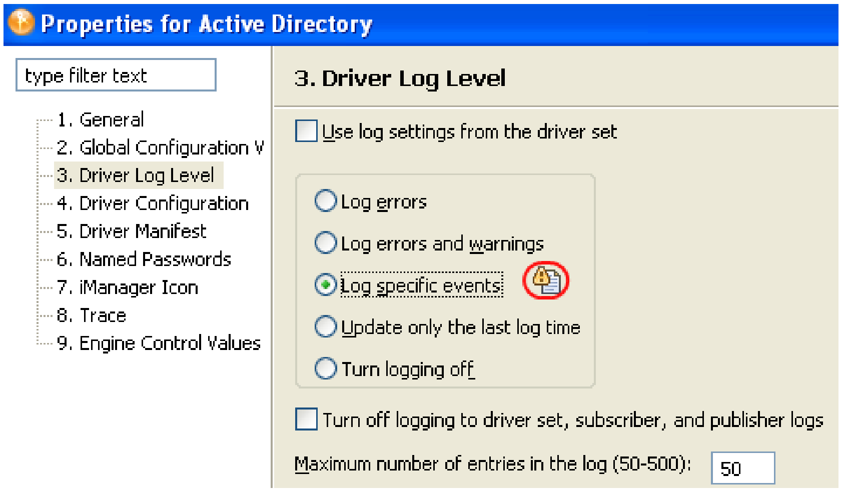
○ Turn logging off

□ Turn off logging to driver set, subscriber, and publisher logs

Maximum number of entries in the log (50-500): 50

If you prefer, you can accept the settings for the driver set, then click *OK*. Otherwise, deselect *Use log settings from the Driver Set*, select *Log specific events*, then click *OK*.

3 Click the *Select Events to Log* icon.



Properties for Active Directory

type filter text

- 1. General
- 2. Global Configuration V
- 3. Driver Log Level
- 4. Driver Configuration
- 5. Driver Manifest
- 6. Named Passwords
- 7. iManager Icon
- 8. Trace
- 9. Engine Control Values

3. Driver Log Level

□ Use log settings from the driver set

○ Log errors

○ Log errors and warnings

● Log specific events

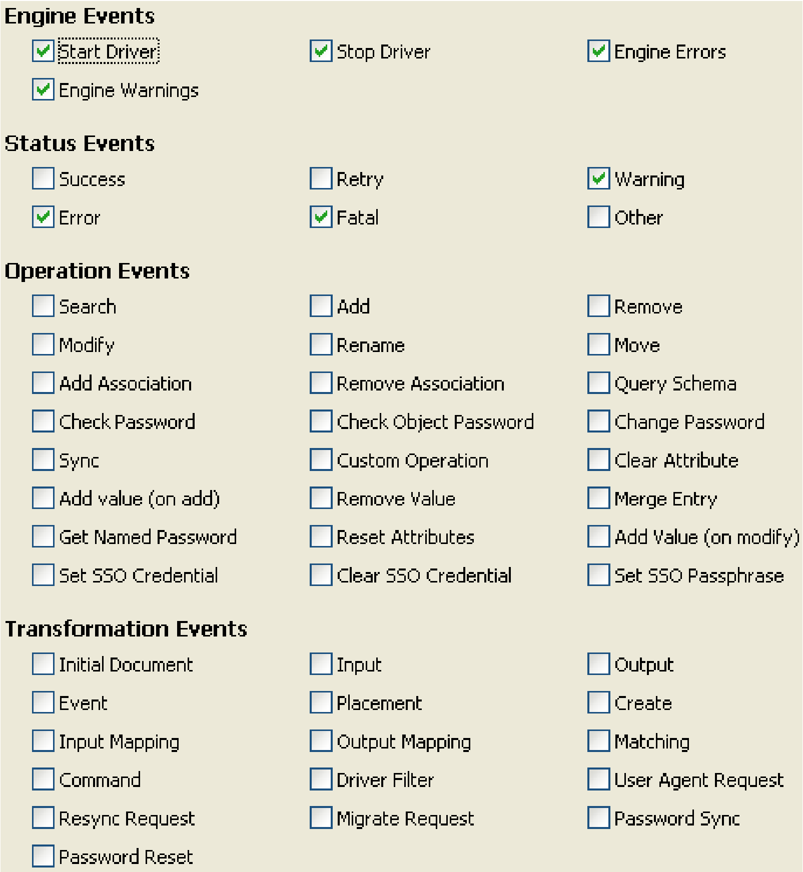
○ Update only the last log time

○ Turn logging off

□ Turn off logging to driver set, subscriber, and publisher logs

Maximum number of entries in the log (50-500): 50

4 Select events to log, then click *OK*.



Engine Events

☑ Start Driver ☑ Stop Driver ☑ Engine Errors

☑ Engine Warnings

Status Events

□ Success □ Retry ☑ Warning

☑ Error ☑ Fatal □ Other

Operation Events

□ Search □ Add □ Remove

□ Modify □ Rename □ Move

□ Add Association □ Remove Association □ Query Schema

□ Check Password □ Check Object Password □ Change Password

□ Sync □ Custom Operation □ Clear Attribute

□ Add value (on add) □ Remove Value □ Merge Entry

□ Get Named Password □ Reset Attributes □ Add Value (on modify)

□ Set SSO Credential □ Clear SSO Credential □ Set SSO Passphrase

Transformation Events

□ Initial Document □ Input □ Output

□ Event □ Placement □ Create

□ Input Mapping □ Output Mapping □ Matching

□ Command □ Driver Filter □ User Agent Request

□ Resync Request □ Migrate Request □ Password Sync

□ Password Reset

Managing Engine Services 10

The following drivers are used only for Metadirectory engine services, not for external connected systems. They are automatically installed when you install Identity Manager.

- Section 10.1, "Entitlements Service Driver," on page 271
- Section 10.2, "Manual Task Service Driver," on page 271
- Section 10.3, "GenericLoopBack Services Driver," on page 287
- Section 10.4, "GenericNull Services Driver," on page 288

You can also manage the engine services with the engine control values. For more information, see Section 10.5, "Engine Control Values," on page 289.

10.1 Entitlements Service Driver

Identity Manager allows you to synchronize data between connected systems. Entitlements allow you to set up criteria for a person or group that, once met, initiate an event to grant or revoke access to business resources within the connected system. This gives you one more level of control and automation for granting and revoking resources. For more information on entitlements, see Chapter 6, "Creating and Using Entitlements," on page 175.

10.2 Manual Task Service Driver

The Manual Task Service Driver is designed to notify one or more users that a data event has occurred and whether any action is required on the users' part. In an employee provisioning scenario, the data event might be the creation of a new User object and the user action might include assigning an office number by entering data into eDirectory™ or by entering data in an application. Other scenarios include notifying an administrator that a new user object has been created, notifying an administrator that a user has changed data on an object, etc.

Configuring the Manual Task Service Driver usually consists of configuring two separate but related subsystems: the Subscriber channel policies and e-mail templates, and the Publisher channel Web server templates and policies.

In addition, driver parameters such as SMTP server name and Web server port number must be configured.

In this section:

- Section 10.2.1, "Installing," on page 271
- Section 10.2.2, "Overview," on page 272
- Section 10.2.3, "Configuring Parameters and Templates," on page 279
- Section 10.2.4, "Additional Information," on page 287

10.2.1 Installing

- **Installation:** The Manual Task Service Driver is automatically installed when you install the *Metadirectory Server* option using the Identity Manager installation program.

- **Platforms:** The driver runs on the platforms supported by Identity Manager and the Remote Loader.
- **Activation:** The driver does not require separate activation. When you activate the Metadirectory engine, this driver is also activated.

10.2.2 Overview

This section contains information about how driver functionality works.

- "Modes of Operation" on page 272
- "How E-Mail Messages and Web Pages Are Created by the Manual Task Service Driver" on page 273
- "Templates" on page 274
- "Replacement Tokens" on page 276
- "Replacement Data" on page 276
- "Template Action Elements" on page 277
- "Subscriber Channel E-Mail" on page 277
- "Publisher Channel Web Server" on page 278

Modes of Operation

Two primary modes of operation are supported:

- **Direct Request for Data:** An e-mail message is sent requesting that a user enter data into eDirectory (possibly for consumption by another application). The e-mail recipient responds to the message by clicking a URL in the message. The URL points to the Web server running in the Publisher channel of the Manual Task Service Driver. The user then interacts with dynamic Web pages generated by the Web server to authenticate to eDirectory and to enter the requested data.
- **Event Notification:** An e-mail message is sent to a user without involving the Publisher channel. The e-mail message might simply be notification that something occurred in eDirectory, or it might be a request for data through a method other than the Publisher channel's Web server, such as Novell iManager, another application, or a custom interface.

Example: Subscriber Channel E-Mail, Publisher Channel Web Server Response

The following is an employee provisioning example scenario in which a new employee's manager assigns the employee a room number:

1. A new User object is created in eDirectory (for example, by the Identity Manager driver for the company's HR system).
2. The Manual Task Service Driver Subscriber channel sends an SMTP message to the user's manager and to the manager's assistant. The SMTP message contains a URL that refers to the Publisher channel Web server. The URL also contains data items identifying the user and identifying those authorized to submit the requested data.

3. The manager or the manager's assistant clicks the URL in the e-mail message to display an HTML form in a Web browser. The manager or assistant then does the following:
   - Selects the DN for his or her eDirectory User object as a means of identifying who is responding to the e-mail message.
   - Enters his or her eDirectory password.
   - Enters the room number for the new employee.
   - Clicks the Submit button.
4. The room number for the new employee is submitted to eDirectory via the Manual Task Service Driver Publisher channel.

Example: Subscriber Channel E-Mail, No Publisher Channel Response

The following is an example scenario in which a new employee's manager assigns the employee a computer in an asset management system:

1. A new User object is created in eDirectory by the Identity Manager driver for the company's HR system.
2. The Manual Task Service Driver Subscriber channel sends an SMTP message to the user's manager and to the manager's assistant. The SMTP message contains instructions for entering data into the asset management system.
3. The manager or assistant enters data into the asset management system.
4. (Optional) The computer identification data is brought into eDirectory via an Identity Manager driver for the asset management system.

How E-Mail Messages and Web Pages Are Created by the Manual Task Service Driver

E-mail messages, HTML Web pages, and XDS documents can all be considered documents. The Manual Task Service Driver creates documents dynamically, based on information supplied to the driver.

Templates are XML documents that contain the boilerplate or fixed portions of a document together with replacement tokens that indicate where the dynamic, or replacement, portions of the final, constructed document appear.

Both the Subscriber channel and the Publisher channel of the Manual Task Service Driver use templates to create documents. The Subscriber channel creates e-mail messages and the Publisher channel creates Web pages and XDS documents.

The dynamic portion of a document is supplied via replacement data. Replacement data on the Subscriber channel is supplied by the Subscriber channel policies, such as the Command Transformation policy. Replacement data on the Publisher channel is supplied by HTTP data to the Web server (both URL data and HTTP POST data). The Manual Task Service Driver can automatically supply certain data known to the Manual Task Service Driver, such as the Web server address.

The templates are processed by XSLT style sheets. These template-processing style sheets are separate from style sheets used as policies in the Subscriber or Publisher channels.

The replacement data is supplied as a parameter to the XSLT style sheet. The output of the style sheet processing is an XML, HTML, or text document that is used as the body of an e-mail message, as a Web page, or as a submission to Identity Manager on the Publisher channel.

Replacement data is passed from the Subscriber channel to the Publisher channel via a URL in the e-mail message. The URL contains a query portion that contains the replacement data items.

The Manual Task Service Driver ships with predefined style sheets sufficient to process templates in order to create e-mail documents, HTML documents, and XDS documents. Other custom style sheets can be written to provide additional processing options.

An advanced method of creating documents is also available, which uses only an XSLT style sheet and replacement data. No template is involved. However, this guide assumes the template method is used because the template method is easier to configure and maintain without XSLT programming knowledge.

Templates

Templates are XML documents that are processed by a style sheet in order to generate an output document. The output document can be XML, HTML, or plain text (or anything else that can be generated using XSLT).

Templates are used in the Manual Task Service Driver to generate e-mail message text on the Subscriber channel, and to generate dynamic Web pages and XDS documents on the Publisher channel.

Templates contain text, elements, and replacement tokens. Replacement tokens are replaced in the output document by data supplied to the style sheet processing the template.

Several examples of templates for various purposes follow. In the examples, the replacement tokens are the character strings that are between two $ characters.

Templates can also contain action elements. Action elements are control elements interpreted by the template-processing style sheet. Action elements are described in Appendix F, "Manual Task Service Driver: Template Action Elements Reference," on page 345.

The following example template is used to generate an HTML e-mail message body:

```
<html xmlns:form="http://www.novell.com/dirxml/manualtask/form">
<head></head>
<body>
Dear $manager$,<p/>
<p>
This message is to inform you that your new employee <b>$given-name$
$surname$</b> has been hired.
<p>
You need to assign a room number for this individual. Click <a
href="$url$">Here</a> to do this.
</p>
<p>
Thank you,<br/>
HR Department
</p>
</body>
</html>
```

The following example template is used to generate a plain text e-mail message body:

```
<form:text xmlns:form="http://www.novell.com/dirxml/manualtask/form">
Dear $manager$,

This message is to inform you that your new employee $given-name$ $surname$
has been hired.

You need to assign a room number for this individual. Use the following link to
do this:

$url$

Thank you,

The HR Department

</form:text>
```

The <form:text> element is required because templates must be XML documents. The <form:text> element is stripped as part of the template processing.

The following template is used to generate an HTML form used as a Web page for entering data:

```
<html  xmlns:form="http://www.novell.com/dirxml/manualtask/form">
<head>
<title>Enter room number for $subject-name$</title>
</head>
<body>
    <link href="novdocmain.css" rel="style sheet" type="text/css"/>
    <br/><br/><br/><br/>
    <form class="myform" METHOD="POST" ACTION="$url-base$/
process_template.xsl">
        <table cellpadding="5" cellspacing="10" border="1" align="center">
          <tr><td>
          <input TYPE="hidden" name="template" value="post_form.xml"/>
          <input TYPE="hidden" name="subject-name" value="$subject-name$"/>
          <input TYPE="hidden" name="association" value="$association$"/>
          <input TYPE="hidden" name="response-style sheet"
value="process_template.xsl"/>
          <input TYPE="hidden" name="response-template"
value="post_response.xml"/>
          <input TYPE="hidden" name="auth-style sheet"
value="process_template.xsl"/>
         <input TYPE="hidden" name="auth-template" value="auth_response.xml"/>
        <input TYPE="hidden" name="protected-data" value="$protected-data$"/>
          You are:<br/>
          <form:if-single-item name="responder-dn">
            <input TYPE="hidden" name="responder-dn" value="$responder-dn$"/>
            $responder-dn$
          </form:if-single-item>          <form:if-multiple-items
name="responder-dn">            <form:menu name="responder-dn"/>          </
form:if-multiple-items>
        </td></tr>
        <tr><td>
          Enter your password: <br/>
<input name="password" TYPE="password" SIZE="20" MAXLENGTH="40"/>
        </td></tr>
        <tr><td>
          Enter room number for $subject-name$:<br/>
          <input TYPE="text" NAME="room-number" SIZE="20" MAXLENGTH="20"
```

```
value="$query:roomNumber$"/>
        </td></tr>
        <tr><td>
          <input TYPE="submit" value="Submit"/> <input TYPE="reset"
value="Clear"/>
        </td></tr>
      </table>
    </form>
  </body>
</html>
```

The following template is used to generate an XDS document:

```
<nds>
  <input>
    <modify class-name="User" src-dn="not-applicable">
      <association>$association$</association>
      <modify-attr attr-name="roomNumber">
        <remove-all-values/>
        <add-value>
          <value>$room-number$</value>
        </add-value>
      </modify-attr>
    </modify>
  </input>
</nds>
```

Replacement Tokens

The items delimited by $ in the above example templates are replacement tokens. For example, $manager$ is replaced by the manager's actual name.

Replacement tokens can appear either in text or in XML attribute values (note the href value on the <a> element in the first example above).

Replacement Data

Replacement data consists of strings that take the place of replacement tokens in the output document generated from a template. Replacement data is either supplied by Subscriber channel data, Publisher channel HTTP data, or it is supplied automatically by the driver. An additional type of replacement data is data retrieved from eDirectory via Identity Manager (query data). Replacement data is more fully described in Appendix D, “Manual Task Service Driver: Replacement Data,” on page 337.

Subscriber channel data: Subscriber channel replacement data is of two types. The first type is used as replacement values for replacement tokens in templates for creating e-mail messages. The second type is placed in the query portion of a URL so that the data is available for use on the Publisher channel when the URL is submitted to the Publisher's Web server.

HTTP data: Replacement data is supplied to the Publisher channel Web server as URL query string data, HTTP POST data, or both.

Automatic data: The Manual Task Service Driver supplies automatic data. Automatic data items are described in Appendix E, “Manual Task Service Driver: Automatic Replacement Data Items,” on page 343.

Query data: Replacement tokens that start with query: are considered requests to obtain current data from eDirectory. The portion of the token that follows query: is the name of an eDirectory object attribute. The object to query is specified by one of the replacement data items `association`, `src-dn`, or `src-entry-id`. The items are considered in the order presented in the preceding sentence.

Template Action Elements

Action elements are namespace-qualified elements in the template that are used for simple logic control or that are used to create HTML elements for HTML forms. The namespace used to qualify the elements is http://www.novell.com/dirxml/manualtask/form. In this document and in the sample templates supplied with the Manual Task Service driver, the prefix used is `form`.

Action elements are described in detail in Appendix F, "Manual Task Service Driver: Template Action Elements Reference," on page 345.

Subscriber Channel E-Mail

The Subscriber channel of the Manual Task Service Driver is designed to send e-mail messages. To accomplish this, the driver supports a custom XML element named `<mail>`. Policies on the Subscriber channel construct a `<mail>` element in response to some eDirectory event, such as the creation of a user. An example `<mail>` element appears below:

```
<mail src-dn="\PERIN-TAO\novell\Provo\Joe">
  <to>JStanley@novell.com</to>
  <cc>carol@novell.com</cc>
  <reply-to>HR@novell.com</reply-to>
  <subject>Room Assignment Needed for: Joe the Intern</subject>
  <message mime-type="text/html">
    <stylesheet>process_template.xsl</stylesheet>
    <template>html_msg_template.xml</template>
    <replacement-data>
      <item name="manager">JStanley</item>
      <item name="given-name">Joe</item>
      <item name="surname">The Intern</item>
      <url-data>
        <item name="file">process_template.xsl</item>
        <url-query>
          <item name="template">form_template.xml</item>
          <item name="responder-dn" protect="yes">\PERIN-TAO\big-org\phb</
item>
          <item name="responder-dn" protect="yes">\PERIN-TAO\big-org\carol</
item>
          <item name="subject-name">Joe The Intern</item>
        </url-query>
      </url-data>
    </replacement-data>
    <resource cid="css-1">novdocmain.css</resource>
  </message>
  <message mime-type="text/plain">
    <stylesheet>process_text_template.xsl</stylesheet>
    <template>txt_msg_template.xml</template>
    <replacement-data>
      <item name="manager">JStanley</item>
      <item name="given-name">Joe</item>
      <item name="surname">The Intern</item>
      <url-data>
```

```
          <item name="file">process_template.xsl</item>
          <url-query>
            <item name="template">form_template.xml</item>
            <item name="responder-dn" protect="yes">\PERIN-TAO\big-org\phb</
item>
            <item name="responder-dn" protect="yes">\PERIN-TAO\big-org\carol</
item>
            <item name="subject-name">Joe The Intern</item>
          </url-query>
        </url-data>
      </replacement-data>
    </message>
  <attachment>HR.gif</attachment>
</mail>
```

The Subscriber channel of the Manual Task Service Driver uses the information contained in the <mail> element to construct an SMTP e-mail message. A URL can be constructed and inserted into the e-mail message through which the e-mail recipient can respond to the e-mail message. The URL can point to the Publisher channel Web server or it can point to some other Web server.

The <mail> element and its content are described in detail in Appendix G, "Manual Task Service Driver: <mail> Element Reference," on page 349.

Publisher Channel Web Server

The Publisher channel of the Manual Task Service Driver runs a Web server configured so that users can enter data into eDirectory through a Web browser. The Web server is designed to work in conjunction with e-mail messages sent from the Subscriber channel of the Manual Task Service Driver.

The Publisher channel Web server can serve static files and dynamic content. Examples of static files are .css style sheets, images, etc. Examples of dynamic content are Web pages that change based on the replacement data contained in the URL or HTTP POST data.

The Publisher channel Web server is normally configured to allow a user to enter data into eDirectory in response to an e-mail that was sent by the Subscriber channel. A typical user interaction with the Web server is as follows:

1. The user uses a Web browser to submit the URL from the e-mail message to the Web server. The URL specifies the style sheet, template, and replacement data used to create a dynamic Web page (typically containing an HTML form).
2. The Web server creates an HTML page by processing the template with the style sheet and replacement data. The HTML page is returned to the user's Web browser as the resource referred to by the URL.
3. The browser displays the HTML page and the user enters the requested information.
4. The browser sends an HTTP POST request containing the entered information as well as other information that originated from the e-mail URL. The DN of the user responding to the e-mail and the user's password must be in the POST data.
5. The Web server uses the user's DN and password to authenticate. If the authentication fails, then a Web page containing a failure message is returned as the result of the POST request. The failure message can be constructed using a style sheet and template specified in the POST data. If authentication succeeds, processing continues.

6. The Web server constructs an XDS document using a style sheet and template specified in the POST data. The XDS document is submitted to Identity Manager on the Publisher channel.
7. The result of the XDS document submission, together with a style sheet and template specified in the POST data, is used to construct a Web page indicating to the user the result of the data submission. This Web page is sent to the browser as the result of the POST request.

10.2.3 Configuring Parameters and Templates

- "Driver Settings" on page 279
- "Subscriber Settings" on page 281
- "Publisher Settings" on page 282
- "Subscriber Channel Policies" on page 282
- "Subscriber Channel E-Mail Templates" on page 284
- "Publisher Channel Policies" on page 284
- "Publisher Channel Web Page Templates" on page 285
- "Publisher Channel XDS Templates" on page 285
- "Trace Settings" on page 286

Driver Settings

This section describes parameters that appear in the Driver Settings section in the driver object user interface.

Many of these parameters are actually for the Publisher channel Web server. They appear under the Driver Settings area because the Manual Task Service Driver Subscriber also needs access to them.

- "DN of the Document Base" on page 279
- "Document Directory" on page 280
- "Use HTTP Server (true|false)" on page 280
- "HTTP IP Address or Host Name" on page 280
- "HTTP Port" on page 281
- "Name of KMO" on page 281
- "Name of Keystore File" on page 281
- "Keystore Password" on page 281
- "Name of Certificate (key alias)" on page 281
- "Certificate Password (key password)" on page 281

DN of the Document Base

This parameter is an eDirectory DN of a container object. The Manual Task Service Driver can load XML documents (including XSLT style sheets) from eDirectory as well as from disk. If XML documents should be loaded from eDirectory, this parameter identifies the root container from which documents are loaded.

Documents loaded from eDirectory reside in the attribute value of an eDirectory object. If unspecified, the attribute is XmlData. The attribute can be specified by appending a # character followed by the attribute name to the name of the object containing the document.

For example, suppose that the document base DN is specified to be *novell\Manual Task Document*s and that there is a container under *Manual Task Documents* named *templates*.

If a DirXML-Style Sheet object named *e-mail _template* resides under the *templates* directory, then the following resource identifiers can be used to refer to the XML document: *templates/e-mail _template* or *templates/e-mail _template#XmlData*.

The resource identifiers can be supplied as replacement data, URL data, or HTTP POST data. For example, the following element might appear under a `<message>` element on the Subscriber channel:

```
<template>templates/e-mail _template#XmlData</template>
```

Document Directory

This parameter identifies a file system directory that is used as the base directory for locating resources such as templates, XSLT style sheets, and other file resources served by the Publisher channel Web server. Example values are:

Windows	`c:\Novell\Nds\mt_files`
NetWare®	`SYS:\SYSTEM\mt_files`
UNIX	`/usr/lib/dirxml/rules/manualtask/mt_files`

Use HTTP Server (true|false)

This parameter indicates whether the Publisher channel should run a Web server or not. Set the parameter to True if the Web server should be run, or False if the Web server should not be run.

If the Manual Task Service Driver is only to be used for sending e-mail with no response URL, or with a URL that points to another application, then the HTTP server should not be run, to save system resources.

HTTP IP Address or Host Name

This parameter allows you to specify which multiple, local IP addresses the Publisher channel Web server should us to listen for HTTP requests.

Leaving the HTTP IP address or host name parameter value blank causes the Publisher channel Web server to listen on the default IP address. For servers with a single IP address, this is sufficient. Placing a dot-notation IP address as the parameter value causes the Publisher channel Web server to listen for HTTP requests on the address specified.

The value specified for HTTP IP address or host name is used by the Subscriber channel mail handler to construct URLs if the host name or address is not specified in the mail command element. If the parameter Use HTTP server (true|false) is set to False, the HTTP IP address or host name can be used to specify the address or name of a Web server to use in constructing URLs for mail messages.

HTTP Port

This parameter is an integer value indicating which TCP port the Publisher channel Web server should listen on for incoming requests. If this value is not specified, the port number defaults to 80 or 443, depending on whether or not SSL is being used for the Web server connections.

If the Manual Task Service Driver is running on the Identity Manager server (that is, it is not being run under the Remote Loader on a remote machine) then the HTTP port should be set to something other than 80 or 443. This is because iMonitor or another process is typically using ports 80 and 443.

Name of KMO

If it is not blank, this parameter is the name of an eDirectory Key Material Object that contains the server certificate and key used for SSL by the Publisher channel Web server.

Setting this parameter causes the Publisher channel Web server to use SSL for servicing HTTP requests.

This parameter takes precedence over any Java keystore parameters (see "Name of Keystore File" on page 281.)

Using SSL is recommended for security reasons because eDirectory passwords are passed in HTTP POST data when using the Publisher channel Web server.

Name of Keystore File

This parameter, together with the Keystore password, Name of certificate(key alias), and Certificate password (key password), is used to specify a Java keystore file that contains a certificate and key used for SSL by the Publisher channel Web server.

Setting this parameter causes the Publisher channel Web server to use SSL for servicing HTTP requests.

If the Name of KMO parameter is set, this parameter and its associated parameters are ignored.

Using SSL is recommended for security reasons because eDirectory passwords are passed in HTTP POST data when using the Publisher channel Web server.

Keystore Password

This parameter specifies the password for the Java keystore file specified with the Name of keystore file parameter.

Name of Certificate (key alias)

This parameter specifies the name of the certificate to use in the Java keystore file specified with the Name of keystore file parameter.

Certificate Password (key password)

This parameter specifies the password for the certificate specified using the Name of certificate (key alias) parameter.

Subscriber Settings

- "SMTP Server" on page 282

- "SMTP Account Name" on page 282
- "Default "From" Address" on page 282
- "Additional Handlers" on page 282

SMTP Server

This parameter specifies the name of the SMTP server that the Subscriber channel uses to send e-mail messages.

SMTP Account Name

If the SMTP server specified by using the SMTP server parameter requires authentication, this parameter specifies the account name to use for authentication. The password used is the Application password associated with the driver Authentication parameters.

Default "From" Address

If specified, this is an e-mail address used in the SMTP from field for e-mail messages sent by the Subscriber channel. If this is not specified, then the <mail> elements sent to the Subscriber must contain a <from> element.

A <from> element under <mail> elements sent to the Subscriber overrides this parameter.

Additional Handlers

If specified, this is a whitespace-separated list of Java class names. Each class name is a custom class that implements the com.novell.nds.dirxml.driver.manualtask.CommandHandler interface and handles a custom XDS element. The handler for <mail> is a built-in handler.

Additional information about custom handlers is available in Appendix I, "Manual Task Service Driver: Custom Element Handlers for the Subscriber Channel," on page 365.

Publisher Settings

This section describes settings for the Publisher channel.

Additional Servlets

If non-blank, this is a whitespace-separated list of Java class names. Each class name is a custom class that extends javax.servlet.http.HttpServer. Custom servlets can be used to extend the functionality of the Publisher channel Web server.

Additional information about custom servlets is available in Appendix J, "Manual Task Service Driver: Custom Servlets for the Publisher Channel," on page 367.

Subscriber Channel Policies

The configuration of the Subscriber channel policies depends on what a particular installation wants to accomplish with the Manual Task Service Driver. However, there are certain guidelines that might be helpful.

In general, the best place to construct a <mail> element to send to the Subscriber is in the Command Transformation policy. The reason for this is that most Metadirectory engine processing has been completed by the time commands reach the Command Transformation policy. This means that

Create policies have been processed for Add events (allowing vetoing of Add events for objects that don't have all the attributes necessary for constructing the e-mail, for example). This also means that Modify events for objects without associations have already been converted to Add events.

The XSLT style sheet that constructs the e-mail message might or might not need to query eDirectory for additional information.

For example, if the e-mail message is simply a welcome message to a new employee, the Add command can contain all the information necessary: Given Name, Surname, and Internet E-mail Address. This is accomplished by specifying in the Create policy that Given Name, Surname, and Internet E-mail Address are required attributes. This ensures that only add commands that contain the necessary information can reach the Command Transformation.

However, if the e-mail message is a message to the manager of an employee, the style sheet needs to query eDirectory. The manager DN can be obtained from the Add event for the employee's User object, but a query must be made to obtain the manager's e-mail address because that information is an attribute of the manager's User object.

In addition, if e-mail notifications are being generated as the result of Modify commands for objects that are associated with the driver, queries must be made to obtain information not contained in the modify command.

Blocking Commands from Reaching the Subscriber Channel

If e-mail messages are to be generated from events other than Add events, the Add events must be allowed to reach the Subscriber channel for those objects that are to be monitored. Allowing Add events to reach the Subscriber channel results in a generated association value being returned to Identity Manager from the Subscriber channel.

It is important that eDirectory objects to be monitored by the Manual Task Service Driver policies have an association for the Manual Task Service Driver. Only objects that have an association have Delete, Rename, and Move events reported to the driver. In addition, Modify events on objects that do not have an association are converted to Add events after the Subscriber channel event transformation.

All other commands (Modify, Move, Rename, and Delete) should be blocked by the Command Transformation policy and prevented from reaching the Subscriber channel. The Subscriber channel handles only `Add` commands and `Mail` commands. Other commands result in the Subscriber channel returning an error.

Generating E-Mail Messages

E-mail messages are sent by the Subscriber in response to receiving a <`mail`> element that describes the e-mail message to be sent. See Appendix G, "Manual Task Service Driver: <mail> Element Reference," on page 349 for a description of the <`mail`> element and its content.

E-mail messages can be generated in response to any Identity Manager event (Add, Modify, Rename, Move, Delete).

The replacement data that is supplied with the <message> element children of a <mail> element depends on two primary factors:

- The template used to generate the message body. Replacement items to be used by the e-mail template appear as children of the <replacement-data> element.
- The information needed by the Web page templates on the Publisher channel if the e-mail is to result in a response on the Publisher channel. Replacement items to be used by the Web page templates appear as children of the <url-query> element, which is a child of <url-data>, which in turn is a child of <replacement-data>.

If the e-mail message contains a URL that points to the Publisher channel Web server and is used to solicit information from a user, the replacement data must contain at least one responder-dn item. The values of the responder-dn items must be the DNs of the User objects of the users to which the message is being sent.

If a query replacement token (see "Replacement Data" on page 276) is used in the template, then the replacement data for the <message> element must contain an item named src-dn, src-entry-id, or association with the appropriate value. An association item can only be used if the eDirectory object to be queried already has an association for the Manual Task Service Driver. The association generated by the Subscriber for unassociated objects cannot be used because it hasn't been written to the eDirectory object when the query takes place.

The <message> element can specify the MIME type of the message body. If the MIME type is specified but a style sheet is not specified (that is, there is no <stylesheet> element child of <message>), one of two default style sheet names is used. If the MIME type is text/plain, the default style sheet name is process_text_template.xsl. If the MIME type is anything other than text/plain, the default style sheet name is process_template.xsl.

Subscriber Channel E-Mail Templates

E-mail templates are XML documents containing boilerplate and replacement tokens. E-mail templates are used to generate e-mail message body text. See "Templates" on page 274 for general information about templates.

The replacement tokens used in an e-mail template dictate the <item> elements that must be supplied as children of the <replacement-data> element that is constructed by the Subscriber channel policy that constructs the <mail> element. For example, if the e-mail template has the replacement token $employee-name$, there must be an <item name="employee-name"> element in the replacement data for the <message> element. If the employee name item is not present, the resulting e-mail message body has no text in the location occupied by the replacement token in the template.

E-mail templates can be used to generate message bodies that are plain text, HTML, or XML.

If an e-mail template generates a plain text message, it must be processed by a style sheet that specifies plain text as its output type. If the style sheet does not specify plain text as its output type, undesirable XML escaping occurs. The default Manual Task Service Driver style sheet, process_text_template.xsl, is normally used for processing templates that result in plain text.

Publisher Channel Policies

In most implementations of the Manual Task Service Driver, no Publisher channel policies are needed. This is because is it possible to construct the Web page and XDS templates so they result in exactly the XDS required and the XDS doesn't need additional processing by policies.

If policies are required, they are very specific to an installation.

Publisher Channel Web Page Templates

Web page templates are XML documents containing boilerplate and replacement tokens. Web page templates are used to generate Web page documents (typically HTML documents). See “Templates” on page 274 for general information about templates.

Replacement tokens in Web page templates dictate what replacement data is supplied as URL query data on the Subscriber channel. Replacement data on the Publisher channel is obtained from the URL query string for HTTP GET requests and from the URL query string and the POST data for HTTP POST requests.

As an example of the flow of replacement data from the Subscriber channel to the e-mail message and then to the Publisher channel Web server, consider the following scenario:

The Manual Task Service Driver is configured so that a new employee's manager is asked to assign a room number to the new employee. The trigger for the e-mail to the manager is the <add> command for a new User object that is processed by the Subscriber channel Command Transformation policy.

When the manager clicks a URL in the e-mail message, a Web page is displayed in the manager's Web browser. The Web page must indicate for whom the manager is entering a room number.

To accomplish this, the <url-query> element on the Subscriber channel contains a replacement data item that identifies the new user by name:

```
<item name=”subject-name”>Joe the Intern</item>
```

This causes the URL query string to contain (among other things) “subject-name=Joe%20the%20Intern”. The “%20” is a URL-encoded space.

The manager's Web browser submits the URL to the Publisher channel Web server when the manager clicks the URL in the e-mail message. The Web server constructs a replacement data item named subject-name with the value Joe the Intern.

The Web page template also specified by the URL contains a replacement token $subject-name$. When the Web page template is processed by the style sheet to construct the Web page, the replacement token is replaced by Joe the Intern, which customizes the Web page for the employee whose User object creation caused the e-mail to be sent.

For additional information on a complete Subscriber-channel-to-Publisher-channel transaction, see Appendix H, “Manual Task Service Driver: Data Flow Scenario for New Employee,” on page 353.

Publisher Channel XDS Templates

XDS templates are XML documents containing boilerplate and replacement tokens. XDS templates are used to generate XDS documents that are submitted to Identity Manager on the Manual Task Service Driver's Publisher channel. See “Templates” on page 274 for general information about templates.

Replacement tokens in XDS templates dictate some of the replacement data that is supplied to the Web server as data in an HTTP POST request.

For example, consider the following XDS template:

```
<nds>
  <input>
    <modify class-name="User" src-dn="not-applicable">
      <association>$association$</association>
      <modify-attr attr-name="roomNumber">
        <remove-all-values/>
        <add-value>
          <value>$room-number$</value>
        </add-value>
      </modify-attr>
    </modify>
  </input>
</nds>
```

The replacement tokens in the template dictate that the HTTP POST data must supply an association value and a room-number value.

Normally, the association value would originate in the Subscriber channel. The Subscriber channel e-mail would place association=value in the query string of the URL that is placed in the e-mail message. The Web page template used to generate the Web page when the URL is submitted to the Web server would typically place the association value in a hidden INPUT element:

```
<INPUT TYPE="hidden" NAME="association" VALUE="$association$"/>
```

Placing the association value as a hidden INPUT element causes the "association=value" pair to be submitted as part of the HTTP POST data.

The room-number value is entered in the Web page using an INPUT element similar to the following:

```
<input TYPE="text" NAME="room-number" SIZE="20" MAXLENGTH="20"/>
```

If the manager enters 1234 and clicks Submit, the Web browser sends "room-number=1234" as part of the HTTP POST data.

The Web server then generates an `<item name="association">` replacement data item and an `<item name="room-number">` replacement data item that are used when processing the XDS template.

The XDS document is generated by processing the XDS template with the style sheet specified in the POST data, then the XDS document is submitted to Identity Manager on the Manual Task Service Driver's Publisher channel.

Trace Settings

The Manual Task Service Driver outputs messages with various trace levels:

Level	Trace Message Description
0	No trace messages
1	Single-line messages tracing basic operation
2	No additional messages (Metadirectory Engine traces XML documents at this level and above)
3	No additional messages

Level	Trace Message Description
4	Messages relating to document construction from templates and style sheets
5	Replacement data documents traced

10.2.4 Additional Information

For additional information on Manual Task Service driver settings, refer to the following sections:

- Appendix D, “Manual Task Service Driver: Replacement Data,” on page 337
- Appendix E, “Manual Task Service Driver: Automatic Replacement Data Items,” on page 343
- Appendix F, “Manual Task Service Driver: Template Action Elements Reference,” on page 345
- Appendix G, “Manual Task Service Driver: <mail> Element Reference,” on page 349
- Appendix H, “Manual Task Service Driver: Data Flow Scenario for New Employee,” on page 353
- Appendix I, “Manual Task Service Driver: Custom Element Handlers for the Subscriber Channel,” on page 365
- Appendix J, “Manual Task Service Driver: Custom Servlets for the Publisher Channel,” on page 367

10.3 GenericLoopBack Services Driver

The GenericLoopBack Services driver is a utility driver that forwards all operations submitted to the Subscriber channel to the Publisher channel of the same driver. It can be used to implement a wide variety of custom behavior that can be implemented in a policy.

The driver does not have any communication with any external application. The Subscriber channel forwards operations to the Publisher channel of the same driver.

The GenericLoopBack Services driver is automatically installed when you install the Metadirectory Server option in the Identity Manager installation program. The driver runs on the platforms supported by Identity Manager and the Remote Loader service. The driver does not require a separate activation. When you activate the Metadirectory engine, the driver is also activated.

A base configuration of the driver does not contain much information:

- No policies.
- An empty filter.
- Publisher heartbeat configuration option is available, but it is set to Disabled.
- No prompts for information when importing the driver.

To configure the GenericLoopBack driver:

1. In iManager, select *Identity Manager Utilities > Import Configuration.*
2. Select a driver set, then click *Next.*

   If you place this driver in a new driver set, you must specify a driver set name, context, and associated server.

3. Select how you want the driver configurations sorted:
   - All configurations
   - Identity Manager 3.5 configurations
   - Identity Manager 3.0 configurations
   - Configurations not associated with an IDM version
4. Select *Loopback Service Driver*, then click *Next*.
5. Specify a name for the driver, then click *Next*.
6. Define security equivalences for the driver and exclude administrative roles from replication, then click *Next*.
7. Read the summary, then click *Finish*.

10.4 GenericNull Services Driver

The GenericNull Services driver provides a minimal driver that allows a driver shim to be set up to perform tasks that are completely implemented in policy. It is similar to the GenericLoopBack driver in the function it serves. It is different in one important way: it does not connect the Subscriber channel to the Publisher channel, but rather acts as a sink for most operations, simulates doing something with operations, but returning success. The only information sent on the Publisher channel to the Identity Manager engine is the driver heartbeat.

The GenericNull Services driver is automatically installed when you install the Metadirectory Server option using the Identity Manager installation program. The driver runs on the platforms supported by Identity Manager and the Remote Loader service. The driver does not require a separate activation. When you activate the Metadirectory engine, the driver is also activated.

A base configuration of the driver does not contain much information:

- No policies.
- An empty filter.
- Publisher heartbeat configuration option is available, but it is set to Disabled.
- No prompts for information when importing the driver.

Typical usage of the GenericNull Services Driver includes the following:

- Adding the classes and attributes that you want to monitor for change in the Subscriber Filter as *Synchronize* for the class and *Notify* for the attribute.
- Add Subscriber Event Transformation policies that react to specific object or attribute changes and perform actions such as:
  - Modifications back into the Identity Vault (using actions that manipulate source attributes and objects).
  - Send e-mail.
  - Generate custom Audit Events.
  - Call extension functions to communicate the change outside of Identity Manager.
- Add a final Subscriber Event Transformation policy that vetoes all events.

To configure the GenericNull Services driver :

1. In iManager, select *Identity Manager Utilities > Import Configuration*.
2. Select a driver set, then click *Next*.

   If you place this driver in a new driver set, you must specify a driver set name, context, and associated server.
3. Select how you want the driver configurations sorted:
   - All configurations
   - Identity Manager 3.5 configurations
   - Identity Manager 3.0 configurations
   - Configurations not associated with an IDM version
4. Select *GenericNull-IDM_3_5_1 Driver*, then click *Next*.
5. Specify a name for the driver, then click *Next*.
6. Define security equivalences for the driver and exclude administrative roles from replication, then click *Next*.
7. Read the summary, then click *Finish*.

10.5 Engine Control Values

The engine control values are a means through which certain default behaviors of the Metadirectory engine can be changed. The values can only be accessed if a server is associated with the Driver Set object.

In iManager:

1. Click *Identity Manager > Identity Manager Overview*, then click *Search* to search for the driver set that is associated with the driver.
2. Browse to the driver, then click the upper right corner of the driver icon.
3. Click *Edit Properties > Engine Control Values*.

   See Table 10-1 for a list of the engine control values.

In Designer:

1. In the Modeler, right-click the driver line.
2. Select *Properties > Engine Control Values*.
3. Click the tooltip icon to the right of the *Engine Controls For Server* field. If a server is associated with the Identity Vault, and if you are authenticated, the engine control values display in the large pane.

   See Table 10-1 for a list of the engine control values.

Table 10-1 *Engine Control Values*

Option	Description
Subscriber channel retry interval in seconds	The Subscriber channel retry interval controls how frequently the Metadirectory engine retries the processing of a cached transaction after the application shim's Subscriber object returns a retry status.
Qualified form for DN-syntax attribute values	The qualified specification for DN-syntax attribute values controls whether values for DN-syntax attribute values are presented in unqualified slash form or qualified slash form. A True setting means the values are presented in qualified form.
Qualified form from rename events	The qualified form for rename events controls whether the new-name portion of rename events coming from the Identity Vault is presented to the Subscriber channel with type qualifiers. For example, CN=. A True setting means the names are presented in qualified form.
Maximum eDirectory replication wait time in seconds	The maximum eDirectory™ replication wait time controls the maximum time that the Metadirectory engine waits for a particular change to replicate between the local replica and a remote replica. This only affects operations where the Metadirectory engine is required to contact a remote eDirectory server in the same tree to perform an operation and might need to wait until some change has replicated to or from the remote server before the operation can be completed (for example, object moves when the Identity Manager server does not hold the master replica of the moved object; file system rights operations for Users created from a template.)
Use non-compliant backwards-compatible mode for XSLT	This control sets the XSLT processor used by the Metadirectory engine to a backward-compatible mode. The backward-compatible mode causes the XSLT processor to use one or more behaviors that are not XPath 1.0 and XSLT 1.0 standards-compliant. This is done in the interest of backward-compatibility with existing DirXML® style sheets that depend on the non-standard behaviors. For example, the behavior of the XPath “!=” operator when one operand is a node set and the other operand is other than a node set is incorrect in DirXML releases up to and including Identity Manager 2.0. This behavior has been corrected; however, the corrected behavior is disabled by default through this control in favor of backward-compatibility with existing DirXML style sheets.
Maximum application objects to migrate at once	This control is used to limit the number of application objects that the Metadirectory engine requests from an application during a single query that is performed as part of a Migrate Objects from Application operation. If java.lang.OutOfMemoryError errors are encountered during a Migrate from Application operation, this number should be set lower than the default. The default is 50. **NOTE:** This control does not limit the number of application objects that can be migrated; it merely limits the batch size.

Option	Description
Set creatorsName on objects created in Identity Vault	This control is used by the Identity Manager engine to determine if the creatorsName attribute should be set to the DN of this driver on all objects created in the Identity Vault by this driver. Setting the creatorsName attribute allows for easily identifying objects created by this driver, but also carries a performance penalty. If not set, the creatorsName attribute defaults to the DN of the NCP™ Server object that is hosting the driver.
Write pending associations	This control determines whether the Identity Manager engine writes a pending association on an object during Subscriber channel processing. Writing a pending association confers little or no benefit but does incur a performance penalty. Nevertheless, the option exists to turn it on for backward compatibility.
Use password event values	This control determines the source of the value reported for the nspmDistributionPassword attribute for Subscriber channel Add and Modify events. Setting the control to False means that the current value of the nspmDistributionPassword is obtained and reported as the value of the attribute event. This means that only the current password value is available. This is the default behavior. Setting the control to True means that the value recorded with the eDirectory event is decrypted and is reported as the value of the attribute event. This means that both the old password value (if it exists) and the replacement password value at the time of the event are available. This is useful for synchronizing passwords to certain applications that require the old password to enable setting a new password.
Enable password synchronization status reporting	This control determines whether the Identity Manager engine reports the status of Subscriber channel password change events. Reporting the status of Subscriber channel password change events allows applications such as the Identity Manager User Application to monitor the synchronization progress of a password change that should be synchronized to the connected application.

High Availability

11

You can use Identity Manager with shared storage to provide high availability. Some steps are required to use Novell® eDirectory™ and Identity Manager in a clustering environment.

In this section:

- Section 11.1, "Configuring eDirectory and Identity Manager for Use with Shared Storage on Linux and UNIX," on page 293
- Section 11.2, "Case Study for SuSE Linux," on page 297

11.1 Configuring eDirectory and Identity Manager for Use with Shared Storage on Linux and UNIX

This section provides steps for configuring eDirectory and Identity Manager for failover in a high availability cluster using shared storage. The information in this section is generalized for shared storage high availability clusters on any Linux or UNIX platform; the information is not specific to a particular cluster manager.

The basic concept to understand is that state data for eDirectory and Identity Manager must be located on the shared storage so that it is available to the cluster node that is currently running the services. In practice, this means the eDirectory datastore, typically located in /var/nds/dib, must be relocated to the cluster shared storage. The Identity Manager state data is also located in /var/nds/dib. Each eDirectory instance on the cluster nodes must be configured to use the datastore on the shared storage. Other eDirectory configuration data must also reside on shared storage.

In addition to the eDirectory datastore, it is also necessary to share NICI (Novell International Cryptographic Infrastructure) data so that server-specific keys are replicated among the cluster nodes. Rather than move the NICI data to shared storage, it is generally better to copy the NICI data to local storage on each cluster node. This is preferable so that client NICI functionality is available on a cluster node even when the cluster node is in a secondary state and is not hosting the shared storage.

Sharing eDirectory and NICI data is discussed in the following sections, and is based on these assumptions:

- You are using the default install locations for NICI, eDirectory, and Identity Manager data and configuration.

  Identity Manager data is not discussed separately from eDirectory data because the Identity Manager data of interest is located with the eDirectory data
- You are familiar with eDirectory and Identity Manager installation procedures.
- You are using a two-node cluster.

  A two-node cluster is by far the most common configuration used for high-availability. However, the concepts in this section can easily be extended to an *n*-node cluster.

In this section:

- Section 11.1.1, “Installing eDirectory,” on page 294
- Section 11.1.2, “Installing Identity Manager,” on page 294
- Section 11.1.3, “Sharing NICI Data,” on page 294
- Section 11.1.4, “Sharing eDirectory and Identity Manager Data,” on page 295
- Section 11.1.5, “Identity Manager Driver Considerations,” on page 296

11.1.1 Installing eDirectory

NOTE: NICI is installed as part of the eDirectory install process.

1 Install eDirectory on the primary cluster node.

2 Configure eDirectory on the primary cluster node. Either create a new tree on the primary cluster node or install the server into an existing tree. For the eDirectory server name, use something other than the name of the UNIX server. Use a name that is common to the cluster, rather than specific to one of the cluster nodes.

3 Install the same version of eDirectory on the secondary cluster node. Do not configure eDirectory on the secondary cluster node.

   The secondary node does not have a separate tree.

11.1.2 Installing Identity Manager

1 Install Identity Manager on the primary cluster node using the *Metadirectory Server* option.

   The installation process installs the Identity Manager files and configures the eDirectory tree for use with Identity Manager.

2 Install the same version of Identity Manager on the secondary cluster node using the secondary cluster switch, by entering

```
dirxml_platform.bin -DCLUSTER_INSTALL="true"
```

   During the install, choose the *Metadirectory Serve*r option.

   Using the secondary cluster switch installs the Identity Manager files but does not attempt to perform any additional eDirectory configuration. No configuration is necessary, because the secondary node does not have a separate tree.

11.1.3 Sharing NICI Data

NICI provides cryptographic services used by eDirectory, Identity Manager, and Novell client applications. When used with eDirectory, NICI provides server-specific keys. These server-specific keys must be the same on all cluster nodes where eDirectory runs as a cluster service.

There are two possible ways of sharing the NICI data:

- Placing the NICI data on the cluster shared storage.

  The disadvantage of this method is that applications that depend on NICI will fail on a cluster node when the cluster node is not hosting the shared storage.
- Copying the NICI data from the primary server to the secondary server's local storage.

To copy the NICI data:

1. Rename /var/novell/nici on the secondary cluster node to something else (such as /var/novell/nici.sav).
2. Copy the /var/novell/nici directory from the primary cluster node to the secondary cluster node.

   This can be done using scp or by creating a tar file of the /var/novell/nici directory on the primary node, transferring it to the secondary node, and untarring the directory on the secondary node.

11.1.4 Sharing eDirectory and Identity Manager Data

By default, eDirectory stores its datastore in /var/nds/dib. Other items of configuration and state are also stored in /var/nds and its subdirectories. The default configuration directory for eDirectory is /etc. The following steps are necessary to configure eDirectory and Identity Manager for use with the shared storage in a high availability cluster. These steps assume that the shared storage is mounted at /shared.

- "On the Primary Node" on page 295
- "On the Secondary Node" on page 296

On the Primary Node

1. Copy the /var/nds directory subtree to /shared/var/nds.
2. Rename the /var/nds directory (for example, to /var/nds.sav).

   This is not required, but creating a backup at this stage gives you the ability to start over, if necessary, without reinstalling eDirectory.
3. Create a symbolic link from /var/nds to /shared/var/nds (for example, `ln -s /shared/var/nds /var/nds`).
4. Create the following symbolic links:

Link from	Link to
/shared/var/nds/class16.conf	/etc/class16.conf
/shared/var/nds/class32.conf	/etc/class32.conf
/shared/var/nds/help.conf	/etc/help.conf
/shared/var/nds/ndsimonhealth.conf	/etc/ndsimonhealth.conf
/shared/var/nds/miscicon.conf	/etc/miscicon.conf
/shared/var/nds/ndsimon.conf	/etc/ndsimon.conf
/shared/var/nds/macaddr	/etc/macaddr

5. Make a backup copy of /etc/nds.conf.
6. Move /etc/nds.conf to /shared/var/nds.

7. Edit /shared/var/nds/nds.conf and place the following entries into the file (overwriting any current entries with the same names):
   - n4u.nds.dibdir=/shared/var/nds/dib
   - n4u.server.configdir=/shared/var/nds
   - n4u.server.vardir=/shared/var/nds
   - n4u.nds.preferred-server=localhost

   For the following entries, replace eth0:0 with the interface name of the cluster-shared ethernet interface. Also replace lo with the interface name of the localhost ethernet interface.
   - n4u.nds.server.interfaces=eth0:0@524,lo@524
   - http.server.interfaces=eth0:0@8008,lo@8008
   - https.server.interfaces=eth0:0@8009,lo@8009
8. Create a symbolic link from /etc/nds.conf to /shared/var/nds/nds.conf.
9. Start ndsd and verify that ndsd runs with the shared storage.
10. Stop ndsd.
11. Place ndsd in the cluster manager's list of resources to be hosted.
12. Remove ndsd from the list of daemons to be started by the init process at boot time.

On the Secondary Node

1. Rename the /var/nds directory (e.g., to /var/nds.sav). This is not strictly necessary, but backups provide a way of starting over at a point beyond the installation of eDirectory.
2. Create a symbolic link from /var/nds to /shared/var/nds
3. Make a backup copy of /etc/nds.conf.
4. Remove /etc/nds.conf.
5. Create a symbolic link from /etc/nds.conf to /shared/var/nds/nds.conf.
6. Place ndsd in the cluster manager's list of resources to be hosted.
7. Remove ndsd from the list of daemons to be started by the init process at boot time.

After the steps for the primary and secondary nodes are completed, start the cluster services. eDirectory and Identity Manager will start on the primary node.

11.1.5 Identity Manager Driver Considerations

Most Identity Manager drivers can run in a clustered configuration. However, the following items need to be considered:

- The driver executables (.jar files and/or shared objects) must be installed on each cluster node.
- If the driver must run on the same server as the application that the driver supports, then the application must also be configured to run as part of the cluster services.
- If the driver has a configurable location for driver-specific state data, then the location must be on the cluster shared storage.

An example of this is the LDAP driver when used without a change log, or the JDBC driver when used in triggerless mode.

- If the driver has configuration data stored outside of eDirectory, then the configuration data must be on the shared storage or must be duplicated on each cluster node. An example of this is the Manual Task Driver's template directories.

11.2 Case Study for SuSE Linux

For a description of running Identity Manager on shared storage with SUSE LINUX Enterprise Server 8, see TID10093317 (http://support.novell.com/cgi-bin/search/searchtid.cgi?/10093317.htm).

Auditing Identity Manager Licenses

12

The Novell® Identity Manager License Auditing Tool enables you to determine how many of the following are being used:

- Identity Manager licenses in a given tree
- Licenses for specific additional fee drivers
- Novell SecretStore® licenses
- Protocom* SecureLogin licenses

An Identity Manager license is counted for each object that meets the following criteria:

- The object is associated within an Identity Manager driver.
- The Identity Manager driver association has a valid association key.
- The association key isn't marked as disabled.

Additional driver licenses are counted for any object that has one or more valid non-disabled associations to an additional fee driver. This concept is more thoroughly explained in Section 12.3, "Understanding Audit Results," on page 304.

The License Auditing Tool generates reports on demand. You can also schedule audits to run at a later time. After an audit has been scheduled, the License Auditing Tool's graphical interface is locked to prevent tampering. It remains locked until after the audit completes, unless you unlock it by providing a password.

This section contains information about the following topics:

- Section 12.1, "Installing License Auditing Tool," on page 299
- Section 12.2, "Auditing a System," on page 301
- Section 12.3, "Understanding Audit Results," on page 304

12.1 Installing License Auditing Tool

You can install the License Auditing Tool on a Windows or Linux machine. After it is installed, the License Auditing Tool can audit the associations of a specified eDirectory server.

- Section 12.1.1, "Installing on Windows," on page 300
- Section 12.1.2, "Installing on Linux," on page 300
- Section 12.1.3, "Auditing License Usage," on page 300

12.1.1 Installing on Windows

The Identity Manager installation routine installs the License Auditing Tool on Windows by default. If you deselected the Utilities check box during the Identity Manager installation, you need to install the License Auditing Tool separately.

1. Launch the Identity Manager install (`install.exe`) utility.

   Proceed through the installation routine as usual until you get to the *Please Select Components to Install* page.
2. Select *Utilities*, then click *Next*.
3. Specify an installation path, then click *Next*.

   The default location is `C:\Novell\NDS\DirXMLUtilities`.
4. Select the utilities that you want to install, then click *Next*.

   Make sure *License Auditing Tool* is selected.
5. On the Installation Summary page, click *Finish* to install the selected utilities.
6. On the Installation Complete page, click *Close*.

   The License Auditing Tool is now installed.

The License Auditing Tool consists of four `.jar` files (`AuditMain.jar`, `forms_rt.jar`, `ldap.jar`, and `ObjDisabler.jar`) and two `.bat` files (`idmadt.bat` and `idmlat.bat`.) On the Identity Manager CD-ROM or image file, these files are located at `\nt\dirxml\utilities\idm_lat`.

12.1.2 Installing on Linux

Although the License Auditing Tool is available for Linux/UNIX platforms, it is not installed as part of the Installation routine. To install the License Auditing Tool, complete the following steps:

1. Locate the License Auditing Tool files on the Identity Manager CD-ROM or in the image file.

   The License Auditing Tool files are located in `/linux/setup/utilities/idm_lat`.
2. Copy the following files to the Linux/UNIX file system:
   - `AuditMain.jar`
   - `forms_rt.jar`
   - `ldap.jar`
   - `ObjDisabler.jar`
   - `idmadt`
   - `idmlat`

12.1.3 Auditing License Usage

Novell License Management Services analysts help you audit license usage with Identity Manager. By analyzing the data that Identity Manager receives, they find products in use, and help you determine your license ownership and product usage.

To audit your product for license usage, use the tools and instructions listed at License Management Services (http://www.novell.com/licensing/lms/tools.html).

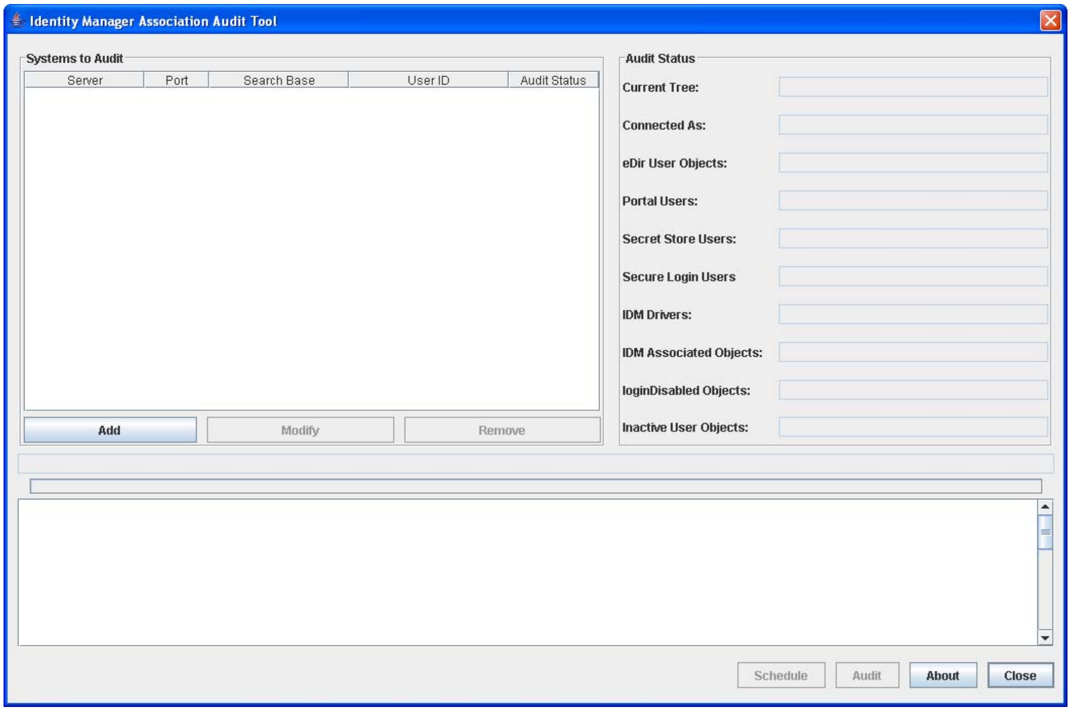
12.2 Auditing a System

Auditing a system with the License Auditing Tool involves the following steps, each of which is described in its own section. After the tool is loaded, the process is identical across all the License Auditing Tool supported platforms.

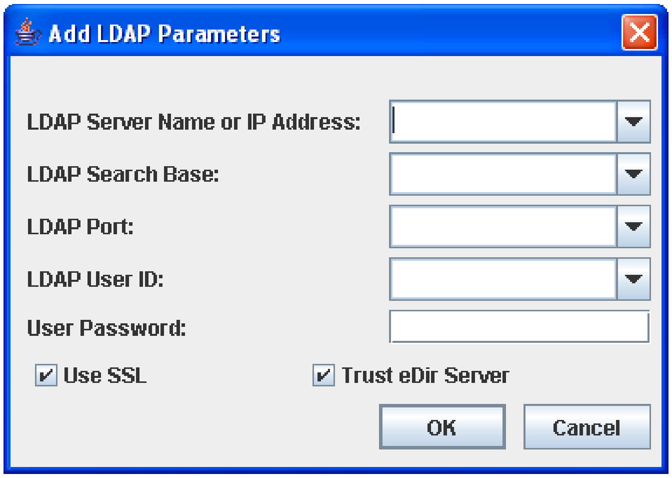
- Section 12.2.1, "Setting the Parameters of an Audit," on page 301
- Section 12.2.2, "Scheduling an Audit," on page 303
- Section 12.2.3, "Unlocking the License Auditing Tool," on page 303
- Section 12.2.4, "Saving Audit Results," on page 304

12.2.1 Setting the Parameters of an Audit

1 Open the License Auditing Tool by launching `idmlat`, which is a `.bat` file for Windows and a script file for Linux/UNIX. The following figure shows the License Auditing Tool user interface.



2 Click *Add* to select a server to audit.



Add LDAP Parameters

LDAP Server Name or IP Address:

LDAP Search Base:

LDAP Port:

LDAP User ID:

User Password:

☑ Use SSL ☑ Trust eDir Server

3 Provide the required information about the target LDAP server, then click *OK*.

LDAP Server Name or IP Address: Specify the LDAP server to which the License Auditing Tool connects in order to audit a directory tree.

LDAP Search Base: Specify the directory container in which the License Auditing Tool performs the audit.

Use a valid LDAP DN (for example `ou=dirxml,o=provo`). Specify `<none>`, or leave this field blank, to start the audit from the root of the tree.

LDAP Port: Specify the port that the License Auditing Tool uses to locate LDAP services on the specified server.

Port 389 is the default LDAP port, and port 636 is the default port for secure LDAP access (via SSL.) However, because the Identity Vault's LDAP ports are configurable, make sure you use valid ports for the specified server.

LDAP User ID: Specify the user ID that the License Auditing Tool uses to connect to the LDAP server.

If you are connecting via SSL, you must specify this parameter.

Make sure that the User ID you specify has access to all objects in the tree. You can specify `anonymous` for this value, but it might not have enough rights to see all the objects for the audit.

User Password: Specify the user password that the License Auditing Tool uses to connect to the LDAP server.

This is the password for the user specified in the LDAP User ID field. If you specified `anonymous` for the LDAP User ID, don't enter a value here.

Use SSL: Select this option to specify that the License Auditing Tool should use SSL when connecting to the LDAP server. This causes the License Auditing Tool to attempt an SSL bind over the port specified in the *LDAP Port* field.

Trust eDir Server: Select this option to specify that the License Auditing Tool can trust the specified eDirectory server.

The License Auditing Tool uses a special feature of the Novell LDAP SDK that informs the LDAP SSL client that the License Auditing Tool already trusts the LDAP server. This means that, when using SSL, the License Auditing Tool doesn't need a copy of the server's certificate.

Because of the context in which License Auditing Tool is used, this is a valid approach, and allows the License Auditing Tool to use SSL without forcing the user to obtain a copy of the server's certificate and also configures the License Auditing Tool to trust the server's certificate.

4 (Optional) Repeat Step 2 and Step 3 as needed to add other LDAP servers on which you want to perform an audit.

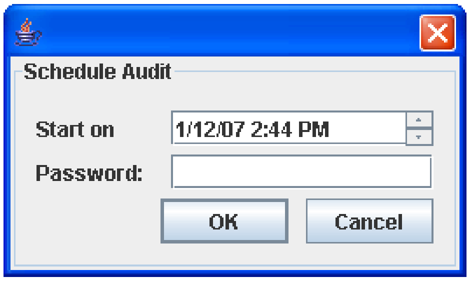
12.2.2 Scheduling an Audit

After you provide the required LDAP parameters, you can run the audit immediately by clicking the *Audit* button. Alternatively, click *Schedule* to configure the audit to run on a specific date and time. Because a tree audit can take quite awhile, depending on the size of the tree, you should schedule the audit to run during off hours if possible.

To schedule an audit, complete the following steps. When you schedule an audit, the License Auditing Tool's interface is locked.

1 In the License Auditing Tool, click *Schedule*.

You must have at least one LDAP server configured in order to schedule an audit.



Schedule Audit

Start on: 1/12/07 2:44 PM

Password:

2 Specify the required scheduling information and click *OK*.

Start on: Specify the date and time to start the audit.

Specify the date and time in the format shown. Alternatively, select one of the numbers in the field and use the arrows on the right side of the field to increase or decrease that value.

Password: Specify a password for this audit.

The password you enter here becomes the key to unlock the Auditing Tool.

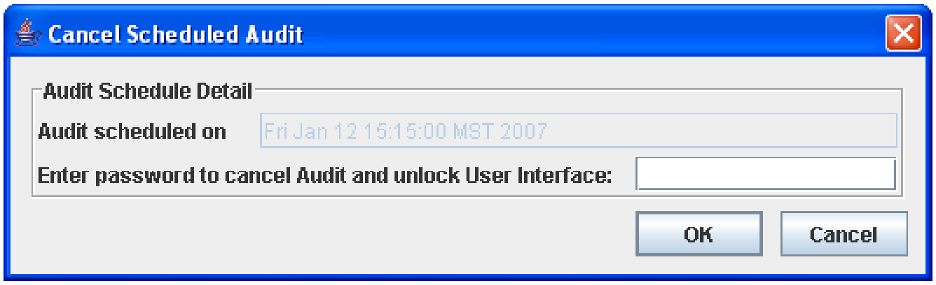
When you schedule an audit, the License Auditing Tool's interface locks to prevent anyone tampering with the audit parameters or results. If you need to unlock the interface to modify audit schedule or parameters, see "Unlocking the License Auditing Tool" on page 303."

12.2.3 Unlocking the License Auditing Tool

When you schedule an audit, the License Auditing Tool's interface locks to prevent anyone tampering with the audit parameters or results. To unlock the License Auditing Tool interface to modify the audit schedule or parameters, you must use the password you specified when you scheduled the audit.

Unlocking the License Auditing Tool interface prior to a scheduled audit terminates the currently scheduled audit.

1 In the License Auditing Tool, click *Unlock*.



Cancel Scheduled Audit

Audit Schedule Detail

Audit scheduled on: Fri Jan 12 15:15:00 MST 2007

Enter password to cancel Audit and unlock User Interface:

2 Enter the audit password and click *OK*.

12.2.4 Saving Audit Results

As the License Auditing Tool performs an audit, it displays results in the Audit Status window. Additionally, a complete audit report displays in the text window at the bottom of the License Auditing Tool interface. You can review the audit results using these two windows. Because of the potential length of the report, the License Auditing Tool also saves the audit data to a series of text files.

treename-summary.log: Contains the complete audit report.

treename-logindisabled.log: (Optional) Contains a list of DNS object names that are login disabled.

treename-inactiveusers.log: (Optional) Contains a list of DNS object names that haven't authenticated in over a year.

12.3 Understanding Audit Results

The major components of the audit report include the following:

Parameter Summary: Displays the parameters for the audit, including LDAP server information and the directory tree being audited.

Audit Results Summary: Displays a summary of the Identity Manager audit findings, including the number of associated objects, which corresponds to the number of Identity Manager base licenses being used in the tree.

Object Class Summary: Displays object associations by object class.

Driver Association Summary: Displays object associations by Identity Manager driver. If the License Auditing Tool recognizes the driver it displays the driver name. Otherwise, is identifies the driver only as `<Custom>`.

Driver Summary: Displays a summary for each Identity Manager driver identified in the audit, including driver name and context, driver module, and the number of processed and disabled object associations for the driver.

An audit report looks similar to the following:

```
Identity Manager License Auditing Tool v1.4
Novell, Inc. Copyright 2001 - 2007

Audit started Wednesday, November 13, 2007 at 10:49 AM

Parameter Summary:
  LDAP Server Name: 10.1.1.222
  LDAP Server Port: 636
  Search Base: (null)
  Connected as: cn=admin,o=lab1
  Tree Name: LABTEST

Audit Results Summary:
  DLAT found 7061 user objects
  DLAT found 0 SecretStore users
  DLAT found 0 SecureLogin users
  DLAT found 26 DirXML Drivers
  DLAT found 12664 associated objects

Object Class Summary
  7060 associations to Object Class inetOrgPerson
  5604 associations to Object Class costCenter

Driver Association Summary
  7058 associations to DirXML Driver for eDirectory drivers
  4408 associations to DirXML Driver for Peoplesoft (Consulting Release)
drivers
  12473 associations to <Custom> drivers
  12626 associations to DirXML Driver for JDBC drivers
  12643 associations to DirXML Driver for Peoplesoft drivers

Driver: CN=NDSTONDS - PRV-NDS4,CN=NDS DRIVERS,O=SERVICES
  Driver Name: DirXML Driver for eDirectory
  Driver Module: com.novell.nds.dirxml.driver.nds.drivershimimpl
  Processed Associations: 1162
  Disabled Associations: 2

Driver: CN=WSE - SECURITY,CN=TELECOM DRIVER SET,O=SERVICES
  Driver Name: <Custom>
  Driver Module:com.novell.nds.dirxml.driver.wsejdbc.wsedrivershim
  Processed Associations: 7033
  Disabled Associations: 1

Driver: CN=PS8 DRIVER,CN=PEOPLESOFT DRIVER,O=SERVICES
  Driver Name: DirXML Driver for Peoplesoft
  Driver Module: npsshim.dll
  Processed Associations: 7039
  Disabled Associations: 2

Driver: CN=BIG USERS DRIVER,CN=BIG EWORKS PSCOSTCENTER,O=SERVICES
  Driver Name: DirXML Driver for JDBC
  Driver Module: com.novell.nds.dirxml.driver.jdbc.jdbcdrivershim
  Processed Associations: 6978
  Disabled Associations: 1

Driver: CN=PS8 COSTCENTER DRIVER,CN=BIG EWORKS PSCOSTCENTER,O=SERVICES
  Driver Name: DirXML Driver for Peoplesoft
  Driver Module: npsshim.dll
  Processed Associations: 5604
  Disabled Associations: 0

Audit completed Wednesday, November 13, 2007 at 12:16 PM
```

DirXML Command Line Utility A

The DirXML® Command Line utility allows you to use a command line interface to manage the driver. You can create scripts to manage the driver with the commands.

The utility and scripts are installed on all platforms during the Identity Manager installation. The utility is installed to the following locations:

- Windows: `\Novell\Nds\dxcmd.bat`
- NetWare®: `sys:\system\dxcmd.ncf`
- UNIX: `/usr/bin/dxcmd`

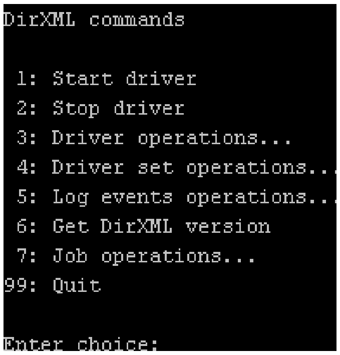
There are two different methods for using the DirXML Command Line utility:

- Section A.1, "Interactive Mode," on page 307
- Section A.2, "Command Line Mode," on page 316

A.1 Interactive Mode

The interactive mode provides a text interface to control and use the DirXML Command Line utility.

1. At the console, enter `dxcmd`.
2. Enter the name of a user with sufficient rights to the Identity Manager objects, such as admin.novell.
3. Enter the user's password.



```
DirXML commands

 1: Start driver
 2: Stop driver
 3: Driver operations...
 4: Driver set operations..
 5: Log events operations..
 6: Get DirXML version
 7: Job operations...
99: Quit

Enter choice:
```

4. Enter the number of the command you want to perform.

   Table A-1 on page 308 contains the list of options and what functionality is available.
5. Enter 99 to quit the utility.

NOTE: If you are running eDirectory™ 8.8 on UNIX or Linux, you must specify the -host and -port parameters. For example, `dxcmd -host 10.0.0.1 -port 524`. If the parameters are not specified, a jclient error occurs.

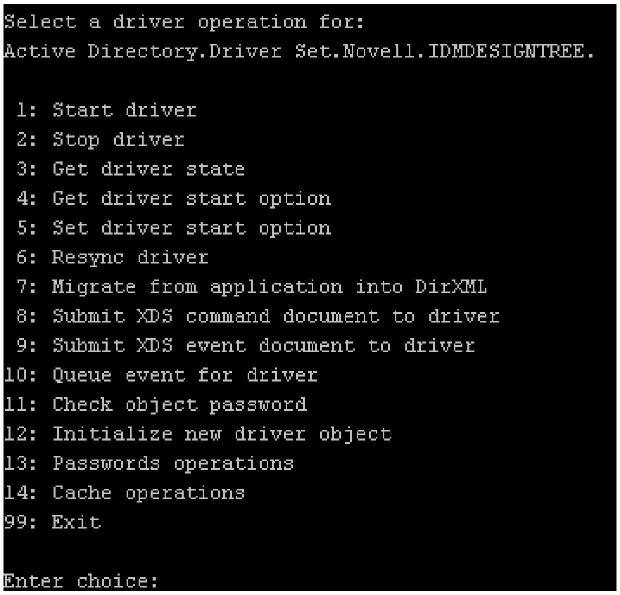
```
novell.jclient.JCException: connect (to address) 111 UNKNOWN ERROR
```

By default, eDirectory 8.8 is not listening to localhost. The DirXML Command Line utility needs to resolve the server IP address or hostname and the port to be able to authenticate.

Table A-1 *Interactive Mode Options*

Option	Description
1: *Start Driver*	Starts the driver. If there is more than one driver, each driver is listed with a number. Enter the number of the driver to start the driver.
2: *Stop Driver*	Stops the driver. If there is more than one driver, each driver is listed with a number. Enter the number of the driver to stop the driver.
3: *Driver operations*	Lists the operations available for the driver. If there is more than one driver, each driver is listed with a number. Enter the number of the driver to see the operations available. See Table A-2 on page 309 for a list of operations.
4: *Driver set operations*	Lists the operations available for the driver set. ◆ 1: Associate driver set with server ◆ 2: Disassociate driver set from server ◆ 99: Exit
5: *Log events operations*	Lists the operations available for logging events through Novell® Audit. See Table A-5 on page 313 for a description of these options.
6: *Get DirXML version*	Lists the version of the Identity Manager installed.
7: *Job operations*	Manages jobs created for Identity Manager.
99: *Quit*	Exits the DirXML Command Line utility

Figure A-1 *Driver Options*



```
Select a driver operation for:
Active Directory.Driver Set.Novell.IDMDESIGNTREE.

 1: Start driver
 2: Stop driver
 3: Get driver state
 4: Get driver start option
 5: Set driver start option
 6: Resync driver
 7: Migrate from application into DirXML
 8: Submit XDS command document to driver
 9: Submit XDS event document to driver
10: Queue event for driver
11: Check object password
12: Initialize new driver object
13: Passwords operations
14: Cache operations
99: Exit

Enter choice:
```

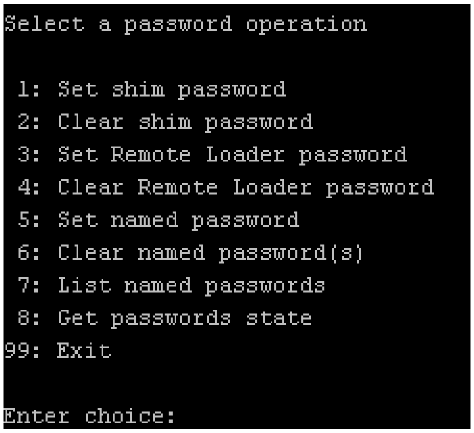
Table A-2 *Driver Options*

Options	Description
1: *Start driver*	Starts the driver.
2: *Stop driver*	Stops the driver.
3: *Get driver state*	Lists the state of the driver. ◆ 0 - Driver is stopped ◆ 1 - Driver is starting ◆ 2 - Driver is running ◆ 3 - Driver is stopping
4: *Get driver start option*	Lists the current driver start option. ◆ 1 - Disabled ◆ 2 - Manual ◆ 3 - Auto
5: *Set driver start option*	Changes the start option of the driver. ◆ 1 - Disabled ◆ 2 - Manual ◆ 3 - Auto ◆ 99 - Exit
6: *Resync driver*	Forces a resynchronization the driver. It prompts for a time delay: *Do you want to specify a minimum time for resync? (yes/no).* If you enter Yes, specify the date and time you want the resynchronization to occur: *Enter a date/time (format 9/27/05 3:27 PM)*. If you enter No, the resynchronization occurs immediately.
7: *Migrate from application into DirXML*	Processes an XML document that contains a query command: *Enter filename of XDS query document:* Create the XML document that contains a query command by using the Novell `nds.dtd` (http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/ref/ndsdtd/query.html). Examples: NetWare: `sys:\files\query.xml` Windows: `c:\files\query.xml` Linux: `/files/query.xml`

Options	Description
8: *Submit XDS command document to driver*	Processes an XDS command document: *Enter filename of XDS command document:* Examples: NetWare: `sys:\files\user.xml` Windows: `c:\files\user.xml` Linux: `/files/user.xml` *Enter name of file for response:* Examples: NetWare: `sys:\files\user.log` Windows: `c:\files\user.log` Linux: `/files/user.log`
9: Submit XDS event document to driver	Processes and XDS event document: *Enter filename of XDS event document:* Examples: NetWare: `sys:\files\add.xml` Windows: `c:\files\add.xml` Linux: `/files/add.xml`
10: Queue event for driver	Adds an event to the driver queue: *Enter filename of XDS event document:* Examples: NetWare: `sys:\files\add.xml` Windows: `c:\files\add.xml` Linux: `/files/add.xml`
11: *Check object password*	Validates that an object's password in the connected system is associated with a driver. It matches the object's eDirectory password (Distribution Password, used with Universal Password). *Enter user name*:
12: *Initialize new driver object*	Performs an internal initialization of data on a new Driver object. This is only for testing purposes.
13: *Password operations*	There are nine Password options. See Table A-3 on page 311 for a description of these options.
14: *Cache operations*	There are five Cache operations. See Table A-4 on page 312 for a descriptions of these options.

Options	Description
99: *Exit*	Exits the driver options.

Figure A-2 *Password Operations*



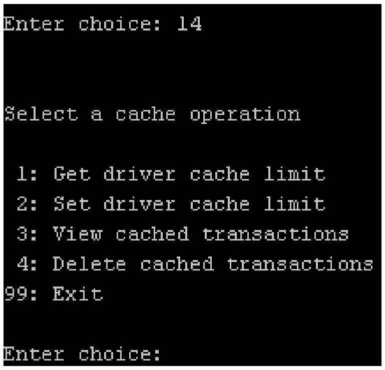
```
Select a password operation

 1: Set shim password
 2: Clear shim password
 3: Set Remote Loader password
 4: Clear Remote Loader password
 5: Set named password
 6: Clear named password(s)
 7: List named passwords
 8: Get passwords state
99: Exit

Enter choice:
```

Table A-3 *Password Operations*

Operation	Description
1: *Set shim password*	Sets the application password. This is the password of the user account you are using to authenticate into the connected system with.
2: *Clear shim password*	Clears the application password.
3: *Set Remote Loader password*	The Remote Loader password is used to control access to the Remote Loader instance. Enter the Remote Loader password, then confirm the password by typing it again.
4: *Clear Remote Loader password*	Clears the Remote Loader password so no Remote Loader password is set on the Driver object.
5: *Set named password*	Allows you to store a password or other pieces of security information on the driver. See Section 2.11, "Using Named Passwords," on page 37 for more information. There are four prompts to fill in: ◆ *Enter password name:* ◆ *Enter password description:* ◆ *Enter password:* ◆ *Confirm password*

Operation	Description
6: *Clear named passwords*	Clears a specified Named Password or all Named Passwords that are stored on the driver object: *Do you want to clear all named passwords? (yes/no).* If you enter Yes, all Named Passwords are cleared. If you enter No, you are prompted to specify the password name that you want to clear.
7: *List named passwords*	Lists all Named Passwords that are stored on the driver object. It lists the password name and the password description.
8: *Get password state*	Lists if a password is set for: ◆ Driver Object password: ◆ Application password: ◆ Remote loader password: The dxcmd utility allows you to set the Application password and the Remote Loader password. You cannot set the Driver Object password with this utility. It shows if the password has been set or not.
99: *Exit*	Exits the current menu and takes you back to the Driver options.

Figure A-3 *Cache Operations*



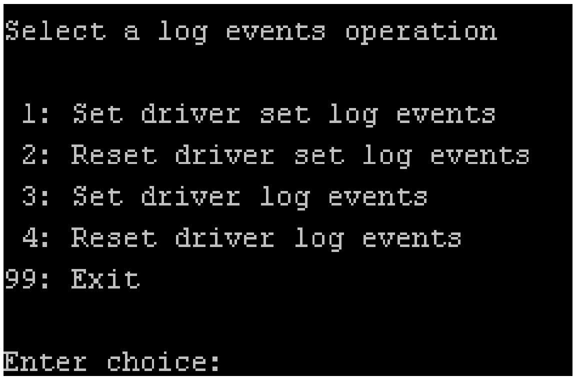
```
Enter choice: 14


Select a cache operation

 1: Get driver cache limit
 2: Set driver cache limit
 3: View cached transactions
 4: Delete cached transactions
99: Exit

Enter choice:
```

Table A-4 *Cache Operations*

Operation	Description
1: *Get driver cache limit*	Displays the current cache limit that is set for the driver.
2: *Set driver cache limit*	Sets the driver cache limit in kilobytes. A value of 0 is unlimited.

Operation	Description
3: *View cached transactions*	A text file is created with the events that are stored in cache. You can select the number of transactions to view. ◆ *Enter option token* (default=0): ◆ *Enter maximum transactions records to return* (default=1): ◆ *Enter name of file for response:*
4: *Delete cached transactions*	Deletes the transactions stored in cache. ◆ *Enter position token* (default=0): ◆ *Enter event-id value of first transaction record* to delete (optional): ◆ *Enter number of transaction records to delete* (default=1):
99: *Exit*	Exits the current menu and takes you back to the Driver options.

Figure A-4 *Log Event Operations*



```
Select a log events operation

  1: Set driver set log events
  2: Reset driver set log events
  3: Set driver log events
  4: Reset driver log events
99: Exit

Enter choice:
```

Table A-5 *Log Events Operations*

Operation	Description
1: *Set driver set log events*	Allows you to log driver set events through Novell Audit. There are 49 items you can select to log. See Table A-6 on page 314 for a list of these options. Type the number of the item you want to log. After the items are selected, type 99 to accept the selections.
2: *Reset driver set log events*	Resets all of the log event options.

Operation	Description
3: *Set driver log events*	Allows you to log driver events through Novell Audit. There are 49 items to select to log. See Table A-6 on page 314 for a list of these options. Type the number of the item you want to log. After the items are selected, type 99 to accept the selections.
4: *Reset driver log events*	Resets all of the log event options.
99: *Exit*	Exits the log events operations menu.

Table A-6 *Driver Set and Driver Log Events*

Options
1: Status success
2: Status retry
3: Status warning
4: Status error
5: Status fatal
6: Status other
7: Query elements
8: Add elements
9: Remove elements
10: Modify elements
11: Rename elements
12: Move elements
13: Add-association elements
14: Remove-association elements
15: Query-schema elements
16: Check-password elements
17: Check-object-password elements
18: Modify-password elements
19: Sync elements
20: Pre-transformed XDS document from shim
21: Post input transformation XDS document
22: Post output transformation XDS document
23: Post event transformation XDS document

Options
24: Post placement transformation XDS document
25: Post create transformation XDS document
26: Post mapping transformation <inbound> XDS document
27: Post mapping transformation <outbound> XDS document
28: Post matching transformation XDS document
29: Post command transformation XDS document
30: Post-filtered XDS document <Publisher>
31: User agent XDS command document
32: Driver resync request
33: Driver migrate from application
34: Driver start
35: Driver stop
36: Password sync
37: Password request
38: Engine error
39: Engine warning
40: Add attribute
41: Clear attribute
42: Add value
43: Remove value
44: Merge entire
45: Get named password
46: Reset Attributes
47: Add Value - Add Entry
48: Set SSO Credential
49: Clear SSO Credential
50: Set SSO Passphrase
51: User defined IDs
99: Accept checked items

Table A-7 *Enter Table Title Here*

Options	Description
1: *Get available job definitions*	Allows you to select an existing job. *Enter the job number:* *Do you want to filter the job definitions by containment?* Enter Yes or No *Enter name of the file for response:* Examples: NetWare: `sys:\files\user.log` Windows: `c:\files\user.log` Linux: `/files/user.log`
2: *Operations on specific job object*	Allows you to perform operations for a specific job.

A.2 Command Line Mode

The command line mode allows you to use script or batch files. Table A-8 on page 316 contains the different options that are available.

To use the command line options, decide which items you want to use and string them together.

Example: `dxcmd -user admin.headquarters -host 10.0.0.1 -password n0vell -start test.driverset.headquarters`

This example command starts the driver.

Table A-8 *Command Line Options*

Option	Description
Configuration	
-user *<user name>*	Specify the name of a user with administrative rights to the drivers you want to test.
-host *<name or IP address>*	Specify the IP address of the server where the driver is installed.
-password *<user password>*	Specify the password of the user specified above.
-port *<port number>*	Specify a port number, if the default port is not used.
-q *<quiet mode>*	Displays very little information when a command is executed.
-v *<verbose mode>*	Displays detailed information when a command is executed.
-s *<stdout>*	Writes the results of the dxcmd command to stdout.

Option	Description
-? *<show this message>*	Displays the help menu.
-help <*show this message*>	Displays the help menu.
Actions	
-start <*driver dn*>	Starts the driver.
-stop <*driver dn*>	Stops the driver.
-getstate <*driver dn*>	Shows the state of the driver as running or stopped.
-getstartoption <*driver dn*>	Shows the startup option of the driver.
-setstartoption <*driver dn*> <*disabled*|*manual*|*auto*> <*resync*|*noresync*>	Sets how the driver starts if the server is rebooted. Sets whether the objects are to be resynchronized when the driver restarts.
-getcachelimit <*driver dn*>	Lists the cache limit set for the driver.
-setcachelimit <*driver dn*> <*0 or positive integer*>	Sets the cache limit for the driver.
-migrateapp <*driver dn*> <*filename*>	Processes an XML document that contains a query command. Create the XML document that contains a query command by using the Novell `nds.dtd` (http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_dtd/data/dtdndsoverview.html#dtdndsoverview).
-setshimpassword <*driver dn*> <*password*>	Sets the application password. This is the password of the user account you are using to authenticate into the connected system with.
-clearshimpassword <*driver dn*> <*password*>	Clears the application password.
-setremoteloaderpassword <*driver dn*> <*password*>	Sets the Remote Loader password. The Remote Loader password is used to control access to the Remote Loader instance.
<clearremoteloaderpassword <*driver dn*>	Clears the Remote Loader password.

Option	Description
-sendcommand *<driver dn> <input filename> <output filename>*	Processes an XDS command document. Specify the XDS command document as the input file. Examples: NetWare: `sys:\files\user.xml` Windows: `c:\files\user.xml` Linux: `/files/user.log` Specify the output filename to see the results. Examples: NetWare: `sys:\files\user.log` Windows: `c:\files\user.log` Linux: `/files/user.log`
-sendevent *<driver dn> <input filename>*	Submits a document to the driver's Subscriber channel, bypassing the driver cache. The document gets processed ahead of anything that might be in the cache at the time of the submission. It also means that the submission fails if the driver is not running.
-queueevent *<driver dn> <input filename>*	Submits a document to the driver's Subscriber channel by queuing the document in the driver cache. The document gets processed after anything that might be in the cache at the time of the submission. The submission won't fail if the driver isn't running.
-setlogevents *<dn> <integer ...>*	Sets Novell Audit log events on the driver. The integer is the option of the item to log. See Table A-6 on page 314 for the list of the integers to enter.
-clearlogevents *<dn>*	Clears all Novell Audit log events that are set on the driver.
-setdriverset *<driver set dn>*	Associates a driver set with the server.
-cleardriverset	Clears the driver set association from the server.
-getversion	Shows the version of Identity Manager that is installed.
-initdriver object *<dn>*	Performs an internal initialization of data on a new Driver object. This is only for testing purposes.
-setnamedpassword *<driver dn> <name> <password> [description]*	Sets Named Passwords on the driver object. You specify the name, the password, and the description of the Named Password.
-clearnamedpassword *<driver dn> <name>*	Clears a specified Named Password.
-startjob *<job dn>*	Starts the specified job.

Option	Description
-abortjob *<job dn>*	Aborts the specified job.
-getjobrunningstate *<job dn>*	Returns the specified job's running state.
-getjobenabledstate *<job dn>*	Returns the specified job's enabled state.
-getjobnextruntime *<job dn>*	Returns the specified job's next run time.
-updatejob *<job dn>*	Updates the specified job.
-clearallnamedpaswords *<driver dn>*	Clears all Named Passwords set on a specific driver.

If a command line is executed successfully, it returns a zero. If the command line returns anything other than zero, it is an error. For example 0 means success, and -641 means invalid operation. -641 is an eDirectory error code. Table A-9 on page 319 contains other values for specific command line options.

Table A-9 *Command Line Option Values*

Command Line Option	Values
-getstate	0- stopped
	1- starting
	2- running
	3- shutting down
	11- get schema
	Anything else that is returned is an error.
-getstartoption	0- disabled
	1- manual
	2- auto
	Anything else that is returned is an error.
-getcachelimit	0- unlimited
	Anything else that is returned is an error.
-getjobrunningstate	0- stopped
	1- running
	Anything else that is returned is an error.
-getjobenabledstate	0- disabled
	1- enabled
	2- configuration error
	Anything else that is returned is an error.

Command Line Option	Values
-getjobnextruntime	Return is the next scheduled time for the job in eDirectory time format (number of seconds since 00:00:00 Jan 1, 1970UTC).

Options for Configuring a Remote Loader B

The options in the following table enable you to configure a Remote Loader.

Table B-1 *Remote Loader Options*

Option	Secondary Name	Parameter	Description
address		IP address	An optional parameter. Specifies that the Remote Loader listens on a particular local IP address. This is useful if the server hosting the Remote Loader has multiple IP addresses and the Remote Loader must listen on only one of the addresses. You have three options: address=*address number* address='localhost' Don't use this parameter. If you don't use the address, the Remote Loader listens on all local IP addresses. Example: address=137.65.134.83
-class	-cl	Java class name	Specifies the Java class name of the Identity Manager application shim that is to be hosted. For example, for a Java driver, type one of the following: -class com.novell.nds.dirxml.driver.ldap.LDAPDriverShim -cl com.novell.nds.dirxml.driver.ldap.LDAPDriverShim Java uses a keystore to read certificates. The -class option and the -module option are mutually exclusive. To see a list of the Java class name see Table B-2 on page 328.

Option	Secondary Name	Parameter	Description
-commandport	-cp	port number	Specifies the TCP/IP port that the Remote Loader instance uses for control purposes. If the Remote Loader instance is hosting an application shim, the command port is the port on which another Remote Loader instance communicates with the instance that is hosting the shim. If the Remote Loader instance is sending a command to an instance that is hosting an application shim, the command port is the port on which the hosting instance is listening. If not specified, the default command port is 8000. Multiple instances of the Remote Loader can run on the same server hosting different driver instances by specifying different connection ports and command ports. Example: -commandport 8001 -cp 8001
-config	None	filename	Specifies a configuration file. The configuration file can contain any command line options except `config`. Options specified on the command line override options specified in the configuration file. Example: -config config.txt
-connection	-conn	connection configuration string	Specifies the connection parameters for the connection to the Metadirectory server running the Identity Manager remote interface shim. The default connection method for the Remote Loader is TCP/IP using SSL. The default TCP/IP port for this connection is 8090. Multiple instances of the Remote Loader can run on the same server. Each instance of the Remote Loader hosts a separate Identity Manager application shim instance. Differentiate multiple instances of the Remote Loader by specifying different connection ports and command ports for each Remote Loader instance. Example: -connection "port=8091 rootfile=server1.pem" -conn "port=8091 rootfile=server1.pem"

Option	Secondary Name	Parameter	Description
-description	-desc	short description	Specify a short description string (for example, SAP) to be used for the trace window title and for Novell® Audit logging. Example: -description SAP -desc SAP The Remote Loader Console places long forms in the configuration files. You can use either a long form (for example, -description) or a short form (for example, -desc).

fromaddress	None	IP address	The Remote Loader only accepts connections from the specified IP address. Any other connections are not allowed.
			Example:
			`--conn “port=8092 fromaddress=10.0.0.2”`
			or
			`-connection “port=8094 fromaddress=metaserver1.company.com”`

handshaketimeout	None	number of milliseconds	Increases the time out period of the handshake between the Remote Loader and the Metadirectory engine. Example: `--connection “port=8093 handshaketimeout=1000”` The value is some integer greater than or equal to zero. Zero means never time out. The non-zero number is the number of milliseconds for the time out to occur. The default value is 1000 milliseconds.
-help	-?	None	Displays help. Example: -help -?
-java	-j	None	Specifies that the passwords are to be set for a Java shim instance. This option is only useful in conjunction with the setpasswords option. If -class is specified with -setpasswords, this option isn't necessary.

Option	Secondary Name	Parameter	Description
-javadebugport	-jdp	Port number	Specifies that the Remote Loader instance is to enable Java debugging on the specified port. This is useful for developers of the Identity Manager application shims. Example: -javadebugport 8080 -jdp 8080

keystore			Conditional parameters. Used only for Identity Manager application shims contained in .jar files.
			Specifies the filename of the Java keystore that contains the trusted root certificate of the issuer of the certificate used by the remote interface shim. This is typically the Certificate Authority of the eDirectory™ tree that is hosting the remote interface shim.
			If you are running SSL and need the Remote Loader to communicate with a Java driver, type a key-value pair:
			`keystore='`*`keystorename`*`'`
			`storepass='`*`password`*`'`
			NOTE: For more information on creating the keystore option, see Section 3.8.3, "Creating a Keystore," on page 87

-module	-m	modulename	Specifies the module containing the Identity Manager application shim that is to be hosted. For example, for a native driver, type one of the following: -module "c:\Novell\RemoteLoader\Exchange5Shim.dll" -m "c:\Novell\RemoteLoader\Exchange5Shim.dll" or -module "usr/lib/dirxml/NISDriverShim.so" -m "usr/lib/dirxml/NISDriverShim.so" The -module option uses a rootfile certificate. The -module option and the -class option are mutually exclusive.

Option	Secondary Name	Parameter	Description
-password	-p	password	Specifies the password for command authentication. This password must be the same as the first password specified with `setpasswords` for the loader instance being commanded. If a command option (for example, unload or tracechange) is specified and the `password` option isn't specified, the user is prompted to enter the password for the loader that is the target of the command. Example: -password novell4 -p novell4
port		decimal port number	A required parameter. It specifies the TCP/IP port on which the Remote Loader listens for connections from the remote interface shim. Example: port=8090
rootfile			A conditional parameter. If you are running SSL and need the Remote Loader to communicate with a native driver, type `rootfile='`*`trusted certname`*`'`
-service	-serv	None, or install/ uninstall	To install an instance as a service, use the install argument together with any other arguments necessary to host an application shim. For example, the arguments used must include -module, but any argument can include -connection, -commandport, and so forth. This option installs the Win32 service but doesn't start the service. To uninstall an instance running as a service, use the uninstall argument together with any other arguments necessary to host the application shim. The no-argument version of this option is only used on the command line to an instance being run as a Win32 service. This is automatically set up when installing an instance as a service. Example: -service install -serv uninstall This option isn't available on rdxml or the Java Remote Loader.

Option	Secondary Name	Parameter	Description
-setpasswords	-sp	password password	Specifies the password for the Remote Loader instance and the password of the Identity Manager Driver object of the remote interface shim that the Remote Loader communicates with. The first password in the argument is the password for the Remote Loader. The second password in the optional arguments is the password for the Identity Manager Driver object associated with the remote interface shim on the Metadirectory server. Either no password or both passwords must be specified. If no password is specified, the Remote Loader prompts for the passwords. This is a configuration option. Using this option configures the Remote Loader instance with the passwords specified but doesn't load a Identity Manager application shim or communicate with another loader instance. Example: -setpasswords novell4 staccato3 -sp novell4 staccato3
-storepass		storepass	Used only for Identity Manager application shims contained in .jar files. Specifies the password for the Java keystore specified by the keystore parameter. Example: storepass=mypassword This option applies only to the Java Remote Loader.
-trace	-t	integer	Specifies the trace level. This is only used when hosting an application shim. Trace levels correspond to those used on the metadirectory server. Example: -trace 3 -t 3
-tracechange	-tc	integer	Commands a Remote Loader instance that is hosting an application shim to change its trace level. Trace levels correspond to those used on the metadirectory server. Example: -tracechange 1 -tc 1

Option	Secondary Name	Parameter	Description
-tracefile	-tf	filename	Specify a file to write trace messages to. Trace messages are written to the file if the trace level is greater than zero. Trace messages are written to the file even if the trace window is not open. Example: -tracefile c:\temp\trace.txt -tf c:\temp\trace.txt
-tracefilechange	-tfc	None, or filename	Commands a remote Loader instance that is hosting an application shim to start using a trace file, or to close one already in use and use a new one. Using the no-argument version of this option causes the hosting instance to close any trace file being used. Example: -tracefilechange c:\temp\newtrace.txt tfc c:\temp\newtrace.txt
-tracefilemax	-tfm	size	Specifies the approximate maximum size that trace file data can occupy on disk. If you specify this option, there is a trace file with the name specified using the tracefile option and up to 9 additional “roll-over” files. The roll-over files are named using the base of the main trace filename plus “_*n*”, where n is 1 through 9. The size parameter is the number of bytes. Specify the size by using the suffixes K, M, or G for kilobytes, megabytes, or gigabytes. If the trace file data is larger than the specified maximum when the Remote Loader is started, the trace file data remains larger than the specified maximum until roll-over is completed through all 10 files Example: -tracefilemax 1000M -tfm 1000M In this example, the trace file can be only 1 GB.
-unload	-u	None	Unloads the Remote Loader instance. If the Remote Loader is running as a Win32 Service, this command stops the service. Example: -unload -u

Option	Secondary Name	Parameter	Description
-window	-w	On/Off	Turns the trace window on or off in a Remote Loader instance. Example: -window on -w off This option is available only on Windows platforms. It isn't available on the Java Remote Loader.
-wizard	-wiz	None	Launches the Configuration Wizard. Running dirxml_remote.exe with no command line parameters also launches the wizard. This option is useful if a configuration file is also specified. In this case, the wizard starts with values from the configuration file and the wizard can be used to change the configuration without editing the configuration file directly. Example: -wizard -wiz This option is available only on Windows platforms. It isn't available on the Java Remote Loader.

Table B-2 *Java Class Names*

Java Class Name	Driver
com.novell.nds.dirxml.driver.avaya.PBXDriverShim	Avaya PBX Driver
com.novell.nds.dirxml.driver.delimitedtext.DelimitedTextDriver	Delimited Text Driver
com.novell.nds.dirxml.driver.nds.DriverShimImpl	eDirectory Driver
com.novell.nds.dirxml.driver.entitlement.EntitlementServiceDriver	Entitlement Services Driver
com.novell.gw.dirxml.driver.gw.GWdriverShim	GroupWise Driver
com.novell.nds.dirxml.driver.jdbc.JDBCDriverShim	JDBC Driver
com.novell.idm.driver.jms.JMSDriverShim	JMS Driver
com.novell.nds.dirxml.driver.ldap.LDAPDriverShim	LDAP Driver
com.novell.nds.dirxml.driver.loopback.LoopbackDriverShim	GenericLoopBack Driver
com.novell.nds.dirxml.driver.manualtask.driver.ManualTaskDriver	Manual Task Driver
com.novell.nds.dirxml.driver.nisdriver.NISDriverShim	NIS Driver
com.novell.nds.dirxml.driver.notes.NotesDriverShim	Notes Driver

Java Class Name	Driver
com.novell.nds.dirxml.driver.psoftshim.PSOFTDriverShim	PeopleSoft Driver
com.novell.nds.dirxml.driver.SAPShim.SAPDriverShim	SAP HR Driver
com.novell.nds.dirxml.driver.sapusershim.SAPDriverShim	SAP User Management Driver
com.novell.nds.dirxml.driver.sifagent.SIFShim	SIF Driver
com.novell.nds.dirxml.driver.soap.SOAPDriver	Soap Driver
com.novell.idm.driver.ComposerDriverShim	User Appliction
be.opns.dirxml.driver.ars.arsremedydrivershim.ARSDriverShim	Driver for Remedy ARS
com.novell.nds.dirxml.driver.workorder.WorkOrderDriverShim	WorkOrder

Editing Driver Configuration Files

C

You must have a good knowledge of XML to use the information is this section. This information allows you to add custom prompts to drivers you have created.

- Section C.1, "Variables in a Driver Configuration File," on page 331
- Section C.2, "Flexible Prompting in a Driver Configuration File," on page 334
- Section C.3, "The Informal Identity Manager 3.5 Driver Configuration DTD," on page 336

C.1 Variables in a Driver Configuration File

For the iManager plug-ins, several node types are defined for the driver configuration files. The following is a list of actions that the Identity Manager engine supports:

- Prompting once for a value that is used repeatedly throughout a single driver configuration file.
- Prompting once for a value that is used across multiple driver configuration files, as part of the Import Drivers Wizard.
- Allowing the user to select a value from a drop-down list of values.
- Global modification of the driver configuration files according to a contained XSL style sheet.
- Built-in variables that can be referenced without declaring them to access information about the driver and its environment. For example, tree name, driver set name, driver set DN, server name, server DN, driver name, and driver DN.
- The ability to layer prompts. It is possible to ask the user multiple sets of questions, with the second and later sets being controlled by the user's responses to prior sets of questions. For more information, see Section C.2, "Flexible Prompting in a Driver Configuration File," on page 334.

The primary new node types are:

- **variable-decl:** Allows you to define driver configuration variables that are prompted for and placed into a driver configuration file during its import. Multiple `variable-decl` blocks can be used to define a layered set of prompts. For more information, see Section C.2, "Flexible Prompting in a Driver Configuration File," on page 334.
- **variable-ref:** Used to reference a variable defined in a `variable-decl` within your driver configuration files.
- **xsl-modify:** Used to globally modify the driver configuration file after all variables and prompts have been resolved. The contents of this node are extracted and used as an XSL style sheet that is applied to the patched driver configuration file.

To view the driver configuration file XML extensions, see DriverConfigXMLExtension.txt (../samples/DriverConfigXMLExtension.txt).

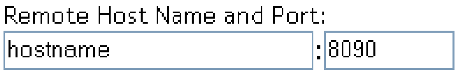
In addition, be aware of the following notes:

- Section C.1.1, "General Notes," on page 332
- Section C.1.2, "Import Driver Notes," on page 334

C.1.1 General Notes

- The `variable-decls` can contain `text-var` but not `node-var`. It can contain `variable-refs` as long as the order they are resolved is taken into account.
- If a `variable-decl` contains an optional prompt attribute and an optional prompt-type attribute and does not contain an optional `browse=“yes”` attribute setting, the prompt-type is treated as follows:
  - A prompt-type of `"ipa"` results in two edit fields. See Figure C-1 for an example. The value the user specifies for the first part is appended by a colon (:) and the value the user specifies for the second part in the value is rendered by the variable.

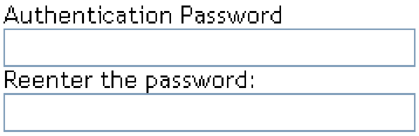
Figure C-1 *Two Edit Fields*



Remote Host Name and Port:
hostname : 8090

  - A prompt-type of `"password"` results in two password edit fields. See Figure C-2 for an example. The first prompt is for the actual password, and the second prompt is used to verify that the password specified in the first field is correct. The value rendered by the variable reference is the password.

Figure C-2 *Two Password Fields*



Authentication Password

Reenter the password:

  - A prompt-type of `"hidden"` results in a field that is not displayed, but is checked to make sure a previous condition is met before proceeding to the next screen.
  - Any other prompt-type value is ignored.
- If a `variable-decl` contains an optional description attribute in addition to a prompt attribute, the description is displayed in the UI along with the prompt. The purpose of the description attribute is to allow a complete description of what's being asked for along with a simple prompt.

  For example:

```
<text-var
        var-name="eProv.Company"
        prompt="Company name:" description="Please enter the name of your
company. This must be the same name as you entered during the initial
installation."
        browse="no">
      Novell
    </text-var>
```

  Note the differences between the `prompt` and the `description`.

  If a `variable-decl` contains an optional description attribute and an optional highlight attribute, the highlight attribute is handled as follows:

  - If the highlight is not two characters in length, it is ignored.

    - If the highlight is two characters in length, all occurrences of the first character are preceded with HTML tags to turn on highlighting and all occurrences of the second character are followed by HTML tags to turn off highlighting.

      For example:

      ```
      <text-var
                  var-name="foo"
                  prompt="Foo:"
                  description="Please enter some foo.  Format:  [foo looks
      like this]">
                Bar
              </text-var>
      ```

      When the description is displayed, [foo looks like this] is displayed and highlighted.

- If a `variable-decl` contains a `browse="yes"` attribute, it is assumed to supply a DN and is formatted in slash format by default when applied to the driver configuration file.

  This is assumed to be more generally useful for driver writers and can be overridden on a per reference basis by adding a `dn-format="dot"` attribute to `variable-ref` nodes that reference it.

- If a `variable-ref` is to `text-var` with a `prompt-type="ipa"` attribute, a `part="..."` attribute can be included in the `variable-ref`. Supported parts are `"ipa"` and `"port"`. If `part="ipa"` is specified, only the IP address portion of the variable's value is returned. If `part="port"` is specified, only the port portion of the variable's value is returned. Any other setting is ignored and the variable's entire value is returned.
- A `dn-format` attribute on a `variable-ref` that does not have `browse="yes"` specified in its `variable-decl` causes that variable to be treated as though it supplies a DN. The DN is rendered in the `dn-format` specified.
- The supported values for the `dn-format` attribute are `"dot"` and `"slash"`. Any other value is treated as `"slash"` and without an error being generated.
- The built-in defined variables are:
    - System.TreeName
    - System.DSetDN
    - System.DSetName
    - System.DriverDN
    - System.DriverName
    - System.ServerDN
    - System.ServerName
- Built-in variables can be overridden. If you include a `variable-decl` for a variable named with one of the built-in variable names, your definition overrides the built-in variable of the same name.

  This is implemented after all variable declarations have been processed (prompting, ...). Just before the code begins applying values, it walks the variables and defines all the built-ins that haven't otherwise been defined.

- The built-in variables that provide a DN can include a dn-format attribute in the variable-ref to control the format the DN is rendered in. By default, these are rendered in slash format.
- A `node-var` and a `text-var` cannot be named the same thing. They use the same namespace.

- If a `variable-ref` references a `node-var` and contains an `attr-name` attribute, the XSL string value of the `node-var` is stored in as the named attribute on the parent node of the `variable-ref`. The `node-var` used in this manner can have a `node-name` attribute of `"#text",` which removes the requirement of having an `attr-name` attribute on the `node-var`.

  A `node-var` with a node-name of `"#text"` can only be referenced in this manner. Any other reference causes an error when the driver configuration file is imported.
- At patch time after the user has responded to the prompts but before the XML is actually imported. patching is done in the following order:
  1. `The text-var` `variable-refs` are processed.
  2. `The node-var` `variable-refs` are processed.
  3. `The xsl-modify` commands are processed.
  4. The `ds-object` commands are processed.
     - Patching is performed in the `variable-decl` so that by the time the `node-var` commands are patched, all the `text-var` commands contained in them have been resolved.
     - The `node-var` commands cannot contain `node-var variable-ref`.

C.1.2 Import Driver Notes

- The order in which the selected driver configuration files are processed is not defined and no order can be assumed.
- For `variable-decl` commands:
  - Commands from selected drivers are carried forward from driver to driver.
  - The first one wins.
    - The first driver encountered that defines a variable foo has its variable foo used throughout all remaining driver configuration files. Care must be taken to coordinate this between drivers.
    - A variable foo that is used in multiple driver configuration files is only prompted for once, with the first driver configuration file encountered that declares it.
- Built-in variables are not propagated between drivers. This includes any variables you define to override a built-in variable. The built-in variables for each driver are handled separately.
- Other prompting is handled unchanged at the beginning of each driver configuration file's import sequence.
- Refer to Section C.2, “Flexible Prompting in a Driver Configuration File,” on page 334 for information about prompt layering supported by flexible prompting.

C.2 Flexible Prompting in a Driver Configuration File

`variable-decl` blocks can be marked to allow them to be prompted for separately, based on user input.

```
DTD changes:
-----------
* <!ENTITY % CompareMode "equals | not-equals">

  <!--**************************************************** -->
  <!--The variable-decl element contains definitions of variables   -->
  <!-- whose values can be prompted for and referred to throughout  -->
  <!-- the pre-configured driver file.                              -->
  <!-- ****************************************************** -->
  <!ELEMENT variable-decl(
    node-var*,
    text-var*)>
*  <!ATTLIST variable-decl
*    <!-- The following are used in the support of flexible -->
*    <!-- prompting.                                        -->
*    use-when-var  CDATA #IMPLIED
*    use-when-value  CDATA #IMPLIED
*    use-when-mode  (%CompareMode) "equals"
  >

* Added for flexible prompting.
```

Semantics

1. All `variable-decl` blocks with no `use-when-var` attribute are added to the prompt set.
2. All `variable-decl` blocks with a `use-when-var` attribute where the variable is defined and the variable value meets the condition are added to the prompt set.

   Variable analysis includes built-ins and variables carried forward from any previous import.
3. The user is prompted.
4. The prompt set is emptied and Steps 2 and 3 are repeated until there are no more prompts to process or all `variable-decl` blocks have been processed.
5. The import proceeds as before.

NOTE: The comparisons for `use-when-var` variables are case insensitive.

Example 1

```
     <variable-decl use-when-var="varCheck" use-when-value="Fu" use-when-
mode="equals">
    <text-var prompt="When Fu?" var-name="fuVar"/>
  </variable-decl>

  <variable-decl use-when-var="varCheck" use-when-value="Fu" use-when-
mode="not-equals">
    <text-var prompt="When not Fu?" var-name="fuVar"/>
  </variable-decl>

  <variable-decl>
    <text-var prompt="Which other <variable-decl>?" var-name="varCheck">
      <dropdown>
```

```
        <value>Fu</value>
        <value>Bar</value>
      </dropdown>
    </text-var>
  </variable-decl>
```

In this example, the user would be prompted with a drop-down list. The description of the drop-down list is “Which other <variable-decl>?” The options in the list are Fu and Bar.

If the user select Fu from the drop-down and clicks Next, he or she is prompted again with a box. The description of the box is “When Fu?”

If the user selects anything else from the drop-down list and clicks Next, he or she is prompted with another box. The description of the box is “When not Fu?”

Example 2

```
<variable-decl use-when-var="varCheck" use-when-value="Fu">
    <text-var prompt="When Fu?" var-name="fuBarVar"/>
  </variable-decl>

  <variable-decl use-when-var="varCheck" use-when-value="Bar">
    <text-var prompt="When when Bar?" var-name="fuBarVar"/>
  </variable-decl>

  <variable-decl>
    <text-var prompt="Which other <variable-decl>?" var-name="varCheck"/>
  </variable-decl>
```

In this example, the user is presented with a box. The description of the box is “Which other <variable-decl>?” If the user specifies “Fu” in the box and clicks Next, he or she is presented with another box. The description on the second box is “When Fu?”

If the user specifies “Bar” in the box and clicks Next, he or she is presented with a box. The description is “When Bar?” If he or she specifies anything else, there are no further prompts and the variable fuBarVar is not defined.

C.3 The Informal Identity Manager 3.5 Driver Configuration DTD

To view the informal Identity Manager 3.5 Driver Configuration DTD, go to PCDrivers.txt (../samples/PCDrivers.txt). The DTD cannot be used for validation. It is not a valid XML DTD. It is a mechanism to document the valid constructs in a driver configuration file.

Manual Task Service Driver: Replacement Data

D

Replacement data is used with XML documents used as templates to construct e-mail messages, Web pages, and XDS documents. The actual replacement is accomplished by processing the template document with an XSLT style sheet that performs the replacement as part of constructing the output document.

Replacement data is supplied to the Manual Task Service Driver through different mechanisms on the Subscriber and Publisher channels.

Subscriber Channel

- Replacement data is supplied as part of the <mail> element.
- Part of the supplied replacement data can be URL data. If URL data is supplied, it is processed and completed and replaced by automatic data items (see Appendix E, "Manual Task Service Driver: Automatic Replacement Data Items," on page 343).
- If the <mail> element specifies that an association value should be constructed (that is the <mail> element has a src-dn attribute) then an automatic data item named "association" is added to the replacement data.

Publisher Channel

- Replacement data is supplied in the HTTP URL data and HTTP POST data.
- Automatic URL replacement data items are added to the replacement data before it is used in template processing.

Replacement data is presented during template processing as an XML document. The replacement data document is passed to the style sheet processing the template as a parameter named replacement-data. If no template is used, the XML document is processed directly by the style sheet.

D.1 Data Security

Data items are passed from the Subscriber channel to the Publisher channel via a URL contained in the e-mail sent by the Subscriber channel. Changing certain data items in the URL represents a security threat. For example, if the responder-dn values in the URL supplied by the Subscriber channel in the URL are replaced by another user's DN in the URL submitted to the Publisher channel Web server, it would allow an unauthorized user to change data in eDirectory.

To ensure that the data in the submitted URL is the same as the data originally supplied by the Subscriber channel, protected data is provided. Protected data is data that cannot be changed for security reasons. This data varies by configuration but always includes the responder-dn data items, and data items corresponding to any eDirectory objects whose values are to be changed.

Data items are protected by encrypting the original values and placing the encrypted values into a URL query string. When the Publisher Web server receives the encrypted values, the Publisher decrypts the values and uses them to compare the unencrypted data items that are supplied by an HTTP GET or POST request.

If an instance of a data item appears in the encrypted data, then an unencrypted data item value must match one of the encrypted data item values. If the unencrypted data item value does not match one of the encrypted data item values, then the HTTP request is rejected by the Publisher channel Web server.

In addition, any HTTP POST request that does not contain protected data is rejected.

Example

In an HTTP POST request, the Publisher channel Web server uses the unencrypted POST data named responder-dn to check the password supplied by the POST data. This is done to authenticate the responding user against the user's eDirectory object.

Suppose the Subscriber channel <url-query> element content specifies two data items as follows:

```
<item name="responder-dn" protect="yes">\PERIN-TAO\novell\phb</item>
```

```
<item name="responder-dn" protect="yes">\PERIN-TAO\novell\carol</item>
```

The URL generated by the Subscriber channel will contain both responder-dn values in the protected data.

Suppose a malicious user obtains the URL that is generated and sent in an e-mail message. The malicious user uses the URL to obtain the HTML form that allows users to change data for an eDirectory object.

In the HTTP POST request that is submitted to the Web server, the malicious user uses his eDirectory DN (responder-dn=\PERIN-TAO\novell\wally) as the unencrypted responder-dn value. The malicious user also submits his own password in the POST data so that the authentication that the Web server performs will succeed.

However, when the Publisher channel Web server receives the HTTP POST data, it fails to find “\PERIN-TAO\novell\wally” in the encrypted protected data and rejects the POST request.

D.2 XML Elements

The elements that make up a replacement data document are described below. If no XML attributes are described for an element, then none are allowed.

D.2.1 <replacement-data>

The <replacement-data> element can appear in the following locations:

1. As a child of the <message> element under a Subscriber channel <mail> element.

   The Manual Task Service Driver processes the supplied <replacement-data> element into a standalone <replacement-data> element for use in template processing. The following processing occurs:

   a. If an association value is created for the enclosing <mail> element, an <item name="association"> element is added to the replacement data. The value of the created element is the association value that is returned to Identity Manager.

   b. If the <replacement-data> element has a <url-data> element child, then the <url-data> element is replaced by several <item> elements that contain constructed URL data. See <url-data> and <url-query>.

2. As the standalone top-level element of a replacement data document used when constructing a document using a style sheet on either the Subscriber or the Publisher channels.

D.2.2 <item>

The <item> element can be a child of the <replacement-data> element, the <url-data> element, or the <url-query> element. The content of the <item> element is the text used in the substitution of replacement tokens in templates. <item> elements are always named using the name attribute.

<item> attributes

name: The value of the name attribute specifies the name by which this data item is referenced by replacement tokens. For example, if the value of the name attribute is manager, then the replacement token $manager$ is replaced by the value contained by <item name="manager"> element. The name attribute is required.

protect: For <item> elements that are children of <url-query> elements, the protect attribute specifies whether the item is added to the protected data section of the URL query string (see <url-query>. If the protect attribute is present, it must have the value yes.

Predefined <item> names

Certain <item> elements have predefined meanings to either the Subscriber channel, the Publisher channel, or both channels.

template: The Publisher channel treats the value of the template item as the name of the template document to use in generating the response to an HTTP GET request.

When <item name="template"> appears as a child of the <url-query> element on the Subscriber channel, the value is placed into the URL query data to specify to the Publisher channel Web server the name of the template document to use when responding to the HTTP GET request.

responder-dn: The Publisher channel uses the value of the responder-dn item in HTTP POST data as the DN of the eDirectory object against which the password supplied in the HTTP POST data is validated.

The Web server rejects any HTTP POST request that does not contain a responder-dn value and a password value. In addition, if the HTTP POST data does not contain a protected-data item, then the request is rejected.

The Subscriber channel supplies one or more <item name="responder-dn" protect="yes"> elements under the <url-query> element. Because the responder-dn items are used for user authentication, the items must be protected.

password: Supplied to the Publisher channel Web server via HTTP POST data. The item content is the password, which is validated against the eDirectory object specified by the responder-dn item in the POST data. The password item is normally entered in the HTML form used to generate the HTTP POST request.

Example:

```
<INPUT TYPE= "password" NAME="password" SIZE="20" MAXLENGTH="40"/>
```

response-template: Supplied to the Web server via HTTP POST data. Used to generate the Web page used as the response to the POST. The response-template item is normally specified using a hidden INPUT element in the HTML form used to generate the HTTP POST request.

Example:

```
<INPUT TYPE="hidden" NAME="response-template" VALUE="post_form.xml"/>
```

response-stylesheet: Supplied to the Web server via HTTP POST data. Used to generate the Web page used as the response to the POST. The response-stylesheet item is normally specified using a hidden INPUT element in the HTML form used to generate the HTTP POST request.

Example:

```
<INPUT TYPE="hidden" NAME="response-stylesheet" VALUE="process_template.xsl"/>
```

auth-template: Supplied to the Web server via HTTP POST data. Used to generate the Web page that is used as the response to the POST if authentication of the user fails. The auth-template item is normally specified using a hidden INPUT element in the HTML form used to generate the HTTP POST request.

Example:

```
<INPUT TYPE="hidden" NAME="auth-template" VALUE="auth_response.xml"/>
```

auth-stylesheet: Supplied to the Web server via HTTP POST data. Used to generate the Web page that is used as the response to the POST if authentication of the user fails. The auth-template item is normally specified using a hidden INPUT element in the HTML form used to generate the HTTP POST request.

Example:

```
<INPUT TYPE="hidden" NAME="auth-stylesheet" VALUE="process_template.xsl"/>
```

protected-data: The protected-data item contains the encrypted data constructed by the Subscriber channel. On the Subscriber channel, the protected data item is an automatically supplied item.

On the Publisher channel, the protected-data item is obtained from the URL query string for an HTTP GET request and is obtained from the POST data for an HTTP POST request.

The protected data item is typically passed from the HTTP GET request into the Web page used to generate the HTTP POST via a replacement token in the template used to construct the response to the HTTP GET.

Example:

```
<INPUT TYPE="hidden" NAME="protected-data" VALUE="$protected-data$"/>
```

D.2.3 <url-data>

The <url-data> element is a child of the <replacement-data> element found under the <message> element on the Subscriber channel. It contains <item> elements used to construct the URL and related data items that are supplied to the template used in constructing the e-mail message. It also contains the <url-query> element.

For the purposes of the Manual Task Service driver, URLs consist of five parts:

1. A scheme such as http, https, or ftp.
2. A host such as www.novell.com or 192.168.0.1.
3. A port number. This is a colon followed by a decimal integer. For example, :80 or :8180.
4. A file or resource specifier. This is typically a filename and can include path information. For example, stylesheets/process_template.xsl.
5. A query string. This is a collection of name-value pairs, separated by & characters. For example, template=form_template.xml&protected-data=AabABJKEL=

Predefined <item> Names Under <url-data>

<item> elements under the <url-data> element are ignored unless they are one of the following. All of them are optional.

file: Specifies the file portion of the URL. If used with the Publisher channel Web server, the file item specifies the style sheet to use to construct the initial HTML page returned in response to the URL. If used with a server other than the Publisher channel Web server, the file item specifies the name of the resource that the URL will refer to.

If the file item does not appear, the URL file portion defaults to process_template.xsl.

scheme: Optional item found under the <url-data> element. If present, it specifies the scheme portion of the URL (such as http or ftp). The scheme item is typically used only if the URL points at a server other than the Publisher's Web server.

If the scheme item does not appear, the URL scheme defaults to either http or https, depending on the configuration of the Publisher channel Web server.

host: Optional item found under the <url-data> element. If present, specifies the host portion of the URL. The host item is typically used only if the URL were to point at a server other than the Publisher's Web server.

If the host item does not appear, the URL host defaults to the IP address of the server on which the Manual Task Service Driver is running (that is, the IP address of the Publisher channel Web server).

port: Optional item found under the <url-data> element. If present, specifies the port portion of the URL. The port item is typically used only if the URL points at a server other than the Publisher's Web server.

If the port item does not appear, the URL port defaults to the port on which the Publisher channel Web server is running.

D.2.4 <url-query>

The <url-query> element is a child of the <url-data> element. It contains <item> elements that are used to construct the query portion of the URL used in the e-mail message.

Each item that appears as a child of the <url-query> element is placed in the query string in the form name="value" where name is the value of the <item> element's name attribute and value is the string content of the <item> element.

Item elements that appear under <url-query> can have a protect attribute with the value "yes." If this is the case, then the item names and values are encrypted and placed within a generated name-value pair in the URL query string. The name of the generated value is protected-data. The value is the Base64 encoded and encrypted name-value pair or pairs for multivalued attributes.

Protecting data ensures that the data cannot be changed when the URL is submitted to the Publisher channel Web server. For example, the responder-dn data items need to be protected to ensure that only those users authorized to respond to the e-mail message are able to change eDirectory data.

If the URL generated is to be used with the Publisher channel Web server, the <url-query> element must contain at least one <item name="responder-dn" protect="yes"> element or the Web server rejects the eventual HTTP POST request.

Manual Task Service Driver: Automatic Replacement Data Items

E

The Manual Task Service Driver automatically supplies certain replacement data item elements. This section describes those data items.

E.1 Subscriber Channel Automatic Replacement Data

The following data items are added automatically to replacement-data documents during processing by the Subscriber channel:

association: An <item name=”association”> element is added to the replacement-data document if the <mail> element has an <association> element child, or if the Subscriber returns an <add-association> element. The content of the <item> element is the association value for the eDirectory object that is associated with the e-mail message being processed. The association value might not yet be written to the eDirectory object; therefore, the association value cannot be used in queries.

url: The content of the <item> element is the complete URL to be used in the e-mail message. On the Subscriber channel, the url item is created from the following items found under the <url-data> element: scheme, host, port, file, and the items underneath the <url-query> element. If scheme, host, or port are not found, then default values are used. The default values are determined from the configuration of the Publisher channel Web server.

url-base: The content of the <item> element is the portion of the generated URL not including the resource identifier (file) and not including the query string.

url-query: The content of the <item> element is a URL query string generated from <item> elements underneath the <url-query> element.

url-file: The content of the <item> element is the resource identifier for the URL.

protected-data: The content of the <item> element is an encrypted form of name-value pairs obtained from <item> elements under the <url-query> element. Only <item> elements whose protect attribute is set to “yes” are added to the protected data value. See Data Security in Appendix D, “Manual Task Service Driver: Replacement Data,” on page 337 for more information about protected data.

E.2 Publisher Channel Automatic Replacement Data

The following data items are automatically added to replacement-data documents during processing by the Publisher channel Web server:

post-status: An <item name="post-status"> element is created and added to the replacement-data document by the Publisher channel Web server during the processing of an HTTP POST request. An HTTP POST request to the Web server is a request to submit an XDS document to Identity Manager. Identity Manager returns a status document as the result of the XDS submission. The content of the <item name="post-status"> element is the value of the level attribute of the <status> element that is returned by Identity Manager as the result of the submission to Identity Manager.

The post-status item is typically used in the construction of the Web page that is returned as the result of the HTTP POST request.

post-status-message: An <item name="post-status-message"> element is created and added to the replacement-data document by the Publisher channel Web server during the processing of an HTTP POST request. An HTTP POST request to the Web server is a request to submit an XDS document to Identity Manager. Identity Manager returns a status document as the result of the XDS submission. The content of the <item name="post-status-message"> element is the content of the <status> element that is returned by Identity Manager as the result of the submission to Identity Manager. The post-status-message item is created only if the <status> element returned by Identity Manager has content.

The post-status-message item is typically used in the construction of the Web page that is returned as the result of the HTTP POST request.

url: An <item name="url"> element is created and added to the replacement-data document by the Publisher channel Web server during processing of HTTP GET and HTTP POST requests . The <item> element is added before using the replacement-data document to construct any documents. The URL scheme, host, and port are determined by the Web server configuration.

url-base: An <item name="url-base"> is created and added to the replacement data document by the Publisher channel Web server during processing of HTTP GET and HTTP POST request. The <item> element is added before using the replacement-data document to construct any documents. The content of the url-base <item> element on the Publisher channel is the same as the url <item> element.

Manual Task Service Driver: Template Action Elements Reference

F

Action elements are namespace-qualified elements in a template document that are used for simple logic control or are used to create HTML elements for HTML forms. The namespace used to qualify the elements is http://www.novell.com/dirxml/manualtask/form. In this document and in the sample templates supplied with the Manual Task Service driver the prefix used is form.

Any action element not specifically covered in this section is stripped from the output document by the template-processing style sheet (unless the style sheet is customized). This behavior allows, for example, the use of a form:text element to enclose the data for a plain text e-mail message, thereby making the template valid XML.

F.1 <form:input>

The <form:input> element is used to generate one or more HTML INPUT elements based on the presence of one or more replacement data items. The number of INPUT elements created corresponds with the number of replacement data items with the name specified by the <form:input> element's name attribute.

Attributes

Name: Specifies the name of the replacement data items that are used to create the INPUT elements. The attribute value is used as the value of the name attribute of the created INPUT elements.

type or TYPE: Specifies the value of the type attribute of the created INPUT elements.

value: If the value attribute's value is equal to “yes,” then a value attribute is added to the created INPUT elements whose value is the string value of the replacement data item. If the value attribute's value is other than “yes,” then the content of the created INPUT elements is set to the string value of the replacement data item.

Example

```
<form:input name=“responder-dn” TYPE=“hidden” value=“yes”/>
```

creates one or more INPUT elements similar to

```
<INPUT name=“responder-dn” TYPE=“hidden” value=“\PERIN-TAO\novell\phb”/>
```

F.2 <form:if-item-exists>

The <form:if-item-exists> element is used to conditionally insert data into the output document. The content of <form:if-item-exists> is processed only if the specified item appears in the replacement data.

Attributes

Name: Specifies the name of the replacement data item. If one or more examples of the replacement data item exist, then the contents of the <form:if-item-exists> element are processed.

Example

```
<form:if-item-exists name="post-status-message">
      <tr>
        <td>
          Status message was: $post-status-message$
        </td>
      </tr>
</form:if-item-exists>
```

This example inserts a row into an HTML table only if there is a replacement data item named post-status-message.

F.3 <form:if-multiple-items>

The form:if-multiple-items element is used to conditionally insert data into the output document. The content of form:if-multiple-items is processed only if the specified item appears more than once in the replacement data.

Attributes

name: Specifies the name of the replacement data item. If more than one example of the replacement data item exists, then the content of the form:if-multiple-items is processed.

Example

```
<form:if-multiple-items name="responder-dn">
    <form:menu name="responder-dn"/>
</form:if-multiple-items>
```

This example builds an HTML SELECT element (see <form:menu>) if there is more than one replacement data item with the name responder-dn.

F.4 <form:if-single-item>

The form:if-single-item element is used to conditionally insert data into the output document. The content of form:if- single -item is processed only if the specified item appears exactly once in the replacement data.

Attributes

name: Specifies the name of the replacement data item. If the named item appears exactly once in the replacement data, then the content of the form:if-single-item is processed.

Example

```
<form:if-single-item name="responder-dn">
  <input TYPE="hidden" name="responder-dn" value="$responder-dn$"/>
  $responder-dn$
</form:if-single-item>
```

This example inserts an HTML INPUT element and some replacement text into the output document if there is exactly one replacement data item named “responder-dn” in the replacement data.

F.5 <form:menu>

The form:menu element is used to generate an HTML SELECT element with one or more OPTION element children. The first OPTION element child is marked as selected.

Attributes

name: Specifies the name of the replacement data item. If the named item appears in the replacement data, then an HTML SELECT element is created in the output document. An HTML OPTION element is created as a child of the SELECT element for each instance of the replacement data item in the replacement data.

Example

```
<form:menu name="responder-dn"/>
```

This example results in HTML elements similar to the following:

```
<SELECT name="responder-dn">
    <OPTION selected>\PERIN-TAO\big-org\php</OPTION>
    <OPTION>\PERIN-TAO\big-org\carol</OPTION>
</SELECT>
```

Manual Task Service Driver: <mail> Element Reference

G

The <mail> element and its content are described in detail in this section. If no attributes are listed for an element, then that element has no attributes defined.

G.1 <mail>

The <mail> element and its content describe the data necessary to construct an SMTP message.

<mail> attributes

src-dn: Contains the DN value of the eDirectory object that is triggering the e-mail. Required if the object's data is to be modified via the Publisher channel's Web server in response to the e-mail.

G.2 <to>

The <to> element is a child of the <mail> element. One or more <to> elements contain the e-mail addresses of the primary recipients of the SMTP message. At least one <to> element is required. Each <to> element must contain only a single e-mail address.

G.3 <cc>

The <cc> element is a child of the <mail> element. Zero or more <cc> elements contain the e-mail addresses of the CC recipients of the SMTP message. No <cc> element is required. Each <cc> element must contain only a single e-mail address.

G.4 <bcc>

The <bcc> element is a child of the <mail> element. Zero or more <bcc> elements contain the e-mail addresses of BCC recipients of the SMTP message. No <bcc> element is required. Each <bcc> element must contain only a single e-mail address.

G.5 <from>

The <from> element is a child of the <mail> element. The <from> element contains the e-mail address of the sender of the e-mail . The <from> element is not required. If the <from> element is not present, then the default from address supplied as part of the Manual Task Service Driver parameters is used.

G.6 <reply-to>

The <reply-to> element is a child of the <mail> element. The <reply-to> element contains the e-mail address of the entity to which replies to the SMTP message will be addressed. The <reply-to> element is not required.

G.7 <subject>

The <subject> element is a child of the <mail> element. Its string content is used to set the SMTP subject field. The <subject> element is not required but is recommended, for obvious reasons.

G.8 <message>

The <message> element is a child of the <mail> element. Its content is used to construct a message body for the SMTP message. At least one <message> element is required. Multiple <message> elements can be supplied when constructing an SMTP message with alternative representations of the message body (such as plain text and HTML, or English and another language).

<message> attributes

mime-type: Optionally specifies the MIME type of the message body constructed by the <message> element (such as text/plain or text/html). If the mime-type attribute is not present, the driver attempts to automatically discover the MIME type.

E-mail clients can use the MIME type when an SMTP message has alternative representations in order to choose the best representation to display.

language: Optionally specifies the language of the message body constructed by the <message> element. The value should follow the SMTP specification. If the language attribute is not present, no default is supplied.

E-mail clients can use the language specification when an SMTP message has alternative representations in order to choose the best representation to display.

G.9 <stylesheet>

The <stylesheet> element is a child of the <message> element. The content of the <stylesheet> element is the name of an XSLT style sheet used to construct the message body. If the <stylesheet> element is not present, then process_template.xsl is used as the style sheet.

G.10 <template>

The <template> element is a child of the <message> element. The content of the <template> element is the name of an XML document used to construct the message body. If the <template> element is not present, then the replacement data document is processed by the message style sheet to construct the message body.

G.11 <filename>

The <filename> element is a child of the <attachment> element. The content of the <filename> element is a filename. The filename value is used to assign a filename to a constructed attachment.

G.12 <replacement-data>

The <replacement-data> element is a child of the <message> element. Its content is used either as a parameter to the style sheet processing the message template, or in the absence of a template, it is processed directly by the message style sheet. The contents of the <replacement-data> element are described in Appendix D, “Manual Task Service Driver: Replacement Data,” on page 337 and Appendix E, “Manual Task Service Driver: Automatic Replacement Data Items,” on page 343.

G.13 <resource>

The <resource> element is a child of the <message> element. Its content is treated as the name of a file to be incorporated into the SMTP message a resource for the message body. For example, a .css style sheet for an HTML message body could be supplied as a resource.

<resource> attributes

cid: Specifies the content ID used to refer to the resource in URLs in the message body. For example, if a .css style sheet is the resource, then the cid value might be css-1. In the HTML message body the following element can be used to refer to the .css style sheet:

```
<link href="cid:css-1" rel="style sheet" type="text/css">
```

G.14 <attachment>

The <attachment> element is a child of the <mail> element. It can have the same content as <message>, or it can have a filename as content. Zero or more <attachment> elements can appear as children of the <mail> element.

<attachment> attributes

mime-type: Optionally specifies the MIME type of the attachment. If the mime-type attribute is not present, the driver will attempt to automatically discover the MIME type.

language: Optionally specifies the language of the attachment. If the language attribute is not present, no default is supplied.

Manual Task Service Driver: Data Flow Scenario for New Employee

H

This section gives a step-by-step examination of the data flow in an example situation when hiring a new employee causes an e-mail message to be sent to the employee's manager. The e-mail message requests that the manager use a URL in the message to enter a room number value for the employee.

The configuration of the Manual Task Service Driver is as follows for the example scenario.

H.1 Subscriber Channel Configuration

Filter

Class: User

Attributes: Given Name, manager, Surname

Policies

Create policy: Requires Given Name, manager, and Surname attributes.

Command Transformation policy: Converts the <add> into the <mail> element.

H.2 Publisher Channel Configuration

Filter

Class: User

Attributes: roomNumber

Policies

None.

H.3 Description of Data Flow

In the following list, the most important data items that flow through the process are responder-dn and association. The responder-dn item is used to authenticate the user entering data through the Web server. The association item identifies the eDirectory object whose data is to be changed.

1. The company hires a new employee. The new employee's data is entered into the company's Human Resource (HR) system.
2. The Identity Manager driver for the HR system creates a new User object in eDirectory. User attributes include Given Name, Surname, and manager.
3. The following <add> event for the new User object is submitted to the Manual Task Service Driver Subscriber channel:

```
<nds dtdversion="1.1" ndsversion="8.6">
  <input>
    <add class-name="User" src-dn="\PERIN-TAO\novell\Provo\Joe" src-entry-
id="281002" timestamp="1023314433#2">
      <add-attr attr-name="Surname">
        <value type="string">the Intern</value>
      <add-attr>
      <add-attr attr-name="Given Name">
        <value type="string">Joe</value>
      <add-attr>
      <add-attr attr-name="manager">
        <value type="dn">\PERIN-TAO\novell\Provo\phb</value>
      <add-attr>
    </add>
  </input>
</nds>
```

a. The Subscriber Command Transformation policy uses the manager DN value to issue a query to eDirectory for the manager's e-mail address and the manager's assistant's DN.

b. If the manager has an assistant, the Subscriber Command Transformation issues a query to eDirectory for the assistant's e-mail address.

c. The Subscriber Command Transformation constructs a <mail> element and replaces the <add> command element with the <mail> element. In the example below, replacement data items are in bold.

```
<nds dtdversion="1.1" ndsversion="8.6">
  <input>
    <mail src-dn="\PERIN-TAO\novell\Provo\Joe">
      <to>phb@company.com</to>
      <cc>carol@company.com</cc>
      <bcc>HR@company.com</bcc>
      <reply-to>HR@company.com</reply-to>
      <subject>Room Assignment Needed for: Joe the Intern</subject>
      <message mime-type="text/html">
        <stylesheet>process_template.xsl</stylesheet>
        <template>html_msg_template.xml</template>
        <replacement-data>
          <item name="manager">JStanley</item>           <item
name="given-name">Joe</item>           <item name="surname">the
Intern</item>
          <url-data>
            <item name="file">process_template.xsl</item>
            <url-query>
            <item name="template">form_template.xml</item>
<item name="responder-dn" protect="yes">\PERIN-TAO\novell\Provo\phb</
item>               <item name="responder-dn" protect="yes">\PERIN-
TAO\novell\Provo\carol</item>               <item name="subject-
name">Joe the Intern</item>
            </url-query>
          </url-data>
        </replacement-data>
        <resource cid="css-1">novdocmain.css</resource>
      </message>
    </mail>
  </input>
</nds>
```

d. The Manual Task Service Driver Subscriber receives the <mail> element from Nsure™ Identity Manager.

e. The Subscriber generates an association value because the <mail> element has a src-dn attribute.

f. The Subscriber constructs a replacement data document from the data in the <mail> element for use in constructing the e-mail message. The URL has various data items in the query portion (that portion of the URL that follows the '?' character and is in bold). The Publisher channel Web server uses these data items when the URL is submitted to the Web server as an HTTP GET request.

```
<replacement-data>
  <item name="manager">JStanley</item>
  <item name="given-name">Joe</item>
  <item name="surname">the Intern</item>
  <item name="template">form_template.xml</item>
  <item name="responder-dn">\PERIN-TAO\novell\Provo\phb</item>
  <item name="responder-dn">\PERIN-TAO\novell\Provo\carol</item>
  <item name="subject-name">Joe the Intern</item>
  <item name="association">1671b2:ee4246a561:-7fff:192.168.0.1</item>
  <item name="url-base">https://192.168.0.1:8180</item>
  <item name="url-file">process_template.xsl</item>
  <item name="protected-data">
rO0ABXNyABlqYXZheC5jcnlwdG8uU2VhbGVkT2JqZWN0PjY9psO3VHACAARbAA
1lbmNvZGVkUGFyYW1zdAACW0JbABBlbmNyeXB0ZWRDb250ZW50cQB+AAFMAAlw
YXJhbXNBbGd0ABJMamF2YS9sYW5nL1N0cmluZztMAAdzZWFsQWxncQB+AAJ4cH
VyAAJbQqzzF/gGCFTgAgAAeHAAAAAPMA0ECEIBRohGPjxEAgEKdXEAfgAEAAAA
uMSFqzHXwtMx8DkRCzkK1O46sEz1u51o3MDvHn+3+fE6SphHr3Hgjli4Jp3rUk
H7y6dXvcu7iq21Vs+9o6iZVzljTIJX/jjRrVZlR5JOuRNhk8JHFZ8FhgsmiIAH
/Fs61k4WmyEcmYfWmfqfBVeThr3Avwcim6ranS5Mm2U5i9Z/DBR13pIAobMpWY
kMaz4+G9e6oovBsiPdp6jSPzbFxcgALI2AMBh4hf9jnx7zOU9Uvd9qXtaE2rR0
AANQQkV0ABBQQkVXaXRoTUQ1QW5kREVT</item>
  <item name="url-query">template=form_template.xml&amp;responder-
dn=%5CPERIN-TAO%5Cnovell%5Cprovo%5Cphb&amp;responder-dn=%5CPERIN-
TAO%5Cnovell%5Cprovo%5Ccarol&amp;subject-
name=Joe+the+Intern&amp;association=1671b2%3Aee4246a561%3A-
7fff%3A192.168.0.1&amp;protected-
data=rO0ABXNyABlqYXZheC5jcnlwdG8uU2VhbGVkT2JqZWN0PjY9psO3VHACAARbAA1l
bmNvZGVkUGFyYW1zdAACW0JbABBlbmNyeXB0ZWRDb250ZW50cQB%2BAAFMAAlwYXJhbXN
BbGd0ABJMamF2YS9sYW5nL1N0cmluZztMAAdzZWFsQWxncQB%2BAAJ4cHVyAAJbQqzzF%
2FgGCFTgAgAAeHAAAAAPMA0ECEIBRohGPjxEAgEKdXEAfgAEAAAAuMSFqzHXwtMx8DkRC
zkK1O46sEz1u51o3MDvHn%2B3%2BfE6SphHr3Hgjli4Jp3rUkH7y6dXvcu7iq21Vs%2B9
o6iZVzljTIJX%2FjjRrVZlR5JouRNhk8JHFZ8FhgsmiIAH%2FFs61k4WmyEcmYfWmfqfB
VeThr3Avwcim6ranS5Mm2U5i9Z%2FDBR13pIAobMpWYkMaz4%2BG9e6oovBsiPdp6jSPz
bFxcgALI2AMBh4hf9jnx7zOU9Uvd9qXtaE2rR0AANQQkV0ABBQQkVXaXRoTUQ1QW5kREV
T</item>
  <item name="url">
https://192.168.0.1:8180/
process_template.xsl?template=form_template.xml&amp;responder-
dn=%5CPERIN-TAO%5Cnovell%5CProvo%5Cphb&amp;responder-dn=%5CPERIN-
TAO%5Cnovell%5Cprovo%5Ccarol&amp;subject-
name=Joe+the+Intern&amp;association=1671b2%3Aee4246a561%3A-
7fff%3A192.168.0.1&amp;protected-
data=rO0ABXNyABlqYXZheC5jcnlwdG8uU2VhbGVkT2JqZWN0PjY9psO3VHACAARbAA1l
bmNvZGVkUGFyYW1zdAACW0JbABBlbmNyeXB0ZWRDb250ZW50cQB%2BAAFMAAlwYXJhbXN
BbGd0ABJMamF2YS9sYW5nL1N0cmluZztMAAdzZWFsQWxncQB%2BAAJ4cHVyAAJbQqzzF%
2FgGCFTgAgAAeHAAAAAPMA0ECEIBRohGPjxEAgEKdXEAfgAEAAAAuMSFqzHXwtMx8DkRC
zkK1O46sEz1u51o3MDvHn%2B3%2BfE6SphHr3Hgjli4Jp3rUkH7y6dXvcu7iq21Vs%2B9
```

```
o6iZVzljTIJX%2FjjRrVZlR5JouRNhk8JHFZ8FhgsmiIAH%2FFs61k4WmyEcmYfWmfqfB
VeThr3Avwcim6ranS5Mm2U5i9Z%2FDBR13pIAobMpWYkMaz4%2BG9e6oovBsiPdp6jSPz
bFxcgALI2AMBh4hf9jnx7zOU9Uvd9qXtaE2rR0AANQQkV0ABBQQkVXaXRoTUQ1QW5kREV
</item>
</replacement-data>
```

g. The Subscriber processes html_msg_template.xml with process_template.xsl. The replacement data document is passed as a parameter to the style sheet. The html_msg_template.xml document follows. Note the replacement tokens in bold. The replacement tokens are replaced by the value of the corresponding <item> elements in the replacement data document.

```
<html xmlns:form="http://www.novell.com/dirxml/manualtask/form">
  <head>
  </head>
  <body>
    <link href="cid:css-1" rel="style sheet" type="text/css"/>
    <p>
    Dear $manager$,
    </p>
    <p>
    This message is to inform you that your new employee <b>$given-
name$ $surname$</b> has been hired.
    </p>
    <p>
    Please assign a room number for this individual. Click <a
href="$url$">Here</a> to do this.
    </p>
    <p>
    Thank you,<br/>
    HR<br/>
    HR Department
    </p>
  </body>
</html>
```

The generated e-mail document follows. The replacement tokens have been replaced with the values of the corresponding <item> elements from the replacement data document.

```
<html>
  <head>
<META http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
    <link href="cid:css-1" rel="style sheet" type="text/css">
    <p>
    Dear J Stanley,
    </p>
    <p>
    This message is to inform you that your new employee <b>Joe the
Intern</b> has been hired.
    </p>
    <p>
    Please assign a room number for this individual. Click <a
href="https://192.168.0.1:8180/
process_template.xsl?template=form_template.xml&responder-
dn=%5CPERIN-TAO%5Cnovell%5CProvo%5Cphb&responder-dn=%5CPERIN-
TAO%5Cnovell%5CProvo%5Ccarol&subject-
name=Joe+the+Intern&association=45f0e3%3Aee45e07709%3A-
7fff%3A192.168.0.1&protected-
```

```
data=rO0ABXNyABlqYXZheC5jcnlwdG8uU2VhbGVkT2JqZWN0PjY9psO3VHACAARbAA1l
bmNvZGVkUGFyYW1zdAACW0JbABBlbmNyeXB0ZWRDb250ZW50cQB%2BAAFMAAlwYXJhbXN
BbGd0ABJMamF2YS9sYW5nL1N0cmluZztMAAdzZWFsQWxncQB%2BAAJ4cHVyAAJbQqzzF%
2FgGCFTgAgAAeHAAAAAPMA0ECIr9Z1iG%2BO3BAgEKdXEAfgAEAAAAuMU%2FSoFRkebvh
2d5SqalF91ttjRY5lyyW5%2B%2FFIfOuDdYikYiDbOJb6607S0dPHjQzeVgu6ptIvGqaE
QOEjBjDkY%2Bi4VoVjUSXS3a8fiXB8moMdPtLJ%2FGyE8QiwbT4xbkQy48i02k99F2vGm
lenRpSP6dD31kZl3dpJ0mGgq2yL%2FeFaynKyqnjkHLMexcqD8WlVooaRl1k2RPk5vDYv
C8o2bn22OKKbOnSRM5YlPS0iWzxo0JVcnVVyt0AANQQkV0ABBQQkVXaXRoTUQ1QW5kREV
T">Here</a> to do this.
    </p>
    <p>
    Thank you,<br>
    HR<br>
    HR Department
    </p>
  </body>
</html>
```

h. The SMTP e-mail message is sent to the manager and to the manager's assistant.

i. The Subscriber returns an XML document containing a <status> element and an <add-association> element to Identity Manager.

4. The manager opens the e-mail message and clicks the “Click here” link.

5. The manager's Web browser submits the URL to the Publisher channel Web server as an HTTP GET request.

   a. The Web server constructs the following replacement data document. Most of the data items come from the query portion of the URL. The exceptions are the automatically generated items url and url-base.

```
<replacement-data>
  <item name="association">45f0e3:ee45e07709:-7fff:192.168.0.1</item>
  <item name="protected-
data">rO0ABXNyABlqYXZheC5jcnlwdG8uU2VhbGVkT2JqZWN0PjY9psO3VHACAARbAA1
lbmNvZGVkUGFyYW1zdAACW0JbABBlbmNyeXB0ZWRDb250ZW50cQB+AAFMAAlwYXJhbXNB
bGd0ABJMamF2YS9sYW5nL1N0cmluZztMAAdzZWFsQWxncQB+AAJ4cHVyAAJbQqzzF/
gGCFTgAgAAeHAAAAAPMA0ECIr9Z1iG+O3BAgEKdXEAfgAEAAAAuMU/
SoFRkebvh2d5SqalF91ttjRY5lyyW5+/
FifOuDdYikYiDbOJb6607S0dPHjQzeVgu6ptIvGqaEQOEjBjDkY+i4VoVjUSXS3a8fiXB
8moMdPtLJ/GyE8QiwbT4xbkQy48i02k99F2vGmlenRpSP6dD31kZl3dpJ0mGgq2yL/
eFaynKyqnjkHLMexcqD8WlVooaRl1k2RPk5vDYvC8o2bn22OKKbOnSRM5YlPS0iWzxo0J
VcnVVyt0AANQQkV0ABBQQkVXaXRoTUQ1QW5kREVT</item>
  <item name="template">form_template.xml</item>
  <item name="responder-dn">\PERIN-TAO\novell\Provo\phb</item>
  <item name="responder-dn">\PERIN-TAO\novell\Provo\carol</item>
  <item name="subject-name">Joe the Intern</item>
  <item name="url-base">https://192.168.0.1:8180</item>
  <item name="url">https://192.168.0.1:8180</item>
</replacement-data>
```

   The Web server processes the form_templates.xml document with the process_template.xsl style sheet. Replacement tokens and action elements are in bold. Note that various data items are placed in hidden INPUT elements so that the data items are passed to the Web server as part of the HTML POST data.

   In addition, there is a $query:roomNumber$ replacement token, which retrieves the current value of the employee's roomNumber attribute (if any).

```
<html  xmlns:form="http://www.novell.com/dirxml/manualtask/form">
  <head>
    <title>Enter room number for $subject-name$</title>
  </head>
  <body>
    <link href="novdocmain.css" rel="style sheet" type="text/css"/>
    <br/><br/><br/><br/>
    <form class="myform" METHOD="POST" ACTION="$url-base$/
process_template.xsl">
        <table cellpadding="5" cellspacing="10" border="1"
align="center">
          <tr><td>
          <input TYPE="hidden" name="template" value="post_form.xml"/>
          <input TYPE="hidden" name="subject-name" value="$subject-
name$"/>
          <input TYPE="hidden" name="association"
value="$association$"/>
          <input TYPE="hidden" name="response-style sheet"
value="process_template.xsl"/>
          <input TYPE="hidden" name="response-template"
value="post_response.xml"/>
          <input TYPE="hidden" name="auth-style sheet"
value="process_template.xsl"/>
          <input TYPE="hidden" name="auth-template"
value="auth_response.xml"/>
         <input TYPE="hidden" name="protected-data" value="$protected-
data$"/>
          <form:if-single-item name="responder-dn">
            You are:<br/>
           <input TYPE="hidden" name="responder-dn" value="$responder-
dn$"/>
            $responder-dn$
          </form:if-single-item>          <form:if-multiple-items
name="responder-dn">
            Indicate your identity:<br/>
            <form:menu name="responder-dn"/>          </form:if-
multiple-items>
        </td></tr>
        <tr><td>
          Enter your password: <br/><input name="password"
TYPE="password" SIZE="20" MAXLENGTH="40"/>
        </td></tr>
        <tr><td>
          Enter room number for $subject-name$:<br/>
          <input TYPE="text" NAME="room-number" SIZE="20"
MAXLENGTH="20" value="$query:roomNumber$"/>
        </td></tr>
        <tr><td>
          <input TYPE="submit" value="Submit"/> <input TYPE="reset"
value="Clear"/>
        </td></tr>
      </table>
    </form>
  </body>
</html>
```

The following HTML page is the result:

```
<html>
  <head>
<META http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Enter room number for Joe the Intern</title>
  </head>
  <body>
    <link href="novdocmain.css" rel="style sheet" type="text/css">
    <br><br><br><br>
<form class="myform" METHOD="POST" ACTION="https://192.168.0.1:8180/
process_template.xsl">
<table cellpadding="5" cellspacing="10" border="1" align="center">
<tr>
<td>
    <input TYPE="hidden" name="template" value="post_form.xml">
    <input TYPE="hidden" name="subject-name" value="Joe the Intern">
    <input TYPE="hidden" name="association" value="45f0e3:ee45e07709:-
7fff:192.168.0.1">
    <input TYPE="hidden" name="response-style sheet"
value="process_template.xsl">
    <input TYPE="hidden" name="response-template"
value="post_response.xml">
    <input TYPE="hidden" name="auth-style sheet"
value="process_template.xsl">
    <input TYPE="hidden" name="auth-template"
value="auth_response.xml">
    <input TYPE="hidden" name="protected-data"
value="rO0ABXNyABlqYXZheC5jcnlwdG8uU2VhbGVkT2JqZWN0PjY9psO3VHACAARbAA
1lbmNvZGVkUGFyYW1zdAACW0JbABBlbmNyeXB0ZWRDb250ZW50cQB+AAFMAAlwYXJhbXN
BbGd0ABJMamF2YS9sYW5nL1N0cmluZztMAAdzZWFsQWxncQB+AAJ4cHVyAAJbQqzzF/
gGCFTgAgAAeHAAAAAPMA0ECIr9Z1iG+O3BAgEKdXEAfgAEAAAAuMU/
SoFRkebvh2d5SqalF91ttjRY5lyyW5+/
FIfOuDdYikYiDbOJb6607S0dPHjQzeVgu6ptIvGqaEQOEjBjDkY+i4VoVjUSXS3a8fiXB
8moMdPtLJ/GyE8QiwbT4xbkQy48i02k99F2vGmlenRpSP6dD31kZl3dpJ0mGgq2yL/
eFaynKyqnjkHLMexcqD8WlVooaRl1k2RPk5vDYvC8o2bn22OKKbOnSRM5YlPS0iWzxo0J
VcnVVyt0AANQQkV0ABBQQkVXaXRoTUQ1QW5kREVT">
    Indicate your identity:<br>
    <SELECT name="responder-dn">
        <OPTION selected>\PERIN-TAO\novell\Provo\phb</OPTION>
        <OPTION>\PERIN-TAO\novell\Provo\carol</OPTION>
    </SELECT>
</td>
</tr>
<tr>
<td>
    Enter your password: <br>

    <input name="password" TYPE="password" SIZE="20" MAXLENGTH="40">
</td>
</tr>
<tr>
<td>
    Enter room number for Joe the Intern:<br>
    <input TYPE="text" NAME="room-number" SIZE="20" MAXLENGTH="20"
value="">
</td>
</tr>
<tr>
<td>
    <input TYPE="submit" value="Submit"> <input TYPE="reset"
```

```
value="Clear">
</td>
</tr>
</table>
</form>
</body>
</html>
```

b. The manager selects his or her eDirectory DN from the Web page menu, enters the password, enters the room number for the new employee, and clicks Submit.

c. The Web browser submits an HTTP POST request to the Web server.

d. The Web server constructs the following replacement data document from the POST data. Note the data that was in the various hidden <INPUT> elements. The data entered by the manager in the form is in bold.

```
<replacement-data>
  <item name="room-number">cubicle 1234</item>
  <item name="template">post_form.xml</item>
  <item name="response-template">post_response.xml</item>
  <item name="auth-template">auth_response.xml</item>
  <item name="association">45f0e3:ee45e07709:-7fff:192.168.0.1</item>
  <item name="password" is-sensitive="true"><!-content suppressed ?</
item>
  <item name="protected-
data">rO0ABXNyABlqYXZheC5jcnlwdG8uU2VhbGVkT2JqZWN0PjY9psO3VHACAARbAA1
lbmNvZGVkUGFyYW1zdAACW0JbABBlbmNyeXB0ZWRDb250ZW50cQB+AAFMAAlwYXJhbXNB
bGd0ABJMamF2YS9sYW5nL1N0cmluZztMAAdzZWFsQWxncQB+AAJ4cHVyAAJbQqzzF/
gGCFTgAgAAeHAAAAAPMA0ECIr9Z1iG+O3BAgEKdXEAfgAEAAAAuMU/
SoFRkebvh2d5SqalF91ttjRY5lyyW5+/
FifOuDdYikYiDbOJb6607S0dPHjQzeVgu6ptIvGqaEQOEjBjDkY+i4VoVjUSXS3a8fiXB
8moMdPtLJ/GyE8QiwbT4xbkQy48i02k99F2vGmlenRpSP6dD31kZl3dpJ0mGgq2yL/
eFaynKyqnjkHLMexcqD8WlVooaRl1k2RPk5vDYvC8o2bn22OKKbOnSRM5YlPS0iWzxo0J
VcnVVyt0AANQQkV0ABBQQkVXaXRoTUQ1QW5kREVT</item>
  <item name="responder-dn">\PERIN-TAO\novell\Provo\phb</item>
  <item name="auth-style sheet">process_template.xsl</item>
  <item name="response-style sheet">process_template.xsl</item>
  <item name="subject-name">Joe the Intern</item>
  <item name="url-base">https://192.168.0.1:8180</item>
  <item name="url">https://192.168.0.1:8180</item>
</replacement-data>
```

e. The Web server verifies that the value of item responder-dn matches a responder-dn value contained in the protected data. If the value does not match, the Web server aborts the request. If the value does match, processing continues.

f. The Web server submits a <check-object-password> XDS request to Identity Manager on the Publisher channel to authenticate the user submitting the HTTP POST request.

```
<nds dtdversion="1.0" ndsversion="8.6">
  <source>
    <product build="20020606_0824" instance="Manual Task Service
Driver" version="1.1a">DirXML Manual Task Service Driver</product>
    <contact>Novell, Inc.</contact>
  </source>
  <input>
    <check-object-password dest-dn="\PERIN-TAO\novell\Provo\phb"
```

```
event-id="chkpwd">
      <password><!-- content suppressed --></password>
    </check-object-password>
  </input>
</nds>
```

g. Identity Manager returns <status level=”success”>. If Identity Manager returns other than success, then the templates specified by the data item auth_template and the style sheet specified by the data item auth_stylesheet are used to construct a Web page that is returned as the result of the POST.

h. The Web server processes the post_form.xml template with the process_template.xsl style sheet to generate an XDS document. Replacement tokens are in bold.

```
<nds>
  <input>
    <modify class-name="User" src-dn="not-applicable" event-
id=”wfmod”>
      <association>$association$</association>
      <modify-attr attr-name="roomNumber">
        <remove-all-values/>
        <add-value>
          <value>$room-number$</value>
        </add-value>
      </modify-attr>
    </modify>
  </input>
</nds>
```

i. The Publisher submits the created XDS document to Identity Manager.

```
<nds>
  <input>
    <modify class-name="User" src-dn="not-applicable" event-
id=”wfmod”>
      <association>45f0e3:ee45e07709:-7fff:192.168.0.1</association>
      <modify-attr attr-name="roomNumber">
        <remove-all-values/>
        <add-value>
          <value>cubicle 1234</value>
        </add-value>
      </modify-attr>
    </modify>
  </input>
</nds>
```

j. Identity Manager returns a result document

```
<nds dtdversion="1.1" ndsversion="8.6">
  <source>
    <product version="2.0">Identity Manager</product>
    <contact>Novell, Inc.</contact>
  </source>
  <output>
    <status event-id="wfmod" level="success"></status>
  </output>
</nds>
```

k. The Web server adds the replacement data item post-status (and possibly the replacement data item post-status-message) to the replacement data document. The added data item is in bold:

```
<replacement-data>
  <item name="room-number">cubicle 1234</item>
  <item name="template">post_form.xml</item>
  <item name="response-template">post_response.xml</item>
  <item name="auth-template">auth_response.xml</item>
  <item name="association">45f0e3:ee45e07709:-7fff:192.168.0.1</item>
  <item name="password" is-sensitive=”true”><!-content suppressed ?</
item>
  <item name="protected-
data">rO0ABXNyABlqYXZheC5jcnlwdG8uU2VhbGVkT2JqZWN0PjY9psO3VHACAARbAA1
lbmNvZGVkUGFyYW1zdAACW0JbABBlbmNyeXB0ZWRDb250ZW50cQB+AAFMAAlwYXJhbXNB
bGd0ABJMamF2YS9sYW5nL1N0cmluZztMAAdzZWFsQWxncQB+AAJ4cHVyAAJbQqzzF/
gGCFTgAgAAeHAAAAAPMA0ECIr9Z1iG+O3BAgEKdXEAfgAEAAAAuMU/
SoFRkebvh2d5SqalF91ttjRY5lyyW5+/
FifOuDdYikYiDbOJb6607S0dPHjQzeVgu6ptIvGqaEQOEjBjDkY+i4VoVjUSXS3a8fiXB
8moMdPtLJ/GyE8QiwbT4xbkQy48i02k99F2vGmlenRpSP6dD31kZl3dpJ0mGgq2yL/
eFaynKyqnjkHLMexcqD8WlVooaRl1k2RPk5vDYvC8o2bn22OKKbOnSRM5YlPS0iWzxo0J
VcnVVyt0AANQQkV0ABBQQkVXaXRoTUQ1QW5kREVT</item>
  <item name="responder-dn">\PERIN-TAO\novell\Provo\phb</item>
  <item name="auth-style sheet">process_template.xsl</item>
  <item name="response-style sheet">process_template.xsl</item>
  <item name="subject-name">Joe the Intern</item>
  <item name="url-base">https://192.168.0.1:8180</item>
  <item name="url">https://192.168.0.1:8180</item>
  <status event-id="" level="success"></status>
  <item name="post-status">success</item>
</replacement-data>
```

l. The Web server processes the post_response.xml template with the process_template.xsl style sheet. Replacement tokens and action elements are in bold.

```
<htm xmlns:form="http://www.novell.com/dirxml/manualtask/form">
  <head>
    <title>Result of post for $subject-name$</title>
  </head>
  <body>
    <link href="novdocmain.css" rel="style sheet" type="text/css"/>
    <br/><br/><br/><br/>
    <table class="formtable" cellpadding="5" cellspacing="20"
border="1" align="center">
      <tr>
        <td>
          DirXML reported status = $post-status$
        </td>
      </tr>
<form:if-item-exists name="post-status-message">
      <tr>
        <td>
          Status message was: $post-status-message$
        </td>
      </tr>
</form:if-item-exists>
    </table>
  </body>
</html>
```

m. The resulting Web page is returned as the result of the HTTP POST. The second row of the table is not present because the post-status-message referred to by the <form:if-item-exists> element is not present in the replacement data document.

```
<html>
  <head>
<META http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Result of post for Joe the Intern</title>
  </head>
  <body>
    <link href="novdocmain.css" rel="style sheet" type="text/css">
    <br><br><br><br>
    <table class="formtable" cellpadding="5" cellspacing="20" 
border="1" align="center">
      <tr>
        <td>
          DirXML reported status = success
        </td>
      </tr>
    </table>
  </body>
</html>
```

Manual Task Service Driver: Custom Element Handlers for the Subscriber Channel

I

The driver provides an extension mechanism for sending user notifications using methods other than the Simplified Mail Transport Protocol (SMTP). For example, a customer might have a need to send notifications using the Messaging Application Programming Interface (MAPI) rather than using SMTP.

To use a mechanism other than SMTP for sending notifications, you must write a Java class to handle a custom XML element that is submitted on the driver's Subscriber channel.

The Java custom element handler must implement the com.novell.nds.dirxml.driver.manualtask.CommandHandler Java interface. The name of the custom element class is specified in the Additional Handlers item found in the Subscriber configuration parameters.

When the Subscriber channel encounters a command element, it looks in its table of handlers. When it finds a handler that reports that it handles the command element, the command element is passed to the handler. The handler then performs any processing required.

There are two built-in command element handlers in the driver: a handler for <mail> elements and a handler for <add> elements.

The custom command element definition is up to the author of the custom handler. A reasonable place to start in designing the custom command element is the design of the <mail> element.

The custom elements are created by policies on the Subscriber channel in the same fashion that the <mail> element is created.

The documentation for com.novell.nds.dirxml.driver.manualtask.CommandHandler and the documentation for many utility and support classes are found in the javadocs that ship with the driver. The javadocs are found in the file named manual_task_docs.zip in the distribution image.

I.1 Constructing URLs for Use with the Publisher Channel Web Server

To securely use the driver's Publisher channel web server, it is necessary to use utility classes to construct the URL that is to be included with a notification message. The com.novell.nds.dirxml.driver.manualtask.URLData is designed for this task.

The sample code found in SampleCommandHandler.java illustrates this process.

I.2 Constructing Message Documents using Stylesheets and Template Documents

It is convenient to use the same method to construct documents that the SMTP handler uses, which is a combination of style sheets, template documents, and replacement data. To do this, you must obtain the stylesheets and template documents, and invoke the style sheet processor programmatically.

The sample code found in SampleCommandHandler.java illustrates this process.

I.3 SampleCommandHandler.java

Source code for a sample custom command handler is included with the driver distribution. The source code is found in the manual_task_docs.zip file in the distribution image.

The handler is implemented in the com.novell.nds.dirxml.driver.manualtask.samples.SampleCommandHandler class.

The sample handler simply generates a document using style sheets and templates and writes the resulting document to a file.

I.3.1 Compiling the SampleCommandHandler Class

You can use any Java 2 compiler to compile the SampleCommandHandler class. You must place nxsl.jar, dirxml.jar, collections.jar, and ManualTaskServiceBase.jar in the Java compiler classpath.

I.3.2 Trying the SampleCommandHandler Class

Start by importing the Room Number sample configuration for the driver.

Compile the SampleCommandHandler class and place the resulting class file in a .jar file. Place the .jar file in the DirXML .jar file directory appropriate to the platform on which you are running the driver.

Add the following XML element under the <subscriber-options> element found in the Driver Parameters XML section of the driver properties:

```
<output-path display-name="Sample Output Path"></output-path>
```

Edit the Driver Parameters. In the item labeled Sample Output Path, place a path to a directory in which the SampleCommandHandler will write its created documents. In the item labeled Additional Handlers, add the string com.novell.nds.dirxml.driver.manualtask.samples.SampleCommandHandler.

Replace the Subscriber channel command transformation policy with CommandXform.xsl which is found in the same directory as the SampleCommandHandler.java file.

Create a User object and add a manager reference to the User object. If the manager has an e-mail address value, then a <sample> command element is sent to the Subscriber and the SampleCommandHandler writes a file in the location you specified above.

Manual Task Service Driver: Custom Servlets for the Publisher Channel

J

The driver provides an extension mechanism through which additional functionality can be added to the Publisher channel Web server. Custom servlets can be loaded by the Publisher by specifying the name of the servlet classes in the Driver configuration item labeled Additional Servlets.

J.1 Using the Publisher Channel

If a custom servlet needs to submit data to Identity Manager, the servlet must use the driver's Publisher channel. The com.novell.nds.dirxml.driver.manualtask.ServletRegistrar and com.novell.nds.dirxml.driver.manualtask.PublisherData classes are supplied to facilitate this. The sample code found in SampleServlet.java illustrates this process.

J.2 Authentication

A custom servlet must authenticate users that are submitting information. The sample code found in SampleServlet.java illustrates this process. However, the type of authentication performed using the <check-object-password> element does not check eDirectory™ rights. Changes submitted on the Publisher channel are allowed if the Driver object has rights to perform the changes, regardless of whether the user submitting the changes has rights or not.

If you are using a URL generated by a command handler on the Subscriber channel, you must use the com.novell.nds.dirxml.driver.manualtask.URLData class to validate the URL to ensure that the responder-dn data item has not been tampered with. See the javadocs for information on accomplishing this.

J.3 SampleServlet.java

Source code for a sample servlet is included with the driver distribution. The source code is found in the file manualtask_driver_docs.zip in the distribution image.

The servlet is implemented in the com.novell.nds.dirxml.driver.manualtask.samples.SampleServlet class.

The sample servlet accepts an HTTP GET request for any resource ending in .sample. The query string of the HTTP URL must contain a dest-dn item, an attr-name item, and a value item.

The servlet authenticates the user, then submits a modify request to Identity Manager via the driver's Publisher channel.

J.3.1 Compiling the SampleServlet Class

You can use any Java 2 compiler to compile the SampleServlet class. You must place nxsl.jar, dirxml.jar, collections.jar, and ManualTaskServiceBase.jar in the Java compiler classpath.

J.3.2 Trying the SampleServlet Class

Start by importing the Room Number sample configuration for the driver.

Compile the SampleServlet class and place the resulting class file in a .jar file. Place the .jar file in the DirXML .jar file directory appropriate to the platform on which you are running the driver.

Edit the Driver Parameters. In the item labeled Additional Servlets, add the string com.novell.nds.dirxml.driver.manualtask.samples.SampleServlet.

Add Telephone Number to the Publisher channel filter.

Submit the following URL in a browser (assuming the browser is running on the same machine as the driver):

https:localhost:8180/1.sample?dest-dn=username.container&attr-name=Telephone%20Number&value=555-1212

Replace *username.container* with the DN of a user in your tree.

Documentation Updates

K

The documentation was updated on the following dates:

- Section K.1, “June 3, 2008,” on page 369
- Section K.2, “March 20, 2008,” on page 369
- Section K.3, “March 17, 2008,” on page 370
- Section K.4, “March 7, 2008,” on page 370
- Section K.5, “October 9, 2007,” on page 370

K.1 June 3, 2008

Updates were made to the following sections. The changes are explained below.

K.1.1 Installing the Remote Loader

Location	Change
Section 3.4.2, “Configuring the Remote Loader by Creating a Configuration File,” on page 69	Added the new options of fromaddress and handshaketimeout.
Appendix B, “Options for Configuring a Remote Loader,” on page 321	Added the new options of fromaddress and handshaketimeout.

K.2 March 20, 2008

Section Section 3.4.2, “Configuring the Remote Loader by Creating a Configuration File,” on page 69 has been updated. The changes are explained below:

K.2.1 Configuring the Remote Loader

Location	Change
Section 3.4.2, “Configuring the Remote Loader by Creating a Configuration File,” on page 69	Created separate tables for the `-connection` option and its parameters. Added a new column called ‘Platforms’ to the parameters table

K.3 March 17, 2008

A new section was added to Section 1.1, "Identity Manager," on page 13. The changes are explained below:

K.3.1 Identity Manager

Location	Change
Section 1.1.11, "Identity Manager and Clustering," on page 21	Added this new section to explain how Identity Manager works in a clustered environment.

K.4 March 7, 2008

Updates were made to the following sections. The changes are explained below.

K.4.1 Java Class Names

Location	Change
Table B-2 on page 328	Changed the JDBC driver class from `com.novell.nds.dirxml.jdbc.JDBCDriverShim`to `com.novell.nds.dirxml.driver.jdbc.JDBCDriverShim`

K.5 October 9, 2007

Updates were made to the following sections. The changes are explained below.

K.5.1 Installing the Remote Loader

Location	Change
Section 3.3.3, "Installing a Remote Loader on a Windows Server," on page 58	Added the warning.

K.5.2 Configuring the Remote Loader

The following updates were made in this section:

Location	Change
Step 5 on page 67	Added security information for setting passwords.
Step 6 on page 67	Added security information for setting passwords.

Location	Change
Step 4 on page 85	Added security information for setting passwords.
Step 5 on page 85	Added security information for setting passwords.
Step 5 on page 86	Added security information for setting passwords.

K.5.3 Security: Best Practices

Location	Change
Section 9.2, “Securing Access,” on page 259	Added the File System Access item.