

Novell Identity Manager

3.0.1

www.novell.com

ADMINISTRATION GUIDE

August 14, 2007



Novell[®]

Legal Notices

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

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

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

Copyright © 2005 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

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

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

Novell Trademarks

eDirectory is a trademark of Novell, Inc.

exteNd is a trademark of Novell, Inc.

exteNd Director is a trademark of Novell, Inc.

GroupWise is a registered trademark of Novell, Inc., in the United States and other countries.

NDS is a registered trademark of Novell, Inc., in the United States and other countries.

NetWare is a registered trademark of Novell, Inc., in the United States and other countries.

NMAS is a trademark of Novell, Inc.

Novell is a registered trademark of Novell, Inc., in the United States and other countries.

Novell Certificate Server is a trademark of Novell, Inc.

Novell Client is a trademark of Novell, Inc.

SUSE is a registered trademark of Novell, Inc., in the United States and other countries.

Third-Party Materials

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

Contents

About This Guide	11
1 Overview Of Identity Manager 3.0.1 Architecture	13
1.1 Terminology Changes from Earlier Versions	13
1.2 Identity Manager	14
1.2.1 Metadirectory Engine	15
1.2.2 Driver Configuration Files	15
1.2.3 Identity Manager Event Cache	16
1.2.4 Driver Shim	16
1.2.5 Driver Set	16
1.2.6 Driver Object	18
1.2.7 Publisher and Subscriber Channels	19
1.2.8 Events and Commands	20
1.2.9 Policies and Filters	20
1.2.10 Associations	20
1.3 User Application	21
1.4 Designer	21
2 Managing Identity Manager Drivers	23
2.1 Creating and Configuring a Driver	23
2.1.1 Creating a Driver Object	23
2.1.2 Creating Multiple Drivers	24
2.2 Managing DirXML 1.1a Drivers in an Identity Manager Environment	24
2.3 Upgrading a Driver Configuration from DirXML 1.1a to Identity Manager Format	25
2.4 Starting, Stopping, or Restarting a Driver	25
2.5 Driver Parameters	25
2.6 Using Global Configuration Values	26
2.7 Using the DirXML Command Line Utility	26
2.8 Viewing Versioning Information	26
2.8.1 View a Hierarchical Display of Versioning Information	27
2.8.2 View the Versioning Information as a Text File	28
2.8.3 Save Versioning Information	30
2.9 Using Named Passwords	31
2.9.1 Configuring Named Passwords Using Designer	32
2.9.2 Configuring Named Passwords Using iManager	32
2.9.3 Using Named Passwords in Driver Policies	34
2.9.4 Configuring Named Passwords Using the DirXML Command Line Utility	34
2.10 Reassociating a Driver Object with a Server	37
2.11 Adding Driver Heartbeat	38
2.12 Viewing Identity Manager Processes	39
2.12.1 Adding Trace Levels in Designer	39
2.12.2 Adding Trace Levels in iManager	41
2.12.3 Capturing Identity Manager Processes to a File	42
3 Setting Up a Connected System	45
3.1 Overview	45
3.2 Providing for Secure Data Transfers	47

3.2.1	Creating a Server Certificate	48
3.2.2	Exporting a Self-Signed Certificate	48
3.3	Setting Up Remote Loaders	49
3.3.1	Installing Remote Loaders	50
3.3.2	Configuring Remote Loader	52
3.4	Configuring the Identity Manager Drivers for Use with Remote Loaders	67
3.4.1	Importing and Configuring a New Driver	67
3.4.2	Configuring an Existing Driver	69
3.4.3	Creating a Keystore	70

4 Creating Policies 73

5 Password Synchronization across Connected Systems 75

5.1	Overview	75
5.1.1	Overview of Passwords	75
5.1.2	What is Bidirectional Password Synchronization?	76
5.1.3	Comparison of Password Synchronization 1.0 and Identity Manager Password Synchronization	77
5.1.4	Features of Identity Manager Password Synchronization	78
5.1.5	Overview Illustrations of Password Synchronization Flow	81
5.1.6	How Figures Display	83
5.2	Connected System Support for Password Synchronization	84
5.2.1	Systems that Support Bidirectional Password Synchronization	85
5.2.2	Systems that Accept Passwords from Identity Manager	85
5.2.3	Systems that Don't Accept or Provide Passwords	86
5.2.4	Systems that Don't Support Password Synchronization	87
5.3	Prerequisites for Password Synchronization	87
5.3.1	Support for Universal Password	88
5.3.2	Password Synchronization Capabilities Declared in the Driver Manifest	88
5.3.3	Controlling Password Synchronization by Using Global Configuration Values	88
5.3.4	Policies Required in the Driver Configuration	91
5.3.5	Filters You Install on the Connected System to Capture Passwords	95
5.3.6	NMAS Password Policies You Create for Users	95
5.3.7	NMAS Login Methods	95
5.4	Preparing to Use Identity Manager Password Synchronization and Universal Password	95
5.4.1	Switching Users from NDS Password to Universal Password	96
5.4.2	Helping Users Change Passwords	96
5.4.3	Preparing to Use Universal Password	97
5.4.4	Matching Containers	98
5.4.5	Setting Up E-Mail Notification	98
5.5	Configuring and Synchronizing a New Driver	98
5.6	Upgrading Password Synchronization 1.0	100
5.7	Upgrading Existing Driver Configurations to Support Password Synchronization	100
5.7.1	Step 1: Convert the Driver to Identity Manager 3.0.1 Format	101
5.7.2	Step 2: Add to the Driver Configuration	104
5.7.3	Step 3: Change Filter Settings	105
5.7.4	Step 4: Setting Up Password Synchronization Flow	107
5.8	Implementing Password Synchronization	108
5.8.1	Overview of Identity Manager Relationship to NMAS	108
5.8.2	Scenario 1: Using NDS Password to Synchronize between Two Identity Vaults	110
5.8.3	Scenario 2: Synchronizing by Using Universal Password	112
5.8.4	Scenario 3: Synchronizing an Identity Vault and Connected Systems, with Identity Manager Updating the Distribution Password	122
5.8.5	Scenario 4: Tunneling—Synchronizing Connected Systems but not an Identity Vault, with Identity Manager Updating the Distribution Password	131

5.8.6	Scenario 5: Synchronizing Application Passwords to the Simple Password.	135
5.9	Setting Up Password Filters	139
5.9.1	Setting Up Password Synchronization Filters for Active Directory and NT Domain	139
5.9.2	Setting Up Password Synchronization Filters for NIS	139
5.10	Managing Password Synchronization	140
5.10.1	Setting the Flow of Passwords Across Systems	140
5.10.2	Enforcing Password Policies on Connected Systems	141
5.10.3	Keeping the eDirectory Password Separate from the Synchronized Password	141
5.11	Checking the Password Synchronization Status for a User	142
5.12	Configuring E-Mail Notification	143
5.12.1	Prerequisites	144
5.12.2	Setting Up the SMTP Server To Send E-Mail Notification	145
5.12.3	Setting Up E-Mail Templates for Notification	146
5.12.4	Providing SMTP Authentication Information in Driver Policies	146
5.12.5	Adding Your Own Replacement Tags to E-Mail Notification Templates	148
5.12.6	Sending E-Mail Notifications to the Administrator	154
5.12.7	Localizing E-Mail Notification Templates	154
5.13	Troubleshooting Password Synchronization	155
6	Creating and Using Entitlements	157
6.1	Terminology	158
6.2	Creating Entitlements: Overview	158
6.2.1	Identity Manager Drivers with Preconfigurations that Support Entitlements	159
6.2.2	Enabling Entitlements on Other Identity Manager Drivers	160
6.3	Entitlement Prerequisites	161
6.4	Writing Entitlements in XML through iManager	162
6.4.1	What the Active Directory Driver Adds When Entitlements Are Enabled	162
6.4.2	Using Novell's Entitlement Document Type Definition (DTD)	166
6.4.3	Explaining the Entitlement DTD	168
6.4.4	Creating Entitlements Through Designer	171
6.4.5	Creating and Editing Entitlements in iManager	171
6.4.6	Example Entitlements To Help You Create Your Own Entitlements	172
6.4.7	Completing the Creating Entitlements Steps	176
6.5	Managing Role-Based Entitlements Overview	177
6.5.1	How the Entitlement Service Driver Works	177
6.6	Creating an Entitlements Service Driver Object	179
6.7	Creating Entitlement Policies	180
6.7.1	Defining Membership for an Entitlement Policy	182
6.7.2	Choosing Entitlements for an Entitlement Policy	183
6.8	Conflict Resolution between Role-Based Entitlement Policies	187
6.8.1	Conflict Overview	188
6.8.2	Changing the Conflict Resolution Method for an Individual Entitlement	189
6.8.3	Prioritizing Entitlement Policies	191
6.9	Troubleshooting Role-Based Entitlements	192
6.10	Entitlement Elements that Apply To Role-Based Entitlements and Workflow-Based Provisioning Entitlements	193
6.10.1	Controlling the Meaning of Granting or Revoking Entitlements	193
6.10.2	Preventing Data Loss	193
6.10.3	Password Synchronization and Entitlements	194
7	Security: Best Practices	195
7.1	Using SSL	195
7.2	Securing Access	195
7.3	Managing Passwords	195

7.4	Creating Strong Password Policies	196
7.5	Securing Connected Systems	197
7.6	Designer for Identity Manager	198
7.7	Industry Best Practices for Security	198
7.8	Tracking Changes to Sensitive Information	199
7.8.1	Logging Events by Using iManager	199
7.8.2	Logging Events by Using Designer	200
8	Managing Engine Services	205
8.1	Entitlements Service Driver	205
8.2	Manual Task Service Driver	205
8.2.1	Installing	205
8.2.2	Overview	206
8.2.3	Configuring	213
8.2.4	Additional Information	220
9	High Availability	221
9.1	Configuring eDirectory and Identity Manager for Use with Shared Storage on Linux and UNIX . 221	
9.1.1	Installing eDirectory	222
9.1.2	Installing Identity Manager	222
9.1.3	Sharing NICI Data	222
9.1.4	Sharing eDirectory and Identity Manager Data	223
9.1.5	Identity Manager Driver Considerations	224
9.2	Case Study for SuSE Linux	225
10	Logging and Reporting Using Novell Audit	227
10.1	Overview	227
10.2	Novell Audit	227
10.3	Setting Up Novell Audit	228
10.3.1	Setting Up the Platform Agent	228
10.3.2	Setting Up the Secure Logging Server	230
10.4	Logging Configuration	230
10.4.1	Selecting Events to Log	230
10.4.2	User-Defined Events	235
10.4.3	eDirectory Objects	237
10.5	Querying and Reporting	238
10.5.1	Identity Manager Reports	238
10.5.2	Viewing Identity Manager Events	238
10.6	Sending Notifications Based on Events	239
10.7	Using Status Logs	239
10.7.1	Setting the Maximum Log Size	239
10.7.2	Viewing Status Logs	241
A	DirXML Command Line Utility	243
A.1	Interactive Mode	243
A.2	Command Line Mode	251

B	Options for Configuring a Remote Loader	255
C	Identity Manager Events and Reports	263
C.1	Engine Events	263
C.2	Server Events	272
C.3	Remote Loader Events	274
C.4	Detail Portlets	275
C.5	Change Password Portlet	275
C.6	Forgot Password Change Password Portlet	276
C.7	Search List Portlet	276
C.8	Create Portlet	277
C.9	Security Context	277
C.10	Workflow	279
C.11	Reports	282
D	Manual Task Service Driver: Replacement Data	291
D.1	Data Security	291
D.2	XML Elements	292
D.2.1	<replacement-data>	292
D.2.2	<item>	293
D.2.3	<url-data>	294
D.2.4	<url-query>	295
E	Manual Task Service Driver: Automatic Replacement Data Items	297
E.1	Subscriber Channel Automatic Replacement Data	297
E.2	Publisher Channel Automatic Replacement Data	297
F	Manual Task Service Driver: Template Action Elements Reference	299
F.1	<form:input>	299
F.2	<form:if-item-exists>	299
F.3	<form:if-multiple-items>	300
F.4	<form:if-single-item>	300
F.5	<form:menu>	301
G	Manual Task Service Driver: <mail> Element Reference	303
G.1	<mail>	303
G.2	<to>	303
G.3	<cc>	303
G.4	<bcc>	303
G.5	<from>	303
G.6	<reply-to>	303
G.7	<subject>	304
G.8	<message>	304
G.9	<stylesheet>	304
G.10	<template>	304
G.11	<filename>	304
G.12	<replacement-data>	304
G.13	<resource>	305

G.14	<attachment>	305
H	Manual Task Service Driver: Data Flow Scenario for New Employee	307
H.1	Subscriber Channel Configuration	307
H.2	Publisher Channel Configuration	307
H.3	Description of Data Flow	307
I	Manual Task Service Driver: Custom Element Handlers for the Subscriber Channel	319
I.1	Constructing URLs for Use with the Publisher Channel Web Server	319
I.2	Constructing Message Documents using Stylesheets and Template Documents	320
I.3	SampleCommandHandler.java	320
I.3.1	Compiling the SampleCommandHandler Class	320
I.3.2	Trying the SampleCommandHandler Class	320
J	Manual Task Service Driver: Custom Servlets for the Publisher Channel	321
J.1	Using the Publisher Channel	321
J.2	Authentication	321
J.3	SampleServlet.java	321
J.3.1	Compiling the SampleServlet Class	321
J.3.2	Trying the SampleServlet Class	322
K	Documentation Update	323
K.1	August 14, 2007	323
K.1.1	Managing the Identity Manager Drivers	323
K.2	August 10, 2007	323
K.2.1	Options for Configuring a Remote Loader Appendix B	323

About This Guide

Novell® Identity Manager, formerly DirXML®, 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 \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

Additional Documentation

For documentation on installing and upgrading Identity Manager, see the *Identity Manager 3.0.1 Installation Guide*.

For documentation on Identity Manager policies and filters, see the *Policy Builder and Driver Customization Guide*.

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

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

For documentation on using the Identity Manager drivers, see the *Identity Manager Driver Documentation Web site* (<http://www.novell.com/documentation/idmdrivers/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 Identity Manager

3.0.1 Architecture

Identity Manager has three major components.

- ♦ [Section 1.2, “Identity Manager,” on page 14](#)
- ♦ [Section 1.3, “User Application,” on page 21](#)
- ♦ [Section 1.4, “Designer,” on page 21](#)

1.1 Terminology Changes from Earlier Versions

If you have not used DirXML[®] 1.1a or Identity Manager 2.0, you do not need to review this section.

In DirXML 1.1a, the term “rule” was used to describe a set of rules, the individual rules in the set, and the conditions and actions within the individual rules, depending on the context. This overlap causes confusion when the context is not clear.

In Identity Manager 2, the term “policy” is now used to replace the term “rule”, when describing the high-level transformation that is occurring. You now define a set of policies, where each policy contains one or more rules. The term “rule” is now used to describe only an individual set of conditions and actions.

The following table shows the terminology changes from DirXML 1.1a to Identity Manager 2.x.

Table 1-1 Terminology Changes from DirXML 1.1a to Identity Manager 2.x

Item Being Described	DirXML 1.1a Terminology	Identity Manager 2.x Terminology
Set of transformations	Rule	Set of policies
An individual transformation within a set	Rule	“Policies”
The conditions and actions within an individual transformation	Rule	Rule

The following table shows the terminology changes from Identity Manager 2.x to Identity Manager 3.0.1.

Table 1-2 Terminology Changes from Identity Manager 2.x to Identity Manager 3.0.1

Item Being Described	Identity Manager 2.x Terminology	Identity Manager 3.0.1 Terminology
The product	DirXML	Identity Manager

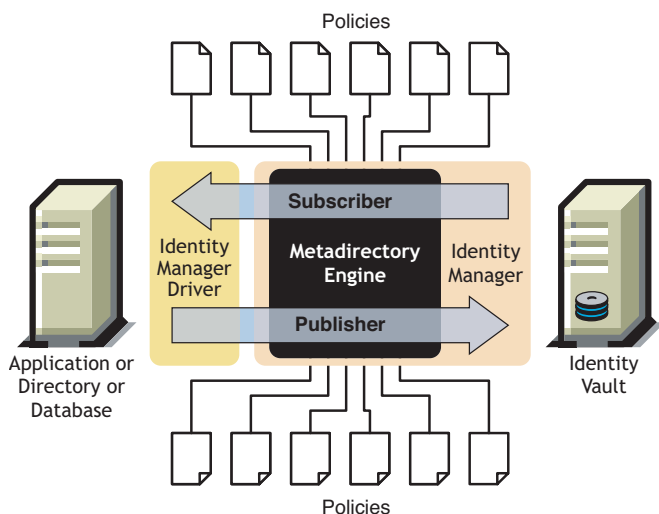
Item Being Described	Identity Manager 2.x Terminology	Identity Manager 3.0.1 Terminology
A server that has the product installed	DirXML server	Metadirectory server
A server in the application or database the data is synchronizing with	DirXML connected system server	Connected system server
Where the objects are stored	eDirectory™	Identity Vault
The processing component	DirXML engine	Metadirectory Engine

1.2 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



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 “[Introduction to Policies](#)” in the *Policy Builder and Driver Customization Guide*.
- ◆ 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.

1.2.1 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 style sheets, but the Policy Builder is simpler to use. For more information about the Policy Builder or DirXML Script, see the *Policy Builder and Driver Customization Guide*.

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.

1.2.2 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.

1.2.3 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.

1.2.4 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.

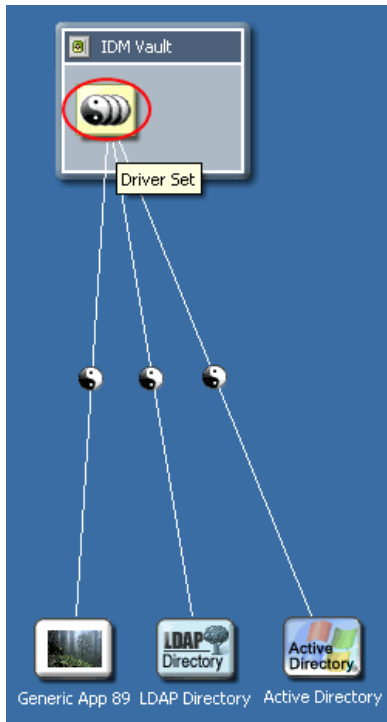
1.2.5 Driver Set

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.

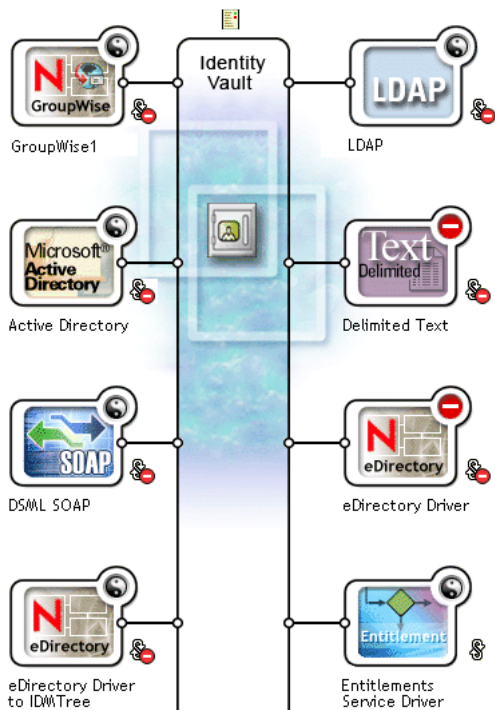
Figure 1-2 Driver Set in Designer



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

Figure 1-3 Driver Set in iManager

Driver Set: [Driver Set.Novell](#) [Identity Manager Bundle Edition](#)



From the Modeler in Designer (shown above [Figure 1-2 on page 17](#)) or the Overview page in iManager (shown above [Figure 1-3 on page 17](#)), 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

1.2.6 Driver Object

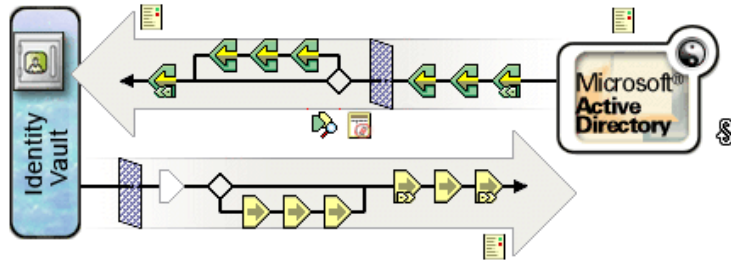
A Driver object represents a driver that connects to the connected system that integrates with the Identity Vault. The following components comprise the driver object and its configuration parameters:

- ◆ 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 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: Active Directory.DriverSet.South.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 \(http://www.novell.com/documentation/edir88/index.html?page=/documentation/edir88/edir88/data/fbachifb.html\)](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.

1.2.7 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

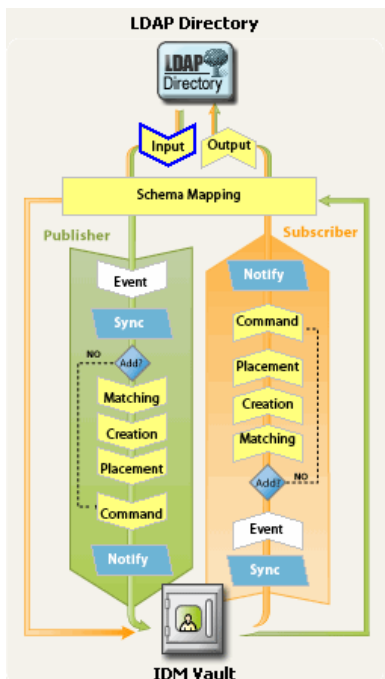
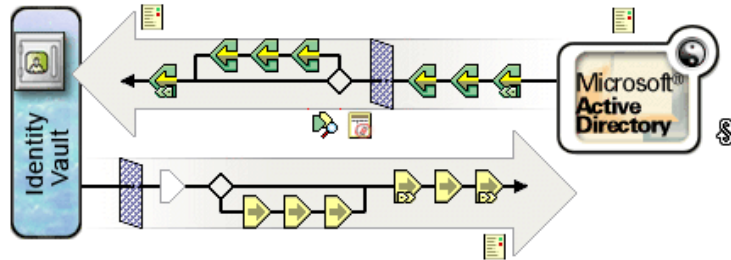


Figure 1-6 The Publisher and Subscriber channels in iManager

Driver: Active Directory.DriverSet.South.Novell



1.2.8 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.

1.2.9 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 the *Policy Builder and Driver Customization Guide*.

1.2.10 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

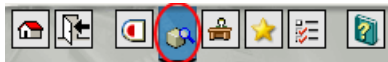
After an association between objects is formed, this association remains in effect until the objects are deleted or the association is deleted by an administrator.

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.

1.3 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 \(http://www.novell.com/documentation/idm/\)](http://www.novell.com/documentation/idm/).

1.4 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 for Identity Manager: Administration Guide \(http://www.novell.com/documentation/designer/\)](http://www.novell.com/documentation/designer/).

Managing Identity Manager Drivers

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 Format,” on page 25](#)
- ◆ [Section 2.4, “Starting, Stopping, or Restarting a Driver,” on page 25](#)
- ◆ [Section 2.5, “Driver Parameters,” on page 25](#)
- ◆ [Section 2.6, “Using Global Configuration Values,” on page 26](#)
- ◆ [Section 2.7, “Using the DirXML Command Line Utility,” on page 26](#)
- ◆ [Section 2.8, “Viewing Versioning Information,” on page 26](#)
- ◆ [Section 2.9, “Using Named Passwords,” on page 31](#)
- ◆ [Section 2.10, “Reassociating a Driver Object with a Server,” on page 37](#)
- ◆ [Section 2.11, “Adding Driver Heartbeat,” on page 38](#)

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 [“Creating a Driver Object” on page 23](#).
- ◆ 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 Mark *Import a driver configuration from the server (.XML file)* and select the .xml file, then click *Next*.
The driver configuration file gets installed on the Web server when you set up iManager.
- 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. Should you want to use Entitlements in the future, you must to 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 in that 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 Drivers*.
- 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 application 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 application'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 will continue to run with Identity Manager.

The Metadirectory engine that ships with Identity Manager 3.0.1 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 3.0.1 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 3.0.1 format.

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

The supported upgrade path from DirXML 1.1a is to install Identity Manager 3.0.1. The Identity Manager 3.0.1 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 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.
You are prompted to convert the driver to the new format.
- 4 Follow the steps in the wizard to complete the conversion.

2.4 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* if the driver is running.

2.5 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.

2.6 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.

2.7 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 243](#) for detailed information about the DirXML Command Line Utility. For daily tasks, use iManager or Designer.

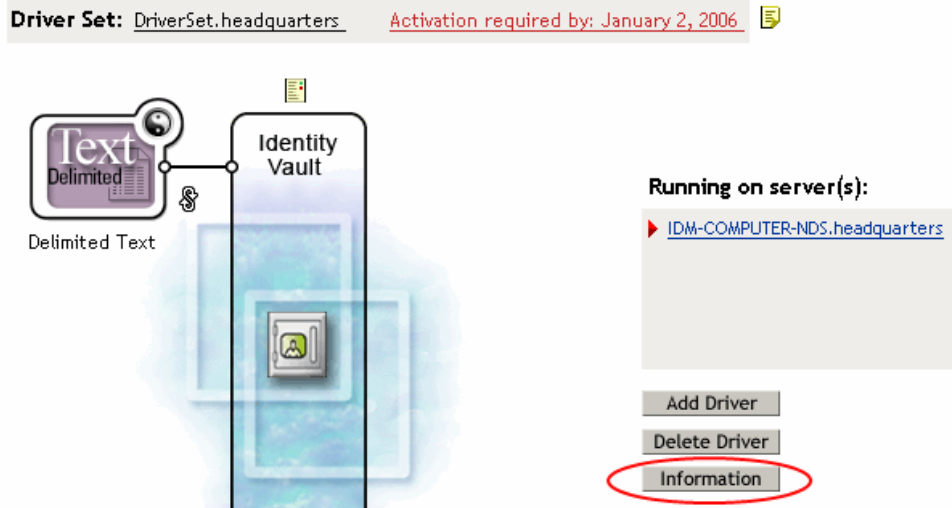
2.8 Viewing Versioning Information

The Versioning Discovery Tool enables you to do the following:

- ♦ [Section 2.8.1, "View a Hierarchical Display of Versioning Information," on page 27](#)
- ♦ [Section 2.8.2, "View the Versioning Information as a Text File," on page 28](#)
- ♦ [Section 2.8.3, "Save Versioning Information," on page 30](#)

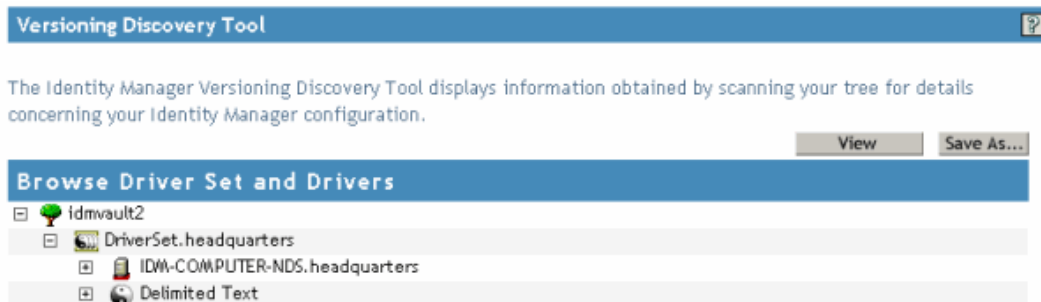
2.8.1 View a Hierarchical Display of Versioning Information

- 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.



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.



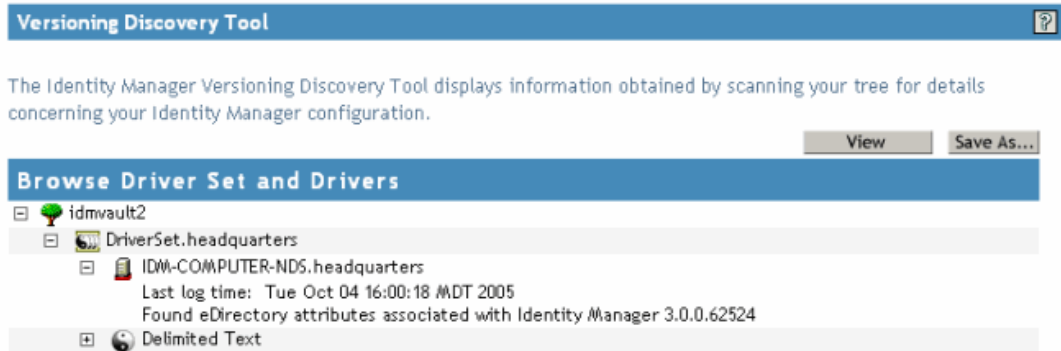
The unexpanded hierarchical view displays the following:

- ♦ 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

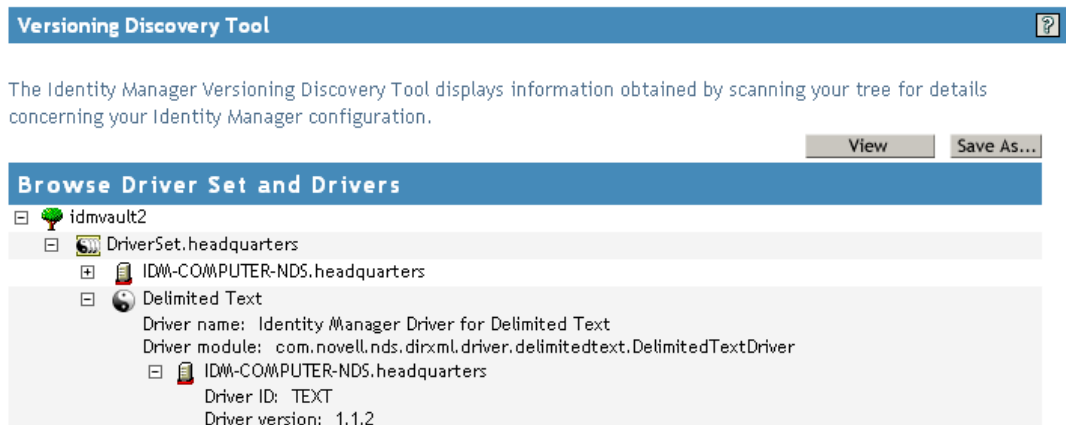
- 4 View versioning information related to servers by expanding the server icon.



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 versioning information related to drivers by expanding the driver icon.



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.delimitedtext.DelimitedTextDriver`)

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

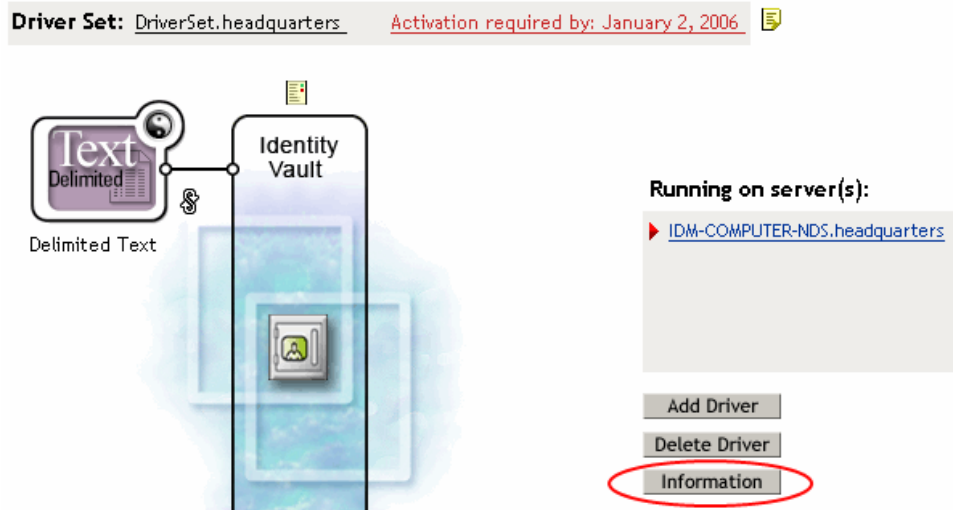
- ◆ The driver ID
- ◆ The version of the instance of the driver running on that server

2.8.2 View the Versioning 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.

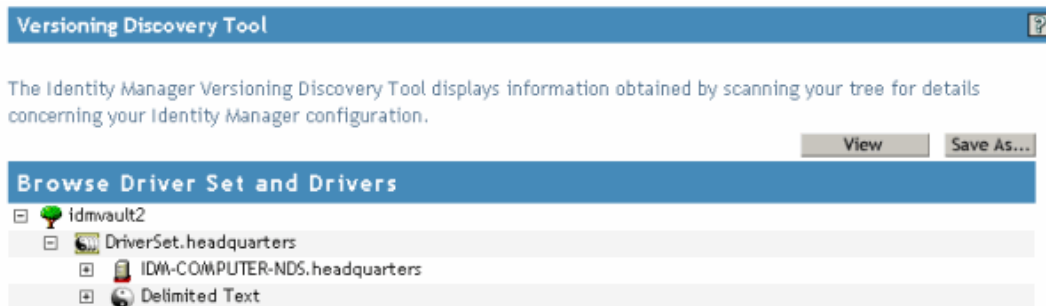
- 1 In iManager click *Identity Manager > Identity Manager Overview*, then click *Search* to find your Driver Set.

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

Versioning Discovery Tool - Report Viewer

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

Version Query started Thursday, October 13, 2005 5:12:28 PM MDT

Parameter Summary:
  Default server's DN:  IDM-COMPUTER-NDS.headquarters
  Default server's IP address:  137.65.149.18
  Logged in as Admin, context headquarters
  Tree name:  idmvault2
  Found 1 Identity Manager Drivers

Driver Set:  DriverSet.headquarters
  Driver Set running on Identity Vault:  IDM-COMPUTER-NDS.headquarters
  Last log time:  Tue Oct 04 16:00:18 MDT 2005
  Found eDirectory attributes associated with Identity Manager 3.0.0.6252
  Driver:  Delimited Text.DriverSet.headquarters
  Driver name:  Identity Manager Driver for Delimited Text
  Driver module:  com.novell.nds.dirxml.driver.delimitedtext.DelimitedTex
  Driver Set running on Identity Vault:  IDM-COMPUTER-NDS.headquarters
  Driver ID:  TEXT
  Driver version:  1.1.2

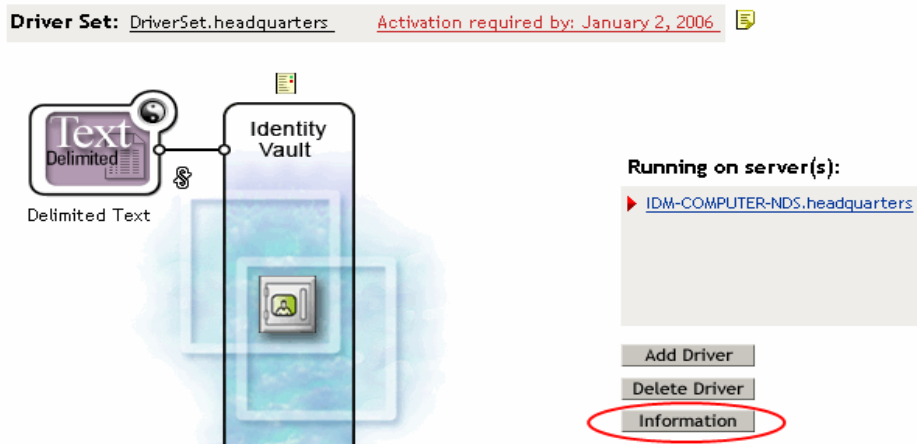
Version Query completed Thursday, October 13, 2005 5:12:28 PM MDT
```

OK

2.8.3 Save Versioning Information

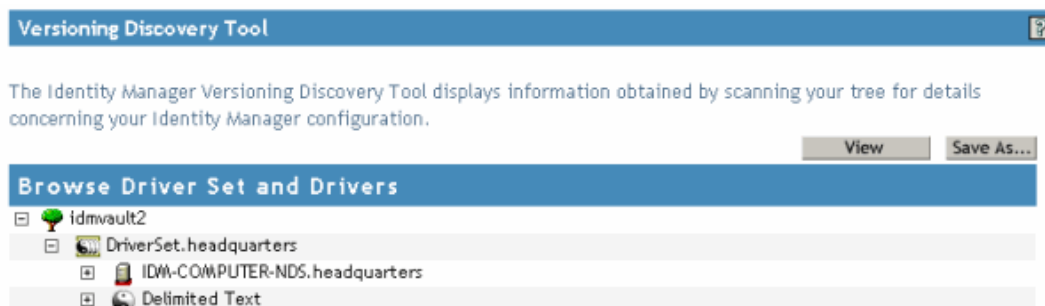
You can save versioning information to a text file on your local or network drive.

- 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*.



- 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.

2.9 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.

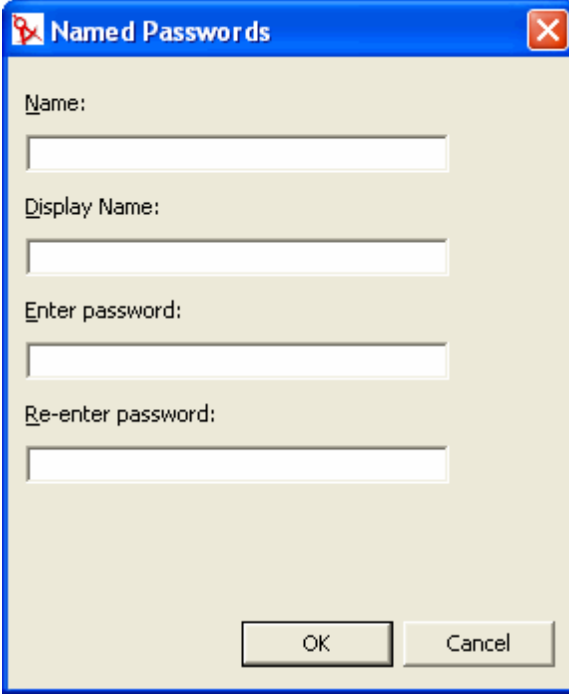
NOTE: The sample configurations provided for the Identity Manager Driver for Lotus Notes include an example of using Named Passwords in this way. The Notes driver shim has also been customized to support other ways of using Named Passwords, and examples of those methods are also included. For more information,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

In this section:

- ◆ [Section 2.9.1, “Configuring Named Passwords Using Designer,” on page 32](#)
- ◆ [Section 2.9.2, “Configuring Named Passwords Using iManager,” on page 32](#)
- ◆ [Section 2.9.3, “Using Named Passwords in Driver Policies,” on page 34](#)
- ◆ [Section 2.9.4, “Configuring Named Passwords Using the DirXML Command Line Utility,” on page 34](#)

2.9.1 Configuring Named Passwords Using Designer

- 1 Select the Driver object, then right-click and select *Properties*.
- 2 Select *Named Password*, click *New*.



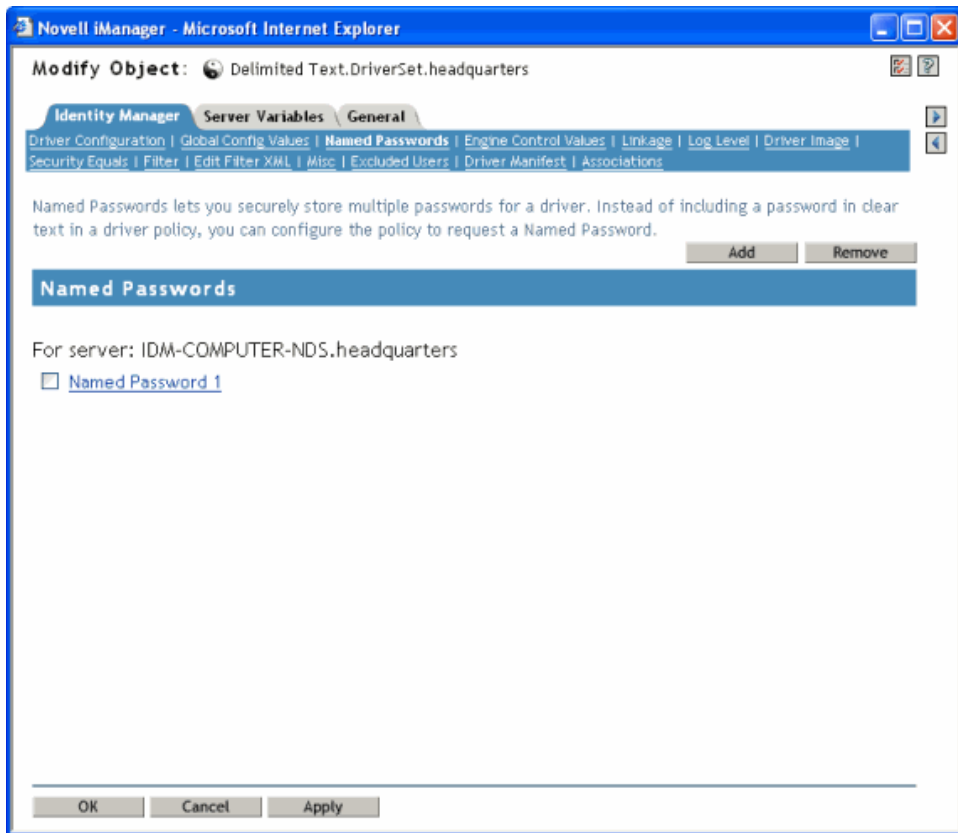
The image shows a dialog box titled "Named Passwords". It has a blue title bar with a red "X" icon in the top right corner. The dialog contains four text input fields, each with a label above it: "Name:", "Display Name:", "Enter password:", and "Re-enter password:". At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

- 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.

2.9.2 Configuring Named Passwords 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.



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



- 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.

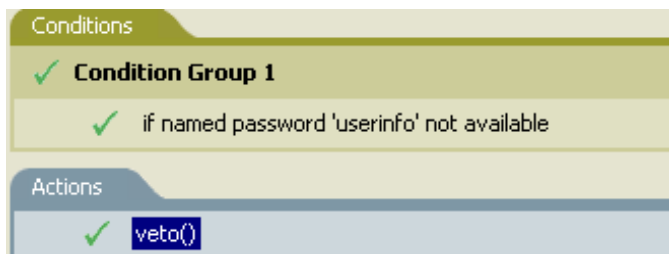
2.9.3 Using Named Passwords in Driver Policies

- ♦ [“Using the Policy Builder” on page 34](#)
- ♦ [“Using XSLT” on page 34](#)

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 2-1 A Policy Using Named Password



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/>
```

2.9.4 Configuring Named Passwords Using the DirXML Command Line Utility

- ♦ [“Creating a Named Password in the DirXML Command Line Utility” on page 34](#)
- ♦ [“Removing a Named Password in the DirXML Command Line Utility” on page 36](#)

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 243](#).

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

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: Check object password
- 10: Initialize new driver object
- 11: Passwords operations
- 12: Cache operations

99: Exit

Enter choice:

5 Enter 11 for password operations.

The following list of options appears.

Select a password operation

- 1: Set shim password
- 2: Reset shim password
- 3: Set named password
- 4: Clear named password(s)
- 5: List named passwords

99: Exit

Enter choice:

6 Enter 3 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 243](#).

- 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
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: Check object password
10: Initialize new driver object
11: Passwords operations
12: Cache operations
99: Exit
```

Enter choice:

- 5 Enter 11 for password operations.

The following list of options appears.

Select a password operation

```
1: Set shim password
2: Reset shim password
3: Set named password
4: Clear named password(s)
5: List named passwords
99: Exit
```

Enter choice:

- 6** (Optional) Enter 5 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 4 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 named password
4: Clear named password(s)
5: List named passwords
99: Exit
```

Enter choice:

- 10** (Optional) Enter 5 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.

2.10 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 object 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 object.

2.11 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>
```

2.12 Viewing Identity Manager Processes

To view Identity Manager 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.

In order to see Identity Manager processes in DSTRACE, values are added to the Driver Set and the Drive objects. You can do this in Designer and iManager.

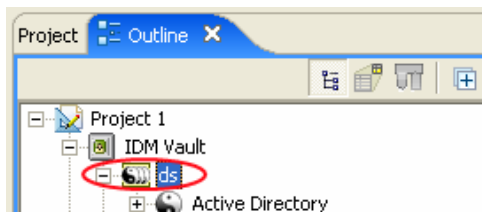
- [Section 2.12.1, “Adding Trace Levels in Designer,” on page 39](#)
- [Section 2.12.2, “Adding Trace Levels in iManager,” on page 41](#)
- [Section 2.12.3, “Capturing Identity Manager Processes to a File,” on page 42](#)

2.12.1 Adding Trace Levels in Designer

You can add trace levels to the Driver Set object or to each Driver object.

Driver Set

- 1 In an open project in Designer, select the Driver Set object 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 40](#) for more information about the Driver Set trace parameters.

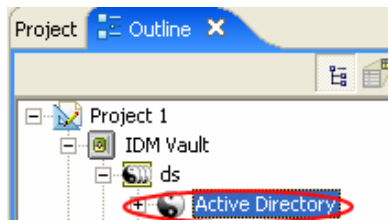
If you set the trace level on the Driver Set object, all drivers appear in the DSTRACE logs.

Table 2-1 Driver Set Trace Parameters

Parameter	Description
Driver trace level	<p>As the Driver object trace level increases, the amount of information displayed in DSTRACE increases.</p> <p>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.</p>
XSL trace level	<p>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.</p>
Java debug port	<p>Allows developers to attach a Java debugger.</p>
Java trace file	<p>When a value is set in this field, all Java information for the Driver Set object is written to a file. The value for this field is the path for that file.</p> <p>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.</p>
Trace file size limit	<p>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.</p> <hr/> <p>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.</p>

Driver

- 1 In an open project in Designer, select the Driver object in the Outline view.



- 2 Right-click and select *Properties*, then click *Trace*.
- 3 Set the parameters for tracing, then click *OK*. See [Table 2-2 on page 41](#) for more information about these parameters.

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

Table 2-2 *Driver Trace Parameters*

Parameter	Description
Trace level	<p>As the Driver object trace level increases, the amount of information displayed in DSTRACE increases.</p> <p>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.</p> <p>If you select <i>Use setting from Driver Set</i>, the value is taken from the Driver Set object.</p>
Trace file	<p>Specify a file name and location of where the Identity Manager information is written for the selected driver.</p> <p>If you select <i>Use setting from Driver Set</i>, the value is taken from the Driver Set object.</p>
Trace file size limit	<p>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.</p> <p>If you select <i>Use setting from Driver Set</i>, the value is taken from the Driver Set object.</p> <hr/> <p>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.</p> <hr/>
Trace name	<p>The driver trace messages are prepended with the value entered instead of the driver name. Use if the driver name is very long.</p>

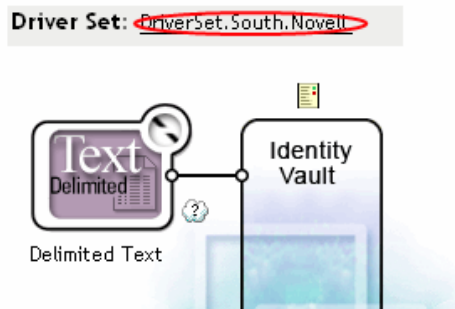
2.12.2 Adding Trace Levels in iManager

You can add trace levels to the Driver Set object or to each Driver object.

Driver Set

- 1 In iManager select *Identity Manager > Identity Manager Overview*.
- 2 Browse to the Driver Set object, then click *Search*.

- 3 Click on the Driver Set name.



- 4 Select the *Misc* tab for the Driver Set object.
- 5 Set the parameters for tracing, then click *OK*. See [Table 2-1 on page 40](#) for more information about these parameters.

Driver

- 1 In iManager select *Identity Manager > Identity Manager Overview*.
- 2 Browse to the Driver Set object where the Driver object resides, then click *Search*.
- 3 Click the upper right corner of the Driver object, then click *Edit properties*.
- 4 Select the *Misc* tab for the Driver object.
- 5 Set the parameters for tracing, then click *OK*. See [Table 2-2 on page 41](#) for more information.

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

2.12.3 Capturing Identity Manager Processes to a File

To save Identity Manager processes to a file, it is saved through the parameter on the Driver object or through DSTRACE. The parameter on the Driver object is the Trace file parameter.

The following methods help 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.
- ◆ `ndsmonitor` 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`

Setting Up a Connected System

3

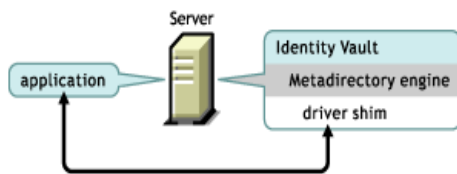
This section provides information on the following:

- ♦ [Section 3.1, “Overview,” on page 45](#)
- ♦ [Section 3.2, “Providing for Secure Data Transfers,” on page 47](#)
- ♦ [Section 3.3, “Setting Up Remote Loaders,” on page 49](#)
- ♦ [Section 3.4, “Configuring the Identity Manager Drivers for Use with Remote Loaders,” on page 67](#)

3.1 Overview

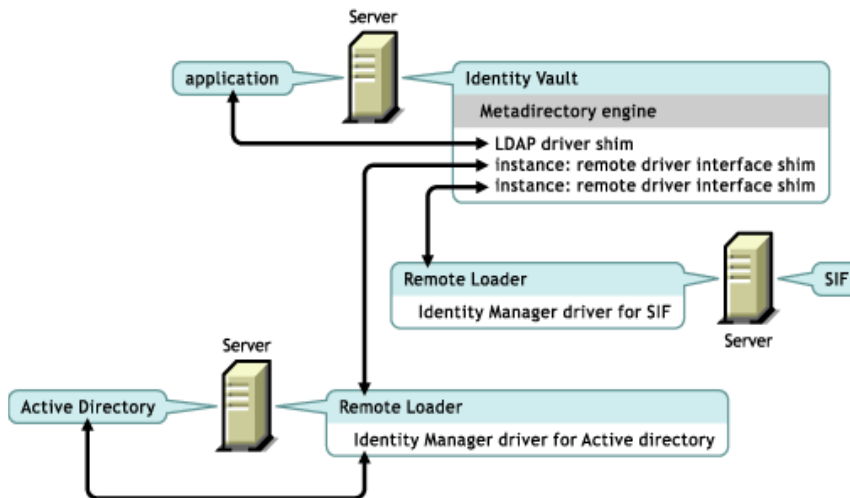
As the following figure illustrates, the Metadirectory engine runs on a server as part of eDirectory. An Identity Manager driver shim and its configured driver communicate with an application and with the Metadirectory engine.

Figure 3-1 *The Metadirectory Engine Running Under eDirectory*



As the following figure illustrates, a connected system extends Identity Manager functionality across applications:

Figure 3-2 *A Connected System, Including the Remote Loader*



A connected system requires a Remote Loader. This service enables the Metadirectory engine to exchange data with Identity Manager drivers running as different processes and in different locations, including the following:

- ◆ As separate processes on the server where the Metadirectory engine is running

The Metadirectory engine runs as part of the 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, you might want the Identity Manager driver to run as a separate process on the server. Typically, though, the Identity Manager drivers run on separate servers.

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

- ◆ On servers other than the one where the Metadirectory engine is running

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 and 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.

Identity Manager provides Remote Loader functionality through `dirxml_remote`, `rdxml`, or `dirxml_jremote`.

Dirxml_remote

`Dirxml_remote` is an executable that enables the Metadirectory engine to communicate with the Identity Manager drivers running on Windows.

The Remote Loader Console uses `dirxml_remote.exe`. If you specify `dirxml_remote.exe` from the command line, without any parameters, the Remote Loader Application Wizard is launched. If you type `dirxml_remote.exe` and then pass in parameters, the Remote Loader is started.

Rdxml

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

`Rdxml` can support both native and Java drivers.

Dirxml_jremote

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

Overview: Main Tasks

Using the Remote Loader involves the following tasks:

- ◆ If you plan to use the Secure Socket Layer (SSL), provide certificates for secure data transfers.
- ◆ Install, configure, and run the Remote Loader.
- ◆ Import, configure, and start the Identity Manager driver.

Some administrators prefer to import and configure the Identity Manager driver before setting up the Remote Loader. For example, the driver might already be running but you want to enable it to run remotely.

On the other hand, if the Remote Loader is running, you can import, configure, and start the driver, then immediately check whether proper communication is occurring among the Metadirectory engine, Remote Loader, and the Identity Manager driver.

3.2 Providing for Secure Data Transfers

If you plan to use the Secure Socket Layer (SSL) so that you can provide secure data transfers, complete the following tasks:

1. Create a server certificate.

If you are unfamiliar with certificates, 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

2. Export a self-signed certificate.

3.2.1 Creating a Server Certificate


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



Create Server Certificate Wizard

Welcome to the Create Server Certificate Wizard

Select the server which will own the certificate.

Server: RDev31 

Certificate nickname: remotecert

Creation method

Standard
(Default parameters)

Custom
(User specifies parameters)

Import
(Allows a PKCS12 file to provide the keys and certificates)

- 2 Select the server that will 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 48.

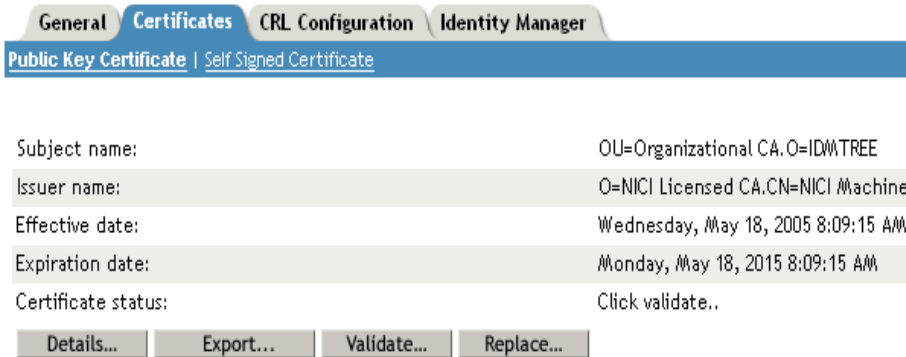
3.2.2 Exporting a Self-Signed Certificate

- 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*.



- 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, akranes-tree CA.b64), then click *Next*.



Select an output format.

- File in binary DER format
- File in Base64 format

- 6 Click the link to *Save the exported certificate to a file*, specify a filename, specify a location, then click *Save*.
Rootfile names require .pem as an extension.
- 7 In the Save As dialog box, copy this file to a local directory.
- 8 Click *Close*.

3.3 Setting Up Remote Loaders

This section provides information on the following:

- ♦ [Section 3.3.1, “Installing Remote Loaders,” on page 50](#)
- ♦ [Section 3.3.2, “Configuring Remote Loader,” on page 52](#)
- ♦ [Section , “Setting Environment Variables on Solaris, Linux, or AIX,” on page 63](#)
- ♦ [Section , “Starting Remote Loader,” on page 64](#)[Section , “Stopping Remote Loader,” on page 66](#)
- ♦ [Section , “Stopping Remote Loader,” on page 66](#)

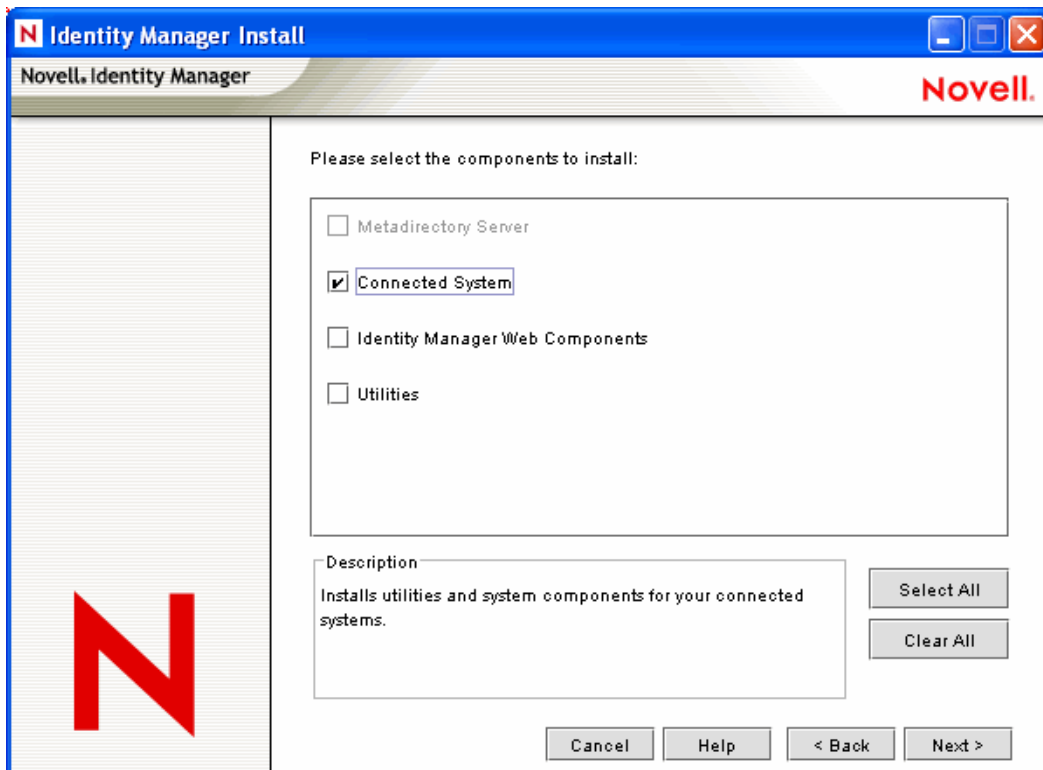
3.3.1 Installing Remote Loaders

This section provides information on the following:

- ♦ “Installing a Remote Loader on a Windows Server” on page 50
- ♦ “Installing a Remote Loader on Solaris, Linux, or AIX” on page 51
- ♦ “Installing a Remote Loader on HP-UX, AS/400, OS/390, or z/OS” on page 52

Installing a Remote Loader on a Windows Server

- 1 Run the Identity Manager installation program (for example, \nt\install.exe).
- 2 View the Welcome page, accept the license agreement, and view the two Overview pages.
- 3 In the Identity Manager Install dialog box, deselect all components except *Connected System*, then click *Next*.



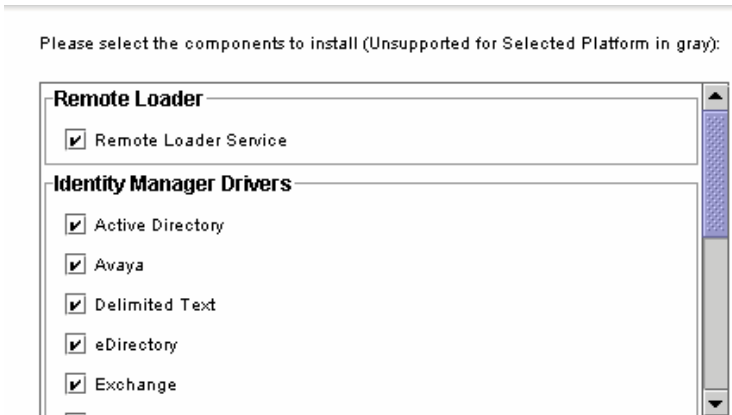
- 4 Select a location for the connected system (the Remote Loader and remote driver shims), then click *Next*.

Connected System will be installed at the following location

Installation Path

C:\Novell\RemoteLoader

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



- 6 Acknowledge the activation requirement, view products to be installed, then click *Finish*.
- 7 Select whether to place the Remote Loader Console icon on your desktop.

Installing a Remote Loader on Solaris, Linux, or AIX

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

After you expand the Identity Manager file that you downloaded from the Novell Web site, complete the following steps:

- 1 Run one of the following installation files, depending on your platform:
 - ♦ dirxml_solaris.bin
 - ♦ dirxml_linux.bin
 - ♦ dirxml_aix.bin
- 2 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
:
```

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

- 4 At 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: █

```

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	<code>dirxml_jremote.tar.gz</code>
AS/400	<code>dirxml_jremote.tar.gz</code>
z/OS	<code>dirxml_jremote_mvs.tar</code>
OS/390	<code>dirxml_jremote_mvs.tar</code>

- 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.

3.3.2 Configuring 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.

- ◆ [“Configuring the Remote Loader on Windows” on page 53](#)

- ◆ “Configuring the Remote Loader by Using Command Line Options” on page 58
- ◆ “Starting Remote Loader” on page 64
- ◆ “Stopping Remote Loader” on page 66

Configuring the Remote Loader on Windows

- ◆ “Using the Remote Loader Console Utility” on page 53
- ◆ “Adding a Remote Loader Instance” on page 54
- ◆ “Editing a Remote Loader Instance” on page 58

Using the Remote Loader Console Utility

The Remote Loader Console only runs on Windows. The Console enables you to manage all Identity Manager drivers running under the Remote Loader on that computer:

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.

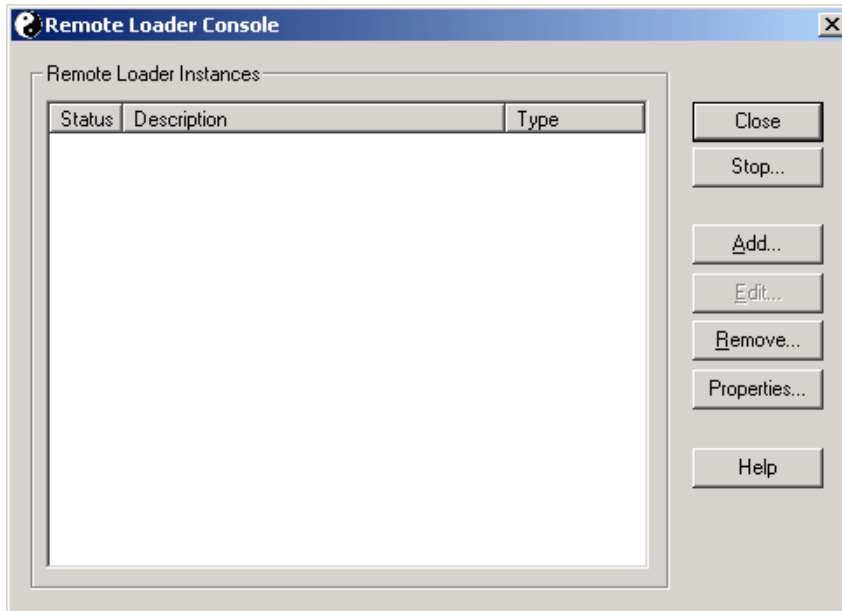
To launch the Remote Loader Console, click the Remote Loader Console icon on your desktop.

Figure 3-3 Remote Loader Console Icon



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

Figure 3-4 The Remote Loader Console



If you type `dirxml_remote.exe` from the command line, without any parameters, the Remote Loader Application Wizard is launched.

NOTE: Using the wizard and the Console together can cause unexpected behavior. Therefore, we recommend that you use the Remote Loader Console going forward and upgrade your existing configurations into the Console.

Adding a Remote Loader Instance

To add a Remote Loader instance, click Add, then provide the following information:

- ◆ “Remote Driver Configuration” on page 56
- ◆ “Communication Parameters” on page 56
- ◆ “Remote Loader Password” on page 57
- ◆ “Driver Object Password” on page 57
- ◆ “Secure Socket Link (Secure Socket Layer)” on page 57
- ◆ “Trace File” on page 58
- ◆ “Establish a Remote Loader Service for this Driver Instance” on page 58

Figure 3-5 Remote Loader Configuration Parameters

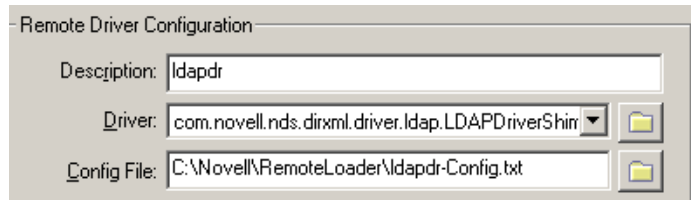
The image shows a Windows-style dialog box titled "Add" with a close button (X) in the top right corner. The dialog is divided into several sections for configuring a remote driver instance.

- Remote Driver Configuration:**
 - Description: [Empty text box]
 - Driver: [Dropdown menu showing "--none--"]
 - Config File: [Text box containing "C:\Novell\RemoteLoader\config.txt"]
- Communication:**
 - IP Address: [Dropdown menu showing "All"]
 - Connection Port - DitzXML Server: [Text box containing "8090"]
 - Command Port - Local host communication only: [Text box containing "8000"]
- Remote Loader Password:**
 - Password: [Empty password field]
 - Confirm: [Empty password field]
- Driver Object Password:**
 - Password: [Empty password field]
 - Confirm: [Empty password field]
- Secure Socket Link (SSL):**
 - Use an SSL Connection
 - Trusted Root File: [Empty text box]
- Trace File:**
 - Trace Level: [Dropdown menu showing "0"] with the text "No information display or tracking." to its right.
 - Trace File: [Text box containing "C:\Novell\RemoteLoader\trace.log"]
 - Maximum Disk Space Allowed for all Trace Logs (Mb):
 - Unlimited [Text box containing "0"]
- Establish a Remote Loader service for this driver instance.

On the right side of the dialog, there are two buttons: "OK" and "Cancel".

Remote Driver Configuration

Figure 3-6 Remote Driver Configuration

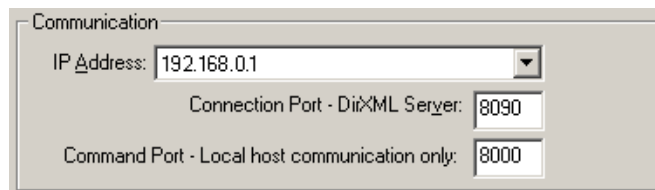


- ◆ Description: Specify a description to identify the Remote Loader instance.
- ◆ Driver: Browse to and select the appropriate shim for your driver.
- ◆ Config File: Specify a name for the configuration file.

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

Communication Parameters

Figure 3-7 Communication Parameters



- ◆ IP Address: Specify the IP address where the Remote Loader listens for connections from the metadirectory server.
- ◆ Connection Port - metadirectory server. 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.

- ◆ Command Port - Local Host Communication Only: 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.

Remote Loader Password

Figure 3-8 Remote Loader Password



Remote Loader Password

Password: xxxxxxx

Confirm: xxxxxxx

- ◆ Password: This password is used to control access to a Remote Loader instance for a driver.
The password must be the same case-sensitive password that you typed in the Enter the Remote Loader Password edit box in the Authentication section on the Identity Manager Configuration page, when you configured the driver.
- ◆ Confirm: Re-enter the password.

Driver Object Password

Figure 3-9 Driver Object Password



Driver Object Password

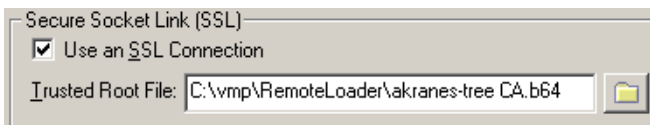
Password: xxxxxxx

Confirm: xxxxxxx

- ◆ Password: The Remote Loader uses this password to authenticate itself to the metadirectory server.
This password must be the same password you typed in the Driver Object Password edit box on the Driver Configuration page, when you configured the driver.
- ◆ Confirm: Re-enter the password.

Secure Socket Link (Secure Socket Layer)

Figure 3-10 Secure Socket Link (Secure Socket Layer)



Secure Socket Link (SSL)

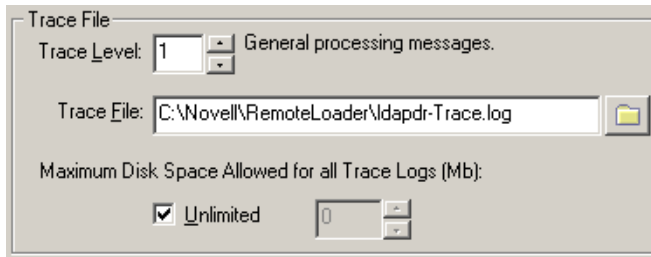
Use an SSL Connection

Trusted Root File: C:\vmp\RemoteLoader\akranes-tree CA.b64

- ◆ Use an SSL Connection: To specify an SSL connection, select this option.
- ◆ Trusted Root File: 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 48.

Trace File

Figure 3-11 Trace File



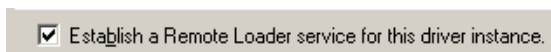
- ◆ Trace Level: For the Remote Loader instance to display a trace window that contains informational messages from both the Remote Loader and the driver, set a trace level greater than zero. The most common setting is trace level 3.

If the trace level is set to 0, the trace window won't appear or display messages.

- ◆ Trace File 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.
- ◆ Maximum Disk Space Allowed for all Trace Logs (MB): Specify the approximate maximum size that trace file data for this instance can occupy on disk.

Establish a Remote Loader Service for this Driver Instance

Figure 3-12 Establish a Remote Loader Service for this Driver Instance



- ◆ To configure the Remote Loader instance as a service, select this option. When the option is enabled, the operating system automatically starts the Remote Loader when the computer starts.

Editing a Remote Loader Instance

- 1 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 as when you add a Remote Loader instance.

Configuring the Remote Loader by Using Command Line Options

To run the Remote Loader, all platforms use a configuration file (for example, LDAPShim.txt). 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. For information on additional parameters, see [Appendix B, “Options for Configuring a Remote Loader,” on page 255](#).

- 1 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 Nsure 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 the connection to the metadirectory server running the Identity Manager remote interface shim by using the -connection option.

Type -connection "*parameter* [parameter] [parameter]".

For example, type one of the following:

```
-connection "port=8091 rootfile=server1.pem"
-conn "port=8091 rootfile=server1.pem"
```

All the parameters must be included within quotation marks. Parameters include the following:

Option	Secondary Name	Parameter	Description
-connection	-conn	connection configuration string	<p>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.</p> <p>Example:</p> <pre>-connection "port=8091 rootfile=server1.pem" -conn "port=8091 rootfile=server1.pem"</pre>
port		decimal port number	<p>A required parameter. It specifies the TCP/IP port on which the Remote Loader listens for connections from the remote interface shim.</p> <p>Example:</p> <pre>port=8090</pre>
address		IP address	<p>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.</p> <p>You have three options: address=<i>address number</i> address='localhost' Don't use this parameter.</p> <p>If you don't use the -address, the Remote Loader listens on all local IP addresses.</p> <p>Example: address=137.65.134.83</p>
rootfile			<p>A conditional parameter. If you are running SSL and need the Remote Loader to communicate with a native driver, type</p> <pre>rootfile='trusted certname'</pre>

Option	Secondary Name	Parameter	Description
keystore			<p>Conditional parameters. Used only for the Identity Manager application shims contained in .jar files.</p> <p>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.</p> <p>If you are running SSL and need the Remote Loader to communicate with a Java driver, type a key-value pair:</p> <pre>keystore='keystorename' storepass='password'</pre>
-storepass		storepass	<p>Used only for the Identity Manager application shims contained in .jar files. Specifies the password for the Java keystore specified by the keystore parameter.</p> <p>Example:</p> <pre>storepass=myspassword</pre> <p>This option applies only to the Java Remote Loader.</p>

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

Option	Secondary Name	Parameter	Description
-trace	-t	integer	<p>Specifies the trace level. This is only used when hosting an application shim. Trace levels correspond to those used on the metadirectory server.</p> <p>Example:</p> <pre>-trace 3 -t 3</pre>

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

Option	Secondary Name	Parameter	Description
-tracefile	-tf	filename	<p>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.</p> <p>Example:</p> <pre>-tracefile c:\temp\trace.txt -tf c:\temp\trace.txt</pre>

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

For example, type one of the following:

```
-tracefilemax 1000M  
-tfm 1000M
```

In this example, the tracefile can be only 1 GB.

Option	Secondary Name	Parameter	Description
-tracefilemax	-tfm	size	<p>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.</p> <p>The size parameter is the number of bytes. Specify the size by using the suffixes K, M, or G for kilobytes, megabytes, or gigabytes.</p> <p>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</p> <p>Example:</p> <pre>-tracefilemax 1000M -tfm 1000M</pre> <p>In this example, the tracefile can be only 1 GB.</p>

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

Option	Secondary Name	Parameter	Description
-class	-cl	Java class name	<p>Specifies the Java class name of the Identity Manager application shim that is to be hosted.</p> <p>For example, for a Java driver, type one of the following:</p> <pre>-class com.novell.nds.dirxml.driver.Idap.LDAPDriverShim -cl com.novell.nds.dirxml.driver.Idap.LDAPDriverShim</pre> <p>Java uses a keystore to read certificates. The -class option and the -module option are mutually exclusive.</p> <p>To see a list of the Java class name see Table B-2 on page 262 in Appendix B, “Options for Configuring a Remote Loader,” on page 255.</p>

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

9 Name and save the file.

You can change some settings while the Remote Loader is running. For information on these settings, refer to [Appendix B, “Options for Configuring a Remote Loader,” on page 255.](#)

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

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, enter the following commands:

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

Starting Remote Loader

- ♦ “Starting Remote Loader on Windows” on page 64
- ♦ “Starting Remote Loader from the Command Line” on page 64

Starting Remote Loader on Windows

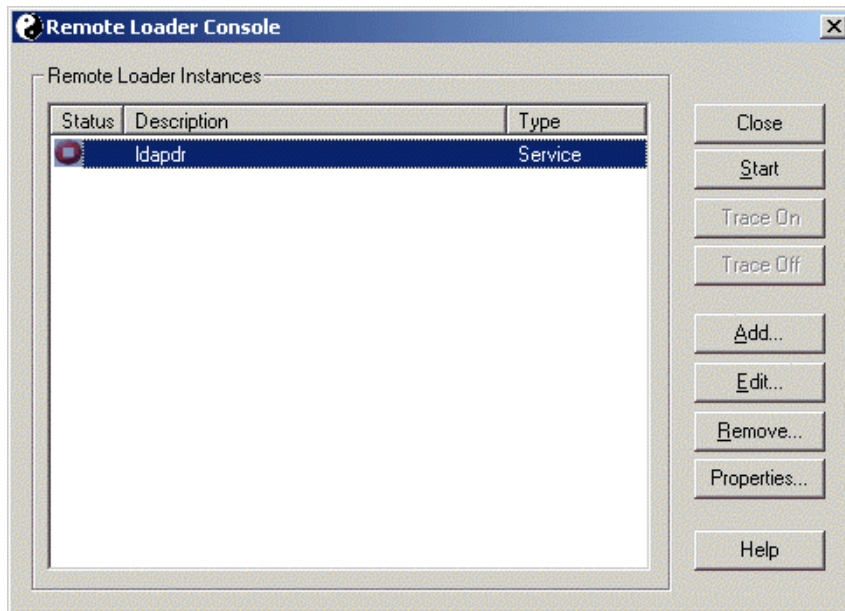
To run the Remote Loader on Windows:

Figure 3-13 Remote Loader Console Icon



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

Figure 3-14 The Remote Loader Console



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

Starting Remote Loader from the Command Line

On Solaris, Linux, or AIX, the binary component `rdxml` provides the Remote Loader functionality. This component is located in the `/usr/bin/` directory. On Windows, the default is `c:\novell\RemoteLoader`.

To run the Remote Loader:

- 1 Set the password.

Platform	Command
Windows	<code>dirxml_remote -config path_to_config_file -sp password password</code>
Solaris Linux AIX	<code>rdxml -config path_to_config_file -sp password password</code>
HP-UX AS/400 OS/390 z/OS	<code>dirxml_jremote -config path_to_config_file -sp password password</code>

Option	Secondary Name	Parameter	Description
-password	-p	password	<p>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.</p> <p>Example:</p> <pre>-password novell4 -p novell4</pre>
-setpasswords	-sp	password password	<p>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.</p> <p>Example:</p> <pre>-setpasswords novell4 staccato3 -sp novell4 staccato3</pre>

2 Start the Remote Loader.

Platform	Command
Windows	<code>dirxml_remote -config path_to_config_file</code>
Solaris Linux AIX	<code>rdxml -config path_to_config_file</code>
HP-UX AS/400 OS/390 z/OS	<code>dirxml_jremote -config path_to_config_file</code>

3 Using iManager, 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 will be shut 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 \(http://support.novell.com/cgi-bin/search/searchtid.cgi?/10097249.htm\)](http://support.novell.com/cgi-bin/search/searchtid.cgi?/10097249.htm).

Stopping Remote Loader

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

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 enter the Remote Loader password.

Scenario: Sufficient Rights. 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.

3.4 Configuring the Identity Manager Drivers for Use with Remote Loaders

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 additional and driver-specific information, refer to the relevant driver implementation guide.

- ♦ [Section 3.4.1, "Importing and Configuring a New Driver," on page 67](#)
- ♦ [Section 3.4.2, "Configuring an Existing Driver," on page 69](#)
- ♦ [Section 3.4.3, "Creating a Keystore," on page 70](#)

3.4.1 Importing and Configuring a New Driver

- 1 In Novell 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:



A screenshot of a configuration dialog box. At the top, the text "Driver is Local/Remote:" is followed by a dropdown menu. The dropdown menu is currently set to "Local", but "Remote" is highlighted in blue, indicating it is the selected option. Below the dropdown menu, there are four buttons: "<< Back", "Next >>", "Cancel", and "Finish".



A screenshot of a configuration dialog box showing four buttons: "<< Back", "Next >>", "Cancel", and "Finish".

3 Enter a remote hostname and port.

 **SAP-HR** (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.

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:

:

4 Type and re-enter a password for the Driver object.

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

Driver Password:

Reenter the password:

5 Enter 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 DirXML Remote Loader.

Remote Password:

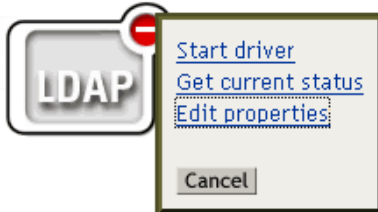
Reenter the password:

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

3.4.2 Configuring an Existing Driver

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

- 1 In Novell 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, enter parameters for the Remote Loader.

Authentication

S3K-NDS.Vmp

Authentication ID:	<input type="text" value="cn=Directory Manager"/>
Authentication context:	<input type="text" value="122.0.0.1:389"/>
Remote loader connection parameters:	<input type="text" value="192.168.0.1. port=8090 kmo='remote'"/>
Driver cache limit (kilobytes):	<input type="text" value="0"/>
Enter the application password:	<input type="password"/>
Reenter the application password:	<input type="password"/>
Enter the remote loader password:	<input type="password"/>
Reenter the remote loader password:	<input type="password"/>

- ◆ Remote Loader Connection Parameters

Earlier, you exported the self-signed certificate. (See [Section 3.2.2, “Exporting a Self-Signed Certificate,” on page 48.](#)) 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.

TIP: The KMO object name is the nickname value you specified in Step 2 of [Section 3.2.1, “Creating a Server Certificate,” on page 48.](#)

- ◆ 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.

- ◆ 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.

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

6 Click *OK*.

3.4.3 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 71](#)
- ◆ [“Keystore on Solaris, Linux, or AIX” on page 71](#)
- ◆ [“Keystore on All Platforms” on page 71](#)

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
```

Filename can be any name (for example, `rdev_keystore`).

Creating Policies

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 *Policy Builder and Driver Customization Guide*. This guide contains:

- ◆ A detailed description of each available policy
- ◆ An in-depth Policy Builder user guide and reference, including examples and syntax for each condition, action, noun, and verb.
- ◆ A discussion on creating policies using XSLT style sheets.

Please refer to the *Policy Builder and Driver Customization Guide* for information on policies.

Password Synchronization across Connected Systems

- ◆ [Section 5.1, “Overview,” on page 75](#)
- ◆ [Section 5.2, “Connected System Support for Password Synchronization,” on page 84](#)
- ◆ [Section 5.3, “Prerequisites for Password Synchronization,” on page 87](#)
- ◆ [Section 5.4, “Preparing to Use Identity Manager Password Synchronization and Universal Password,” on page 95](#)
- ◆ [Section 5.5, “Configuring and Synchronizing a New Driver,” on page 98](#)
- ◆ [Section 5.6, “Upgrading Password Synchronization 1.0,” on page 100](#)
- ◆ [Section 5.7, “Upgrading Existing Driver Configurations to Support Password Synchronization,” on page 100](#)
- ◆ [Section 5.8, “Implementing Password Synchronization,” on page 108](#)
- ◆ [Section 5.9, “Setting Up Password Filters,” on page 139](#)
- ◆ [Section 5.10, “Managing Password Synchronization,” on page 140](#)
- ◆ [Section 5.11, “Checking the Password Synchronization Status for a User,” on page 142](#)
- ◆ [Section 5.12, “Configuring E-Mail Notification,” on page 143](#)
- ◆ [Section 5.13, “Troubleshooting Password Synchronization,” on page 155](#)

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.

- ◆ [“Overview of Passwords” on page 75](#)
- ◆ [“Comparison of Password Synchronization 1.0 and Identity Manager Password Synchronization” on page 77](#)
- ◆ [“What is Bidirectional Password Synchronization?” on page 76](#)
- ◆ [“Features of Identity Manager Password Synchronization” on page 78](#)
- ◆ [“Overview Illustrations of Password Synchronization Flow” on page 81](#)

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 bi-directional 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 7, “Security: Best Practices,” on page 195.](#))

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 108.](#)

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 84.

5.1.3 Comparison of Password Synchronization 1.0 and Identity Manager Password Synchronization

Table 5-1 Comparison: 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	<ul style="list-style-type: none"> ◆ Active Directory ◆ NT Domain ◆ eDirectory 	<p>Full bidirectional password synchronization is supported on these platforms:</p> <ul style="list-style-type: none"> ◆ Active Directory ◆ eDirectory ◆ NIS ◆ NT Domain <p>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.</p> <p>Any connected system that supports the Subscriber password element can subscribe to passwords from Identity Manager.</p> <p>See Section 5.2, "Connected System Support for Password Synchronization," on page 84.</p>
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 108.

	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 100.
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 78](#)
- ◆ [“Distributing Passwords to Connected Systems” on page 79](#)
- ◆ [“Enforcing Password Policies in the Data Store and on Connected Systems” on page 79](#)
- ◆ [“Scenarios for Synchronizing Passwords” on page 80](#)
- ◆ [“Notifying Users of Password Synchronization Failures” on page 80](#)
- ◆ [“Checking the Password Synchronization Status for a User” on page 81](#)

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. It can be distributed.

Identity Manager can accept a password from a connected system. Because Universal Password is reversible, Identity Manager can distribute the password from the Identity Vault to connected systems that support setting initial passwords 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 108](#).

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 itself 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 108. 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** section 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, to send a copy to the administrator

For more information, see [“Configuring E-Mail Notification” on page 143](#).

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.

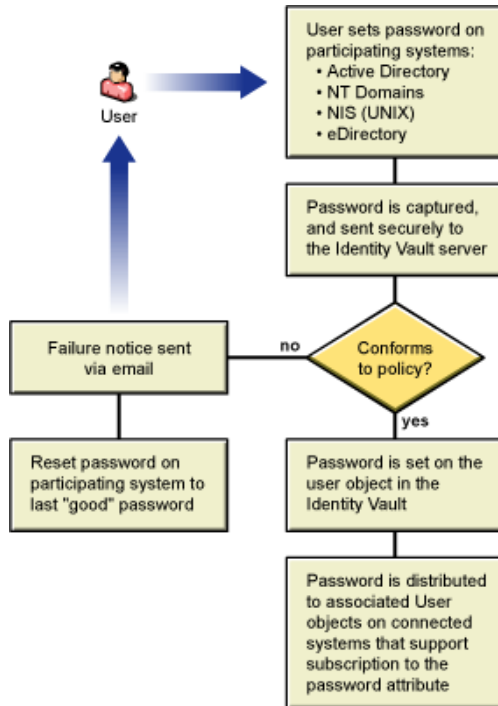
For information on how to check passwords, see [“Checking the Password Synchronization Status for a User” on page 142](#).

For a list of which systems support checking passwords, see [“Connected System Support for Password Synchronization” on page 84](#).

5.1.5 Overview Illustrations of Password Synchronization Flow

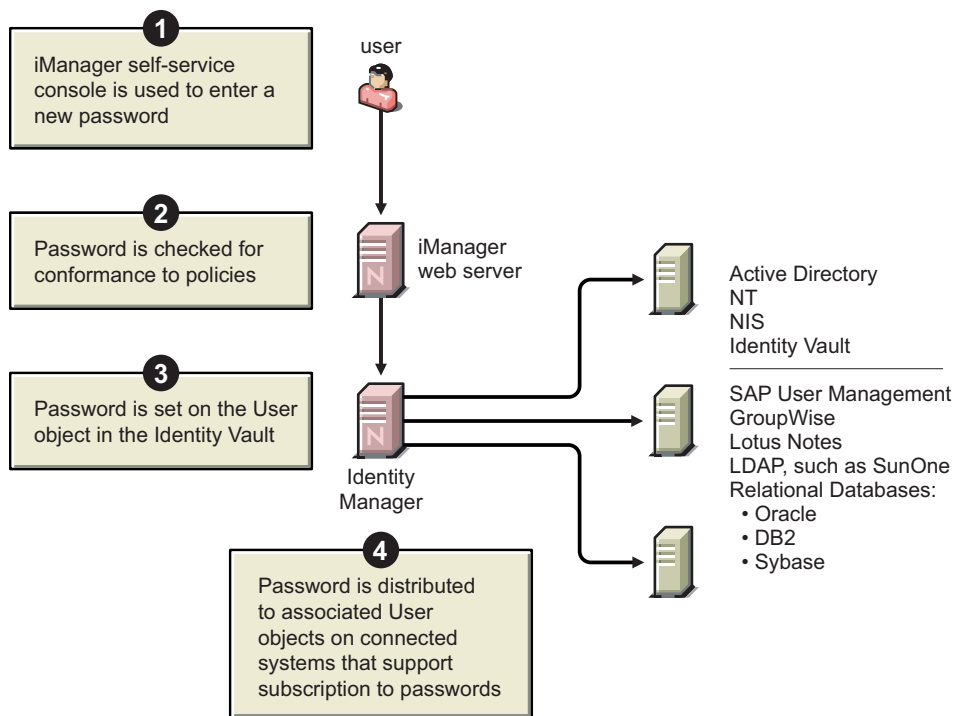
The following figure describes how connected systems publish passwords to Identity Manager.

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



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

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

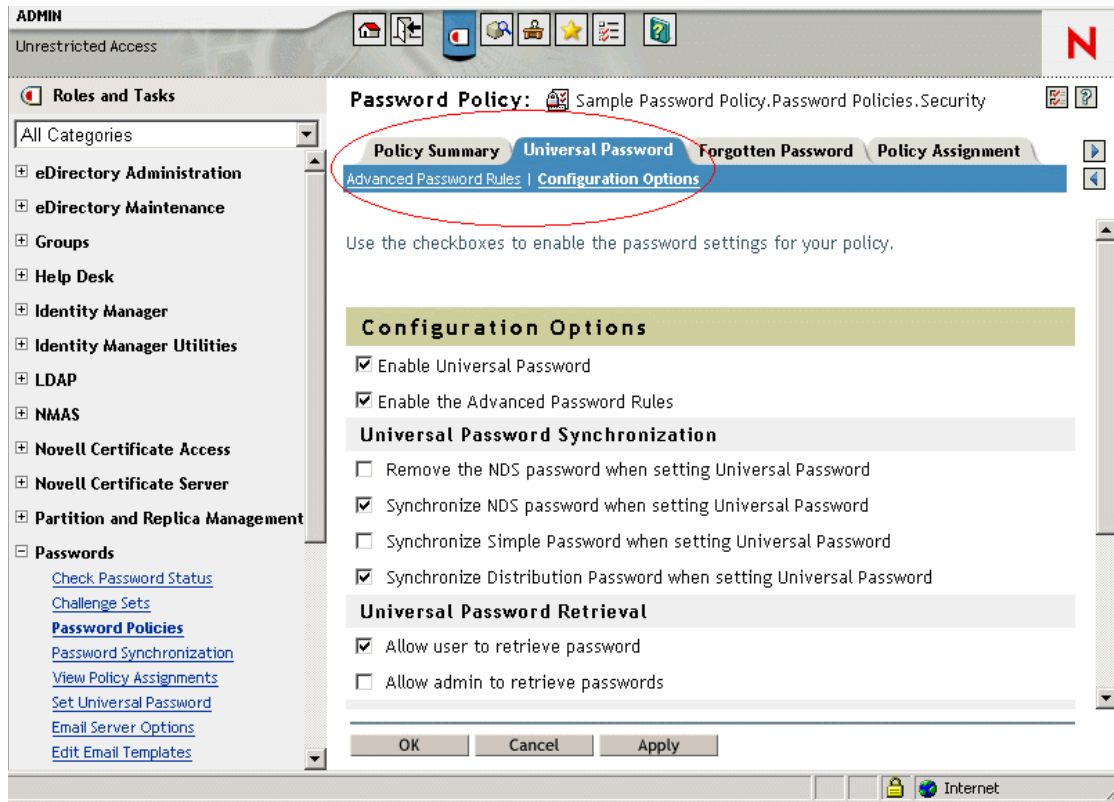


5.1.6 How Figures Display

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

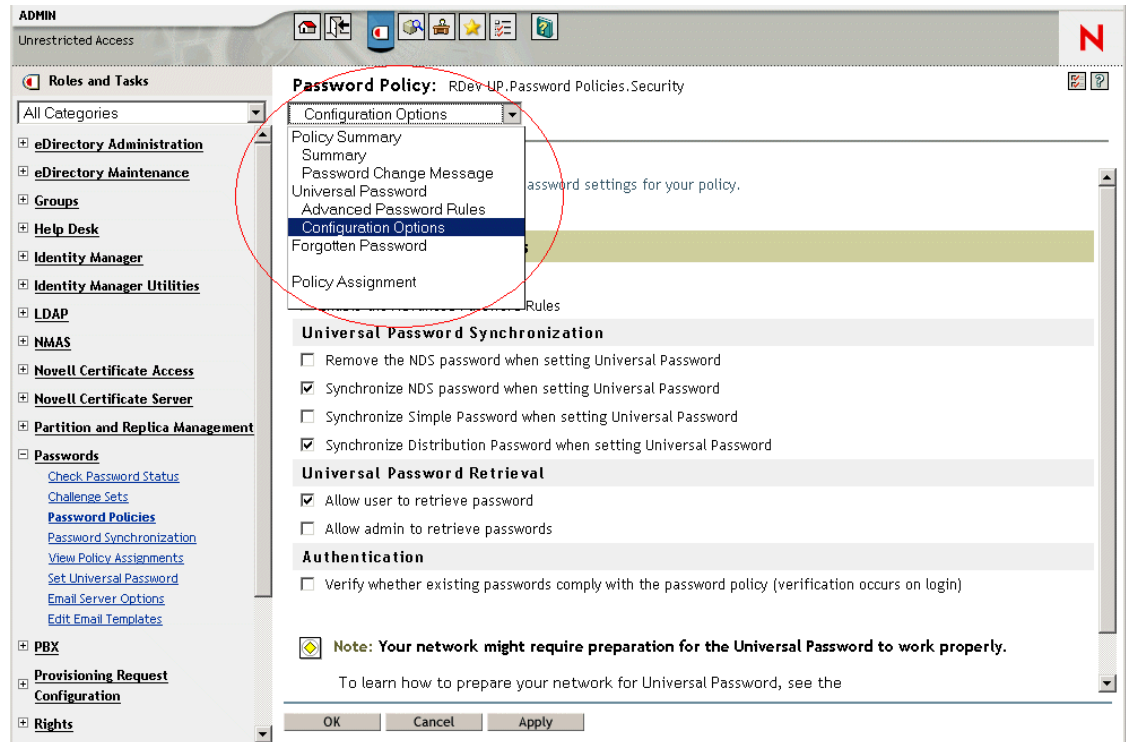
For example, Internet Explorer displays iManager options by using tabs.

Figure 5-3 Tabs in iManager



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

Figure 5-4 A Drop-Down List in iManager



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 set 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 set 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

¹Between 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 110.

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 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	No ⁴	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

²GroupWise 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.

³The 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*.

⁴The 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.

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

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

⁷The 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 using a policy on the Publisher channel, based on other user data in the connected system. (The sample driver configurations demonstrate 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
PeopleSoft 4.0	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
SAP HR	No	No	No	No

⁸The 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
LoopBack 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 88](#)
- ◆ [“Password Synchronization Capabilities Declared in the Driver Manifest” on page 88](#)
- ◆ [“Controlling Password Synchronization by Using Global Configuration Values” on page 88](#)
- ◆ [“Policies Required in the Driver Configuration” on page 91](#)
- ◆ [“Filters You Install on the Connected System to Capture Passwords” on page 95](#)
- ◆ [“NMA Password Policies You Create for Users” on page 95](#)
- ◆ [“NMA Login Methods” on page 95](#)

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 97](#)

5.3.2 Password Synchronization Capabilities Declared 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 need to 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 100](#).

5.3.3 Controlling Password Synchronization by Using Global Configuration Values

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 for each connected system separately.

Table 5-6 *Settings for Connected Systems*

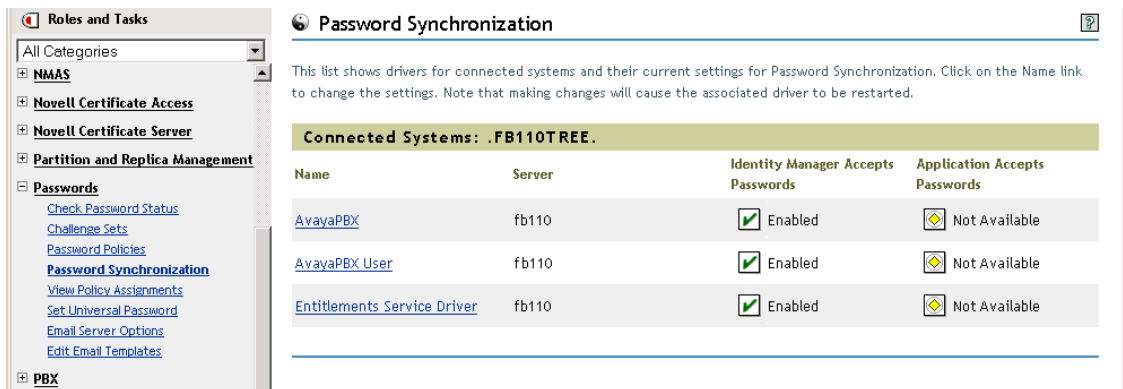
Setting	Description
Whether Identity Manager accepts passwords from the connected system	This setting applies to a password provided by the connected system, as well as a password that could be created by Identity Manager policies in the driver configuration on the Publisher channel. If you disable this setting, both kinds of passwords are stripped out so that they don't reach Identity Manager.
Which method of synchronization Identity Manager uses: updating Universal Password directly, or updating 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: <ol style="list-style-type: none"> 1. In iManager, select <i>Passwords > Password Policies</i>. 2. Select a policy in the <i>Password Policy List</i>. 3. Click <i>Edit</i>. 4. Select an option from the drop-down list or tab (depending on which version of iManager you are using). <p>See Section 5.8, "Implementing Password Synchronization" for scenarios that use these methods.</p>
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. <p>This option is dimmed in the interface if the connected system doesn't support it (as declared in the driver manifest).</p>
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 100](#).

To edit global configuration values:

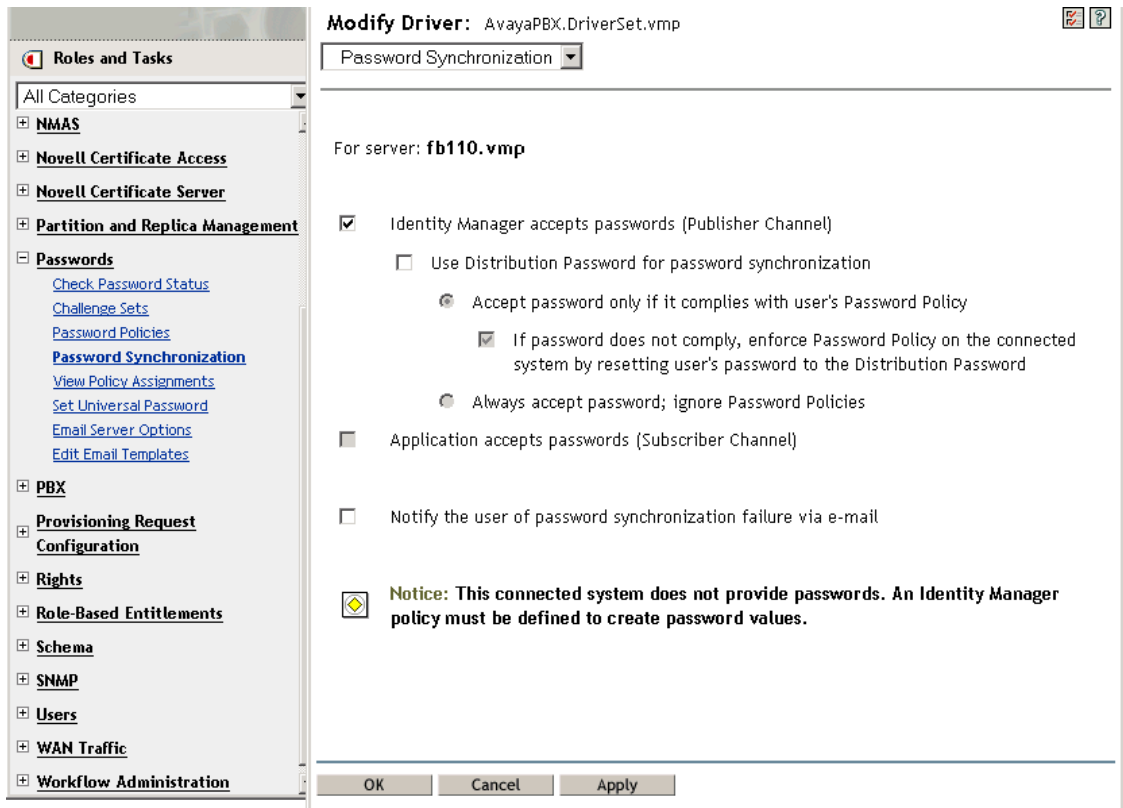
- 1 In iManager, select *Passwords > Password Synchronization*.
- 2 Search for a driver.

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.



3 To view settings, click a driver name.

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



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 100.](#)) 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 91](#)
- ◆ [“Policies Required in the Publisher Input Transformation Policy Set” on page 93](#)
- ◆ [“Policies Required in the Subscriber Command Transformation Policy Set” on page 93](#)
- ◆ [“Policies Required in the Subscriber Output Transformation Policy Set” on page 94](#)

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	<p>Adds a default password to an Add object if the Add object does not already contain a password.</p> <p>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.</p>
	Password(Pub)-Check Password GCV	<p>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.</p> <p>The name of the GCV is enable-password-publish, and the display name is <i>Identity Manager accepts passwords from application</i>.</p>
	Password(Pub)-Publish Distribution Password	<p>Transforms the <password> element to the form that allows it to update Universal Password.</p> <p>This policy references the following GCVs:</p> <ul style="list-style-type: none"> ◆ publish-password-to-dp ◆ enforce-password-policy
	Password(Pub)-Publish NDS Password	<p>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.</p> <p>This policy references the GCV named publish-password-to-nds.</p>
	Password(Pub)-Add Password Payload	<p>Puts in payload data that is passed around in the engine for purposes of e-mail notification.</p>
	Password(Sub)-Add Password Payload	<p>Puts in payload data that is passed around in the engine for purposes of e-mail notification.</p>

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	<p>If the password payload information comes through, and the status shows a problem, it sends e-mail to the user. It sends the mail to the user's e-mail address indicated in the Internet EMail Address attribute in eDirectory.</p> <p>This policy references the GCV named notify-user-on-password-dist-failure to determine whether to send notification e-mails.</p>

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 <i>Application accepts passwords from Identity Manager data store</i> .
	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 139](#).

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 96](#)
- ◆ [“Helping Users Change Passwords” on page 96](#)
- ◆ [“Preparing to Use Universal Password” on page 97](#)
- ◆ [“Matching Containers” on page 98](#)
- ◆ [“Setting Up E-Mail Notification” on page 98](#)

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 user needs the Universal Password to 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 143](#).

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 95](#).

- 2 Create a driver, or replace an existing driver's configuration with the Identity Manager configuration.

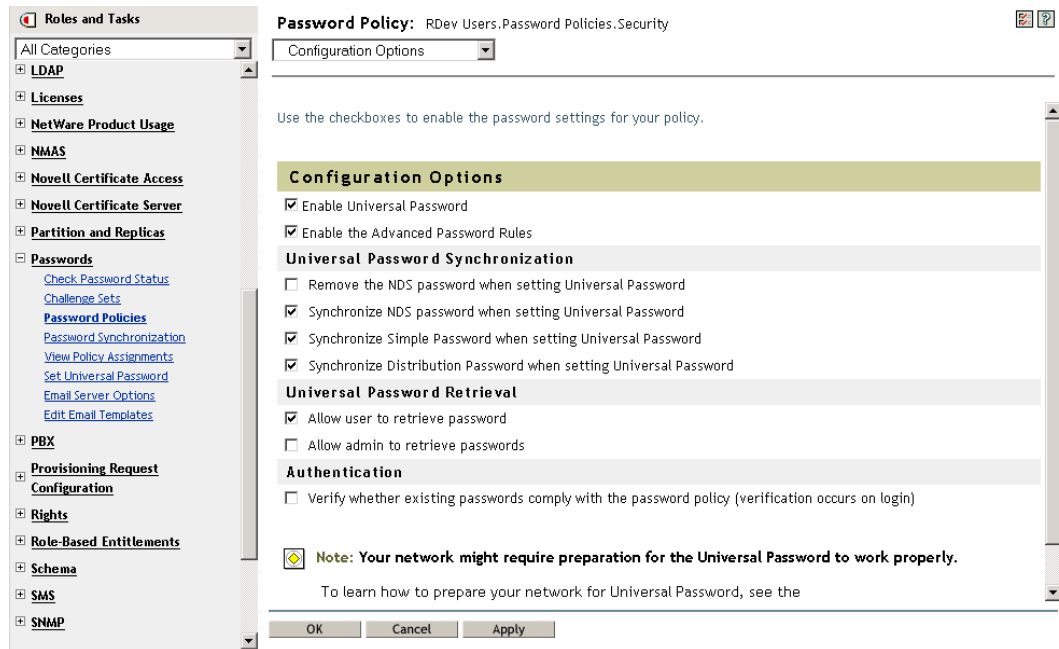
The Identity Manager configurations contain the Identity Manager policies and other items necessary for Identity Manager Password Synchronization. See the individual [Identity Manager Driver Guides](#) (<http://www.novell.com/documentation/beta/dirxml/drivers>) for information on importing the new sample driver configurations.

- 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.



For scenarios on using Password Synchronization, and how Identity Manager password policies fit in, see [Section 5.8, “Implementing Password Synchronization,” on page 108](#). Also see the online help.

- (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/ig/dirxml/drivers/index.html) (<http://www.novell.com/documentation/ig/dirxml/drivers/index.html>).

- For each connected system, make sure that password flow is set the way you want.
 - In iManager, click *Passwords > Password Synchronization*, and search for the drivers for connected systems that you want to manage.
 - 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 which password Identity Manager updates. NMASS 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 99](#).

- ◆ 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 108](#).

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 \(http://www.novell.com/documentation/dirxml/drivers/index.html\)](http://www.novell.com/documentation/dirxml/drivers/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 \(http://www.novell.com/documentation/dirxml/drivers/index.html\)](http://www.novell.com/documentation/dirxml/drivers/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 110.

- ◆ [“Step 1: Convert the Driver to Identity Manager 3.0.1 Format”](#) on page 101
- ◆ [“Step 2: Add to the Driver Configuration”](#) on page 104
- ◆ [“Step 3: Change Filter Settings”](#) on page 105
- ◆ [“Step 4: Setting Up Password Synchronization Flow”](#) on page 107

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 \(http://www.novell.com/documentation/dirxml/drivers/index.html\)](http://www.novell.com/documentation/dirxml/drivers/index.html).

5.7.1 Step 1: Convert the Driver to Identity Manager 3.0.1 Format

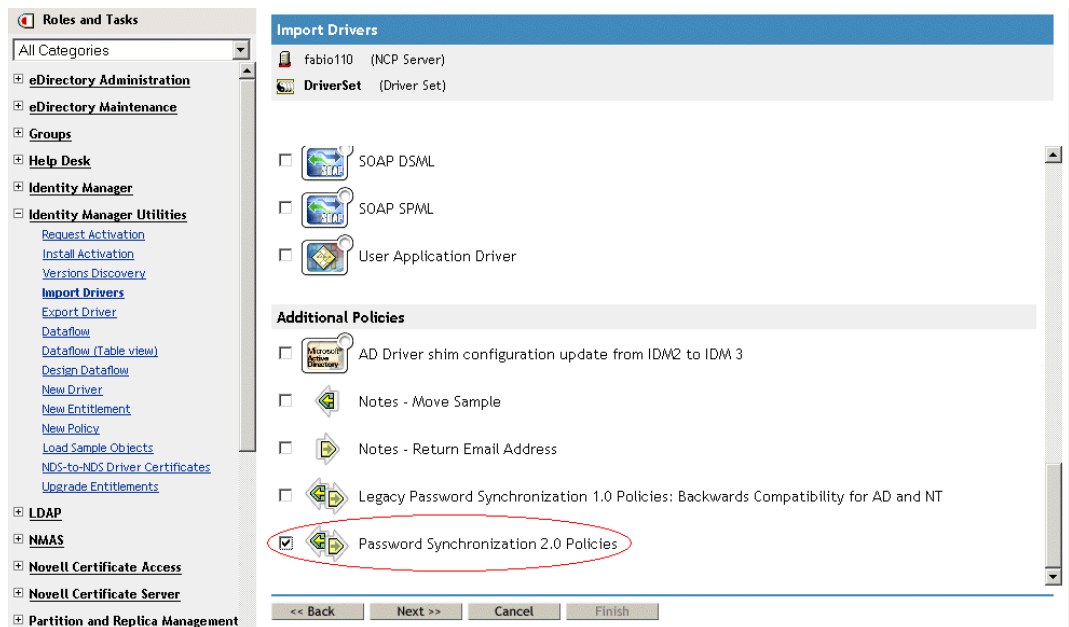
- 1 Make sure that your environment is ready to use Universal Password.

See [Section 5.4, “Preparing to Use Identity Manager Password Synchronization and Universal Password,”](#) on page 95.

If you are using DirXML[®] 1.1a, see [Section 2.3, “Upgrading a Driver Configuration from DirXML 1.1a to Identity Manager 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*.



- 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 usually should not 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 do 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: Add 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 91.

Insert each of the new policies into the correct place on your existing driver configuration.

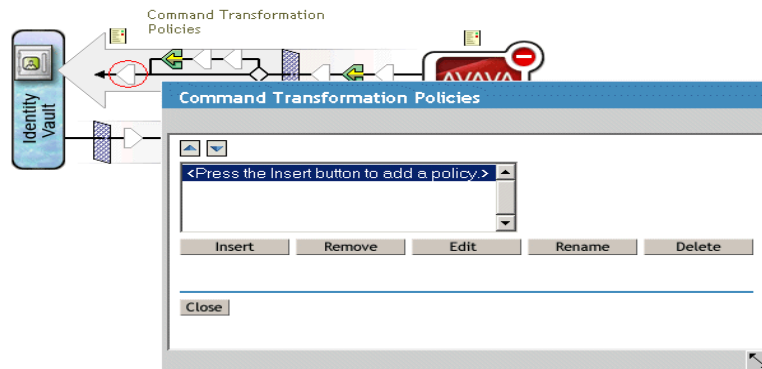
If the policy set has multiple policies, make sure these Identity Manager password synchronization policies are listed last.

Repeat the following steps for each policy.

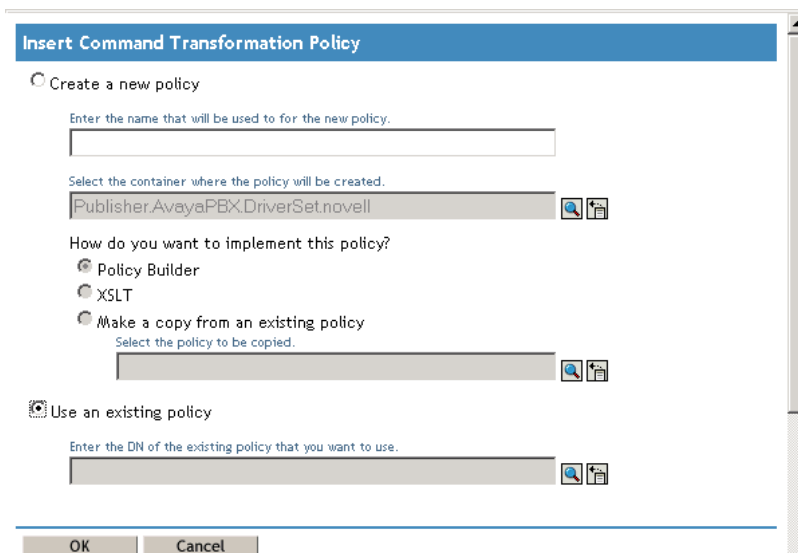
- 1 Select *Identity Manager > Identity Manager Overview*, then search for the driver set that contains the driver you are updating.
- 2 Click the driver that you just updated (for example, AvayaPBX).
- 3 Click the icon (for example, Command Transformation Policies on the Publisher channel) for the place where you need to add one of the new policies.



Identity Manager Driver Overview

Driver: AvayaPBX.DriverSet.vmp



- 4 Click Insert to add the new policy.

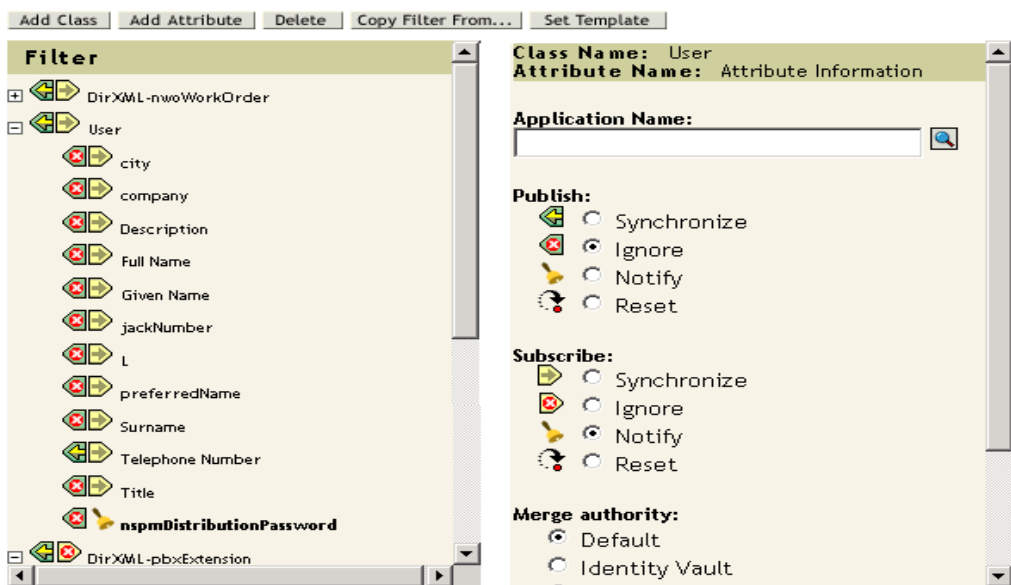


- 5 Click *Use an existing policy*, browse for the new policy object, then click *OK*.
- 6 If you have more than one policy in the list for any of the new policies, use the arrow buttons   to move the new policies to the correct location in the list.

Make sure that the policies are in the order listed in [Section 5.3.4, “Policies Required in the Driver Configuration,”](#) on page 91.

5.7.3 Step 3: Change 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.

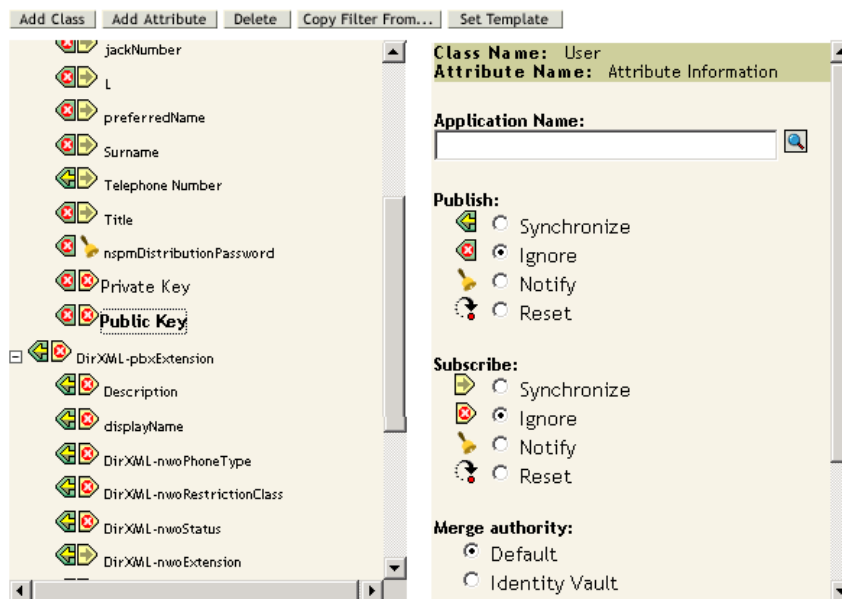


To view the attribute, you might have to scroll to and select the class (for example, User), then scroll through the attributes.

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*.

- For all objects that have *Notify* set for the *nspmDistributionPassword* attribute, set both the Public Key and Private Key attributes to Ignore.



- For each driver that you want to upgrade to participate in password synchronization, repeat **Step 2 on page 101** (in “Converting the Driver to Identity Manager 3.0.1 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.

- Check the individual driver implementation guides for any additional steps or information on setting up Identity Manager Password Synchronization. See [Identity Manager Drivers \(http://www.novell.com/documentation/lg/dirxml/drivers/index.html\)](http://www.novell.com/documentation/lg/dirxml/drivers/index.html).
- 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 108**.

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.

Roles and Tasks

- All Categories
- NMAS
- Novell Certificate Access
- Novell Certificate Server
- Partition and Replica Management
- Passwords
 - [Check Password Status](#)
 - [Challenge Sets](#)
 - [Password Policies](#)
 - Password Synchronization**
 - [View Policy Assignments](#)
 - [Set Universal Password](#)
 - [Email Server Options](#)
 - [Edit Email Templates](#)
- PBX

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	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/> Not Available
AvayaPBX User	fb110	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/> Not Available
Entitlements Service Driver	fb110	<input checked="" type="checkbox"/> Enabled	<input type="checkbox"/> 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.

OK Cancel Apply

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 90 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 108](#). 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 108](#).

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 Relationship to NMAS,” on page 108](#)
- ◆ [Section 5.8.2, “Scenario 1: Using NDS Password to Synchronize between Two Identity Vaults,” on page 110](#)
- ◆ [Section 5.8.3, “Scenario 2: Synchronizing by Using Universal Password,” on page 112](#)
- ◆ [Section 5.8.4, “Scenario 3: Synchronizing an Identity Vault and Connected Systems, with Identity Manager Updating the Distribution Password,” on page 122](#)
- ◆ [Section 5.8.5, “Scenario 4: Tunneling—Synchronizing Connected Systems but not an Identity Vault, with Identity Manager Updating the Distribution Password,” on page 131](#)
- ◆ [“Scenario 5: Synchronizing Application Passwords to the Simple Password” on page 135](#)

5.8.1 Overview of Identity Manager Relationship to NMAS

- ◆ [“Utilities and NMAS” on page 108](#)
- ◆ [“Identity Manager and NMAS” on page 109](#)

Utilities and NMAS

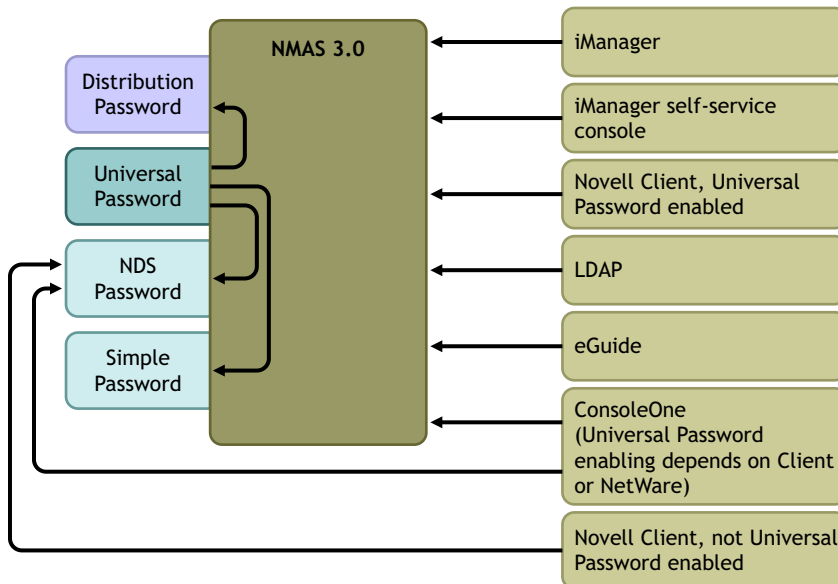
Utilities such as iManager and the Novell Client communicate with NMAS rather than directly updating a specific password. NMAS is the entity that determines which passwords are updated.

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



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 110
- ◆ “Setting Up Scenario 1” on page 111
- ◆ “Troubleshooting Scenario 1” on page 112

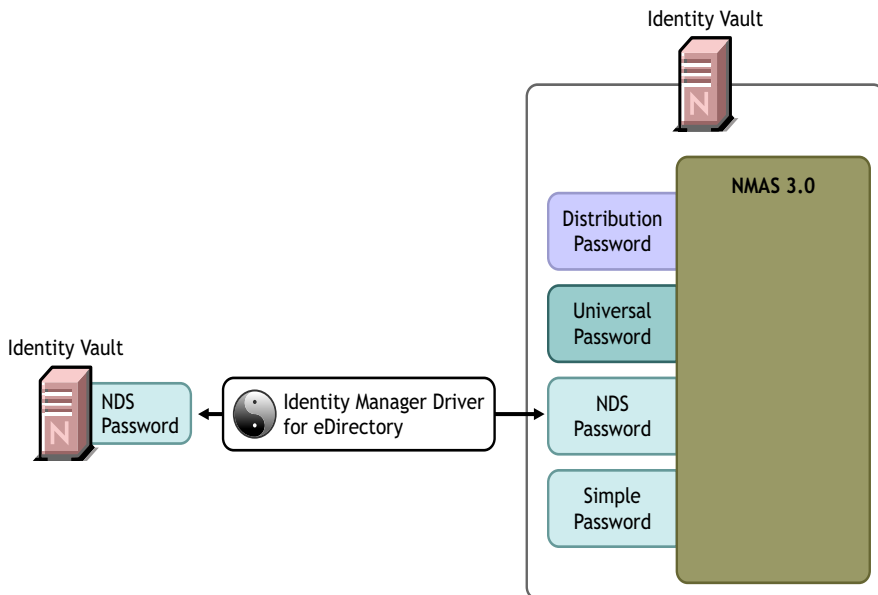
Advantages and Disadvantages of Scenario 1

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.	This method synchronizes passwords between Identity Vaults. Passwords cannot be synchronized to other connected systems.
If you are deploying Identity Manager 3.0.1 and eDirectory 8.7.3 in stages, this method can help you deploy gradually.	Does not update Universal Password or Distribution Password.
<ul style="list-style-type: none">◆ 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.	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.
Enforces the basic password restrictions you can set for NDS Password.	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

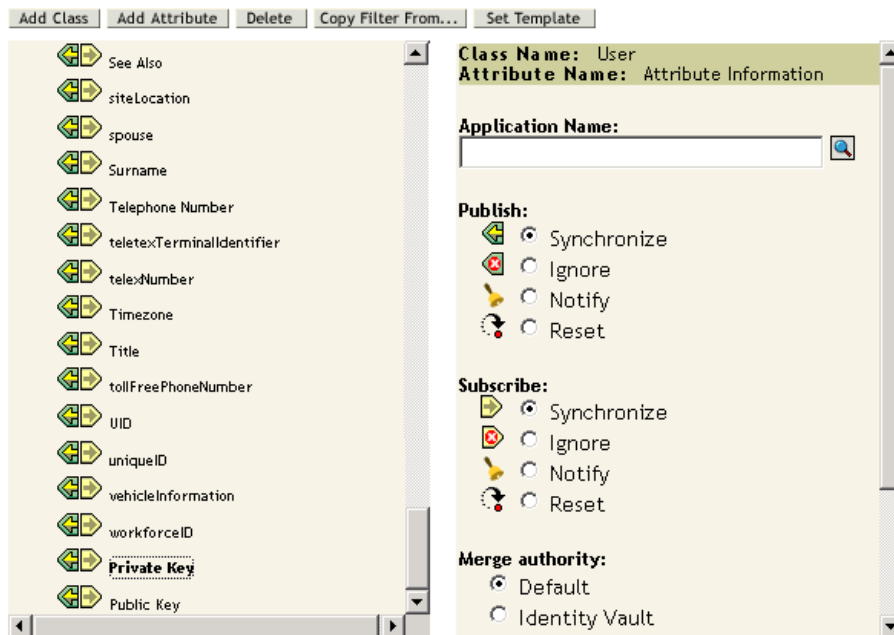
None. The settings on the Password Synchronization page for a driver have no effect on this method of synchronizing NDS Password.

Driver Configuration

Remove the Password Synchronization policies listed in [Section 5.3.4, “Policies Required in the Driver Configuration,” on page 91](#). 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



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 155.

5.8.3 Scenario 2: Synchronizing by Using Universal Password

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 84.

- ◆ [“Advantages and Disadvantages of Scenario 2”](#) on page 113
- ◆ [“Setting Up Scenario 2”](#) on page 114
- ◆ [“Troubleshooting Scenario 2”](#) on page 119

Advantages and Disadvantages of Scenario 2

Table 5-12 *Advantages: Synchronizing by Using Universal Password*

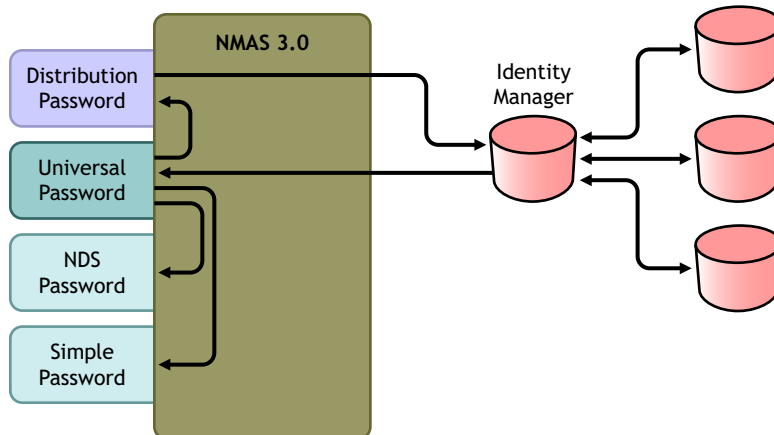
Advantages	Disadvantages
Allows synchronization of passwords to and from the Identity Vault and the connected system.	By design, resetting passwords in the connected system is not supported with this method because the Distribution Password and Universal passwords might not be the same, depending on your settings in the password policies.
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.	

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 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 114
- ◆ “[Password Policy Configuration](#)” on page 114
- ◆ “[Password Synchronization Settings](#)” on page 116
- ◆ “[Driver Configuration](#)” on page 117

Universal Password Deployment

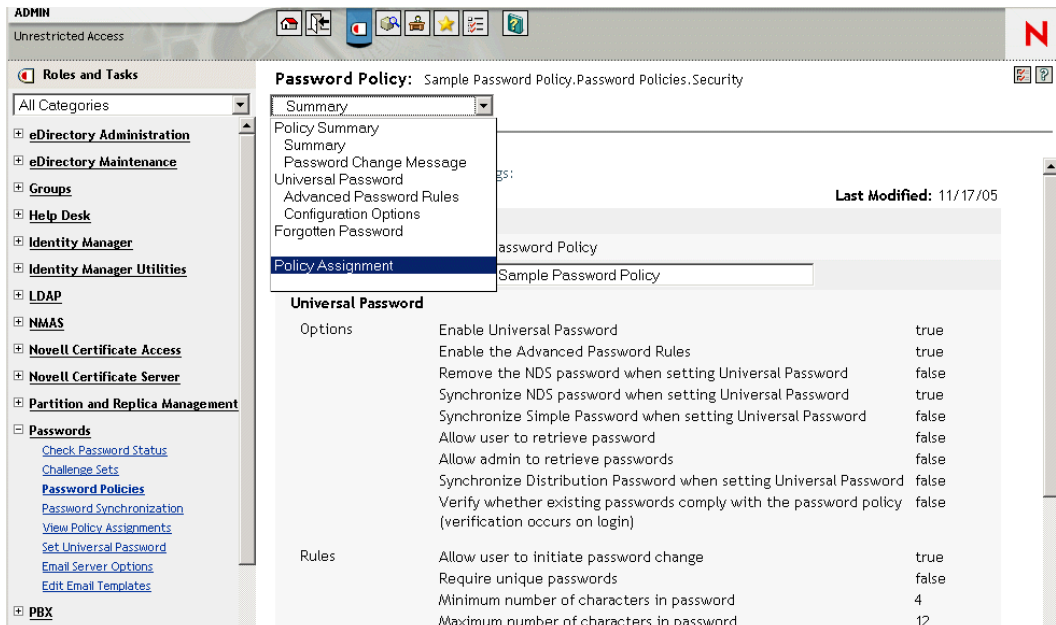
Make sure your environment is ready to use Universal Password. See [Section 5.4, “Preparing to Use Identity Manager Password Synchronization and Universal Password,”](#) on page 95.

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.

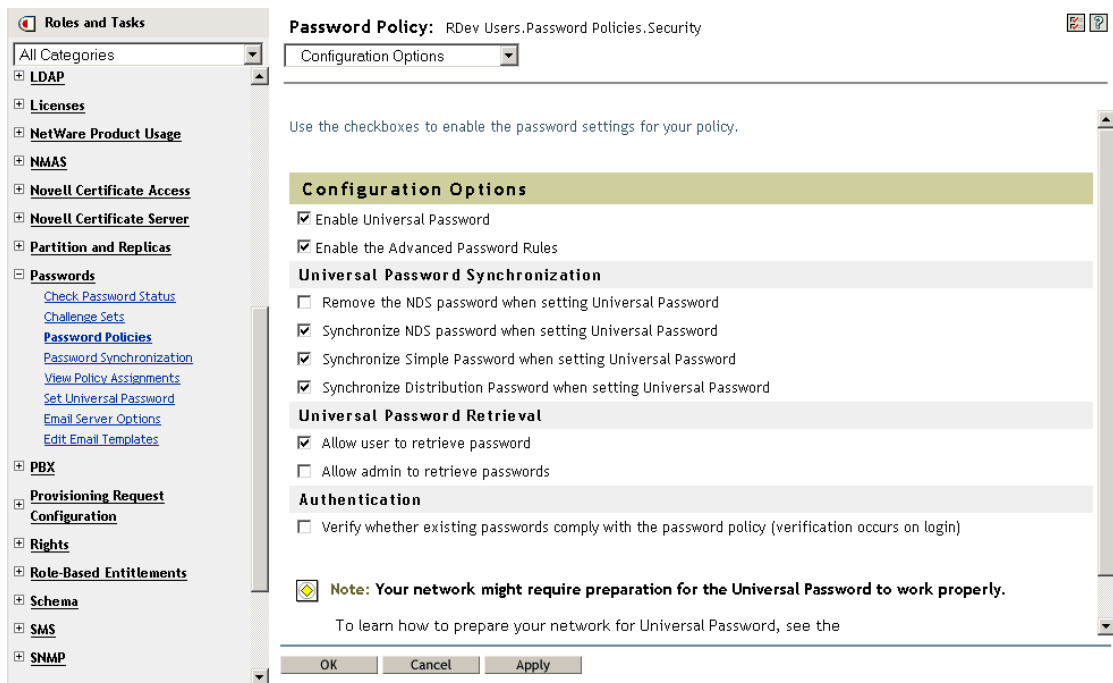
- 1 In iManager, select *Passwords > Password Policies*.
- 2 Select a policy, then click *Edit*.

3 Browse to and select the object where you want password synchronization to occur.



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:



- ◆ *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 checked 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 checked:

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.

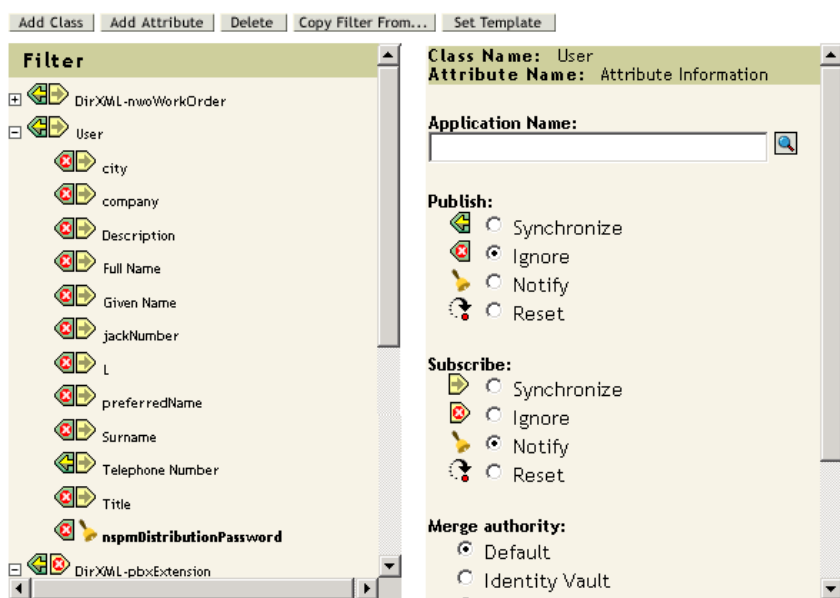
Driver Configuration

- 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.

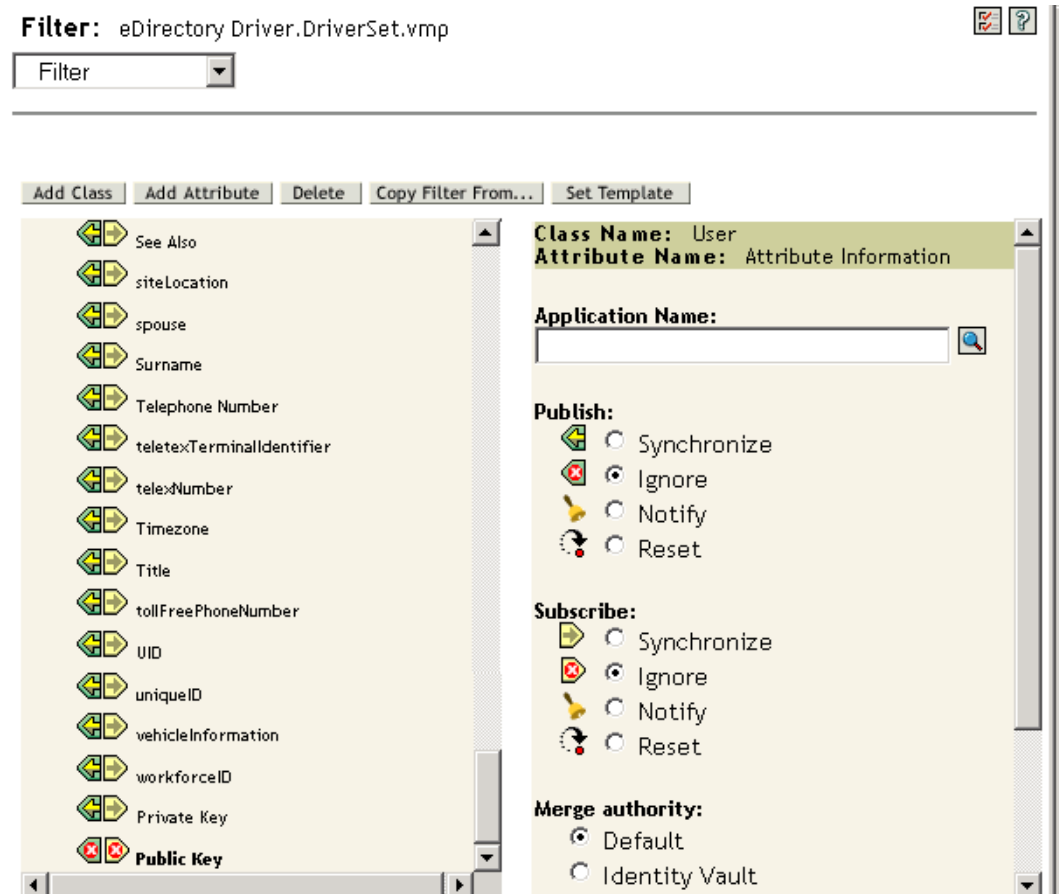
The policies must be in the correct location and the correct order in the driver configuration. For the list of policies, see [Section 5.3.4, “Policies Required in the Driver Configuration,” on page 91](#).

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 100](#).

- 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.



- 3 For all objects that have *Notify* set for the *nspmDistributionPassword* attribute, set both the Public Key and Private Key attributes to *Ignore*.



- 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 119
- ♦ “Trouble Logging in to the Identity Vault” on page 120
- ♦ “Trouble Logging in to Another Connected System that Subscribes to Passwords” on page 120
- ♦ “E-Mail Not Generated on Password Failure” on page 121
- ♦ “Error When Using Check the Object Password” on page 121
- ♦ “Helpful DTrace Commands” on page 121

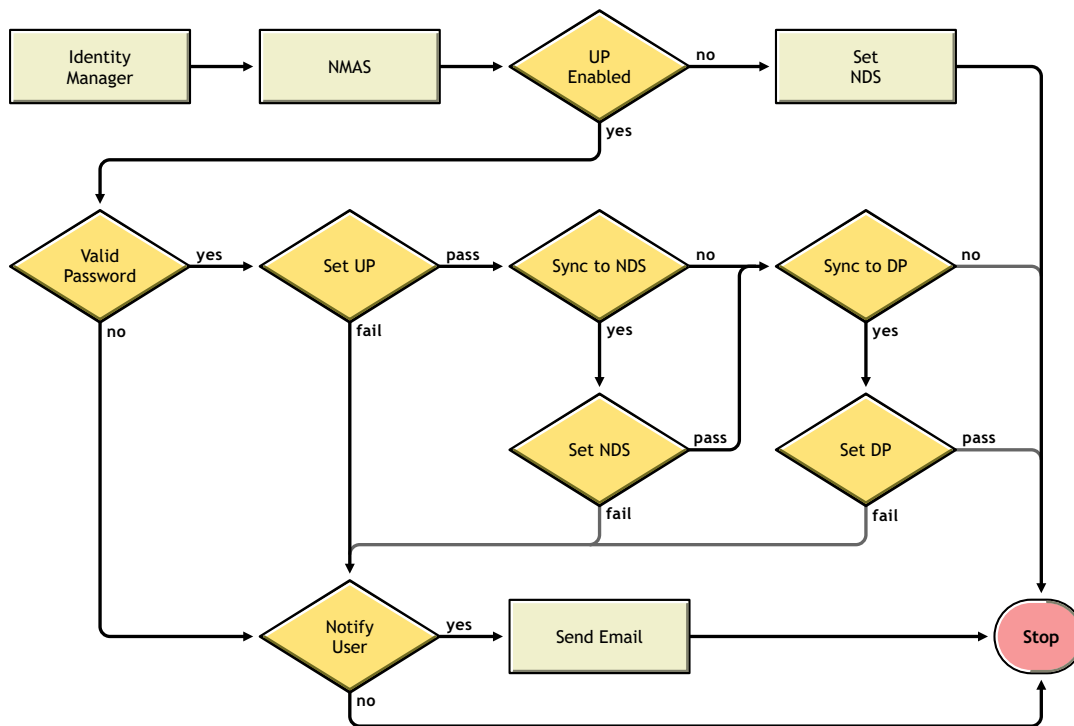
Also see the tips in Section 5.13, “Troubleshooting Password Synchronization,” on page 155.

Flowchart for Scenario 2

The following flowchart 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

```
151.156.165.10 - PTY
TAGS: Show Event Tags      TIME: Show Event Times   ABUF: Agent Buffers
ALOC: Memory Allocation   AREQ: Agent Requests     AUMN: Audit
AUMC: 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
OBUP: 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 `+DXML` 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 117](#).
- ◆ 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 91](#).
- ◆ 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 143](#).

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 DTrace 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 84](#).

- ◆ [“Advantages and Disadvantages of Scenario 3” on page 122](#)
- ◆ [“Setting Up Scenario 3” on page 123](#)
- ◆ [“Troubleshooting Scenario 3” on page 127](#)

Advantages and Disadvantages of Scenario 3

Table 5-13 *Advantages: Synchronizing an Identity Vault and Connected Systems by Updating the Distribution Password*

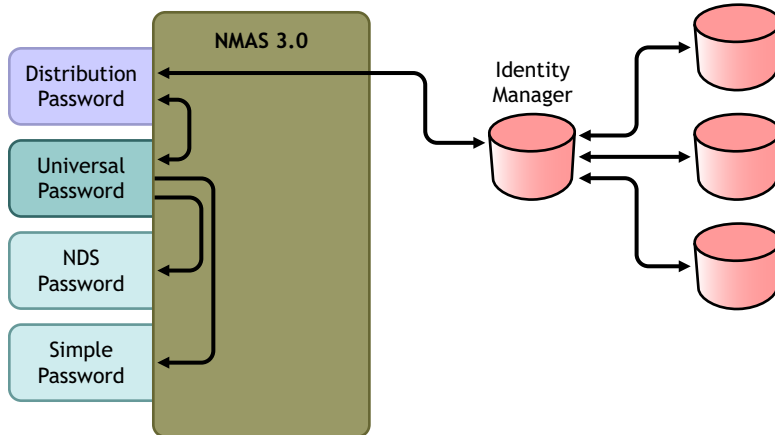
Advantages	Disadvantages
Allows synchronization of passwords between the Identity Vault and connected systems.	
Lets you choose whether or not to enforce password policies for passwords coming from connected systems.	
You can specify that notification be sent if password synchronization fails.	
If you 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 this figure, 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 123
- ◆ “[Password Policy Configuration](#)” on page 123
- ◆ “[Password Synchronization Settings](#)” on page 124
- ◆ “[Driver Configuration](#)” on page 126

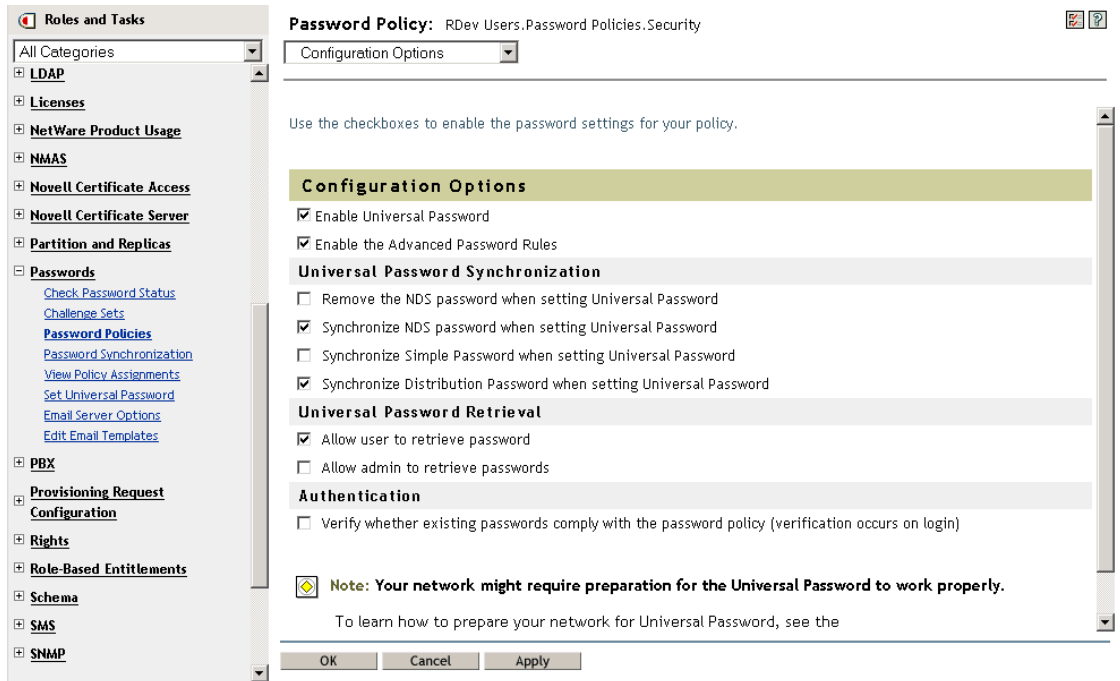
Universal Password Deployment

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 95.

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:



- ◆ *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.wmp  

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.

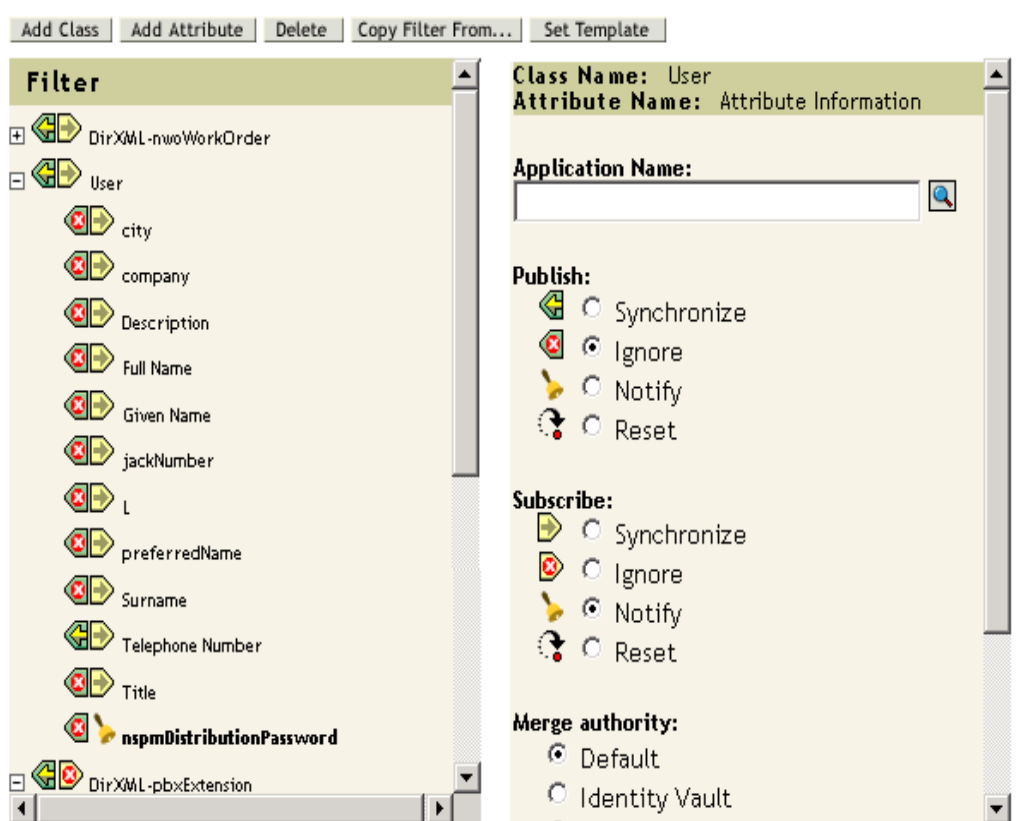
Driver Configuration

- 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.

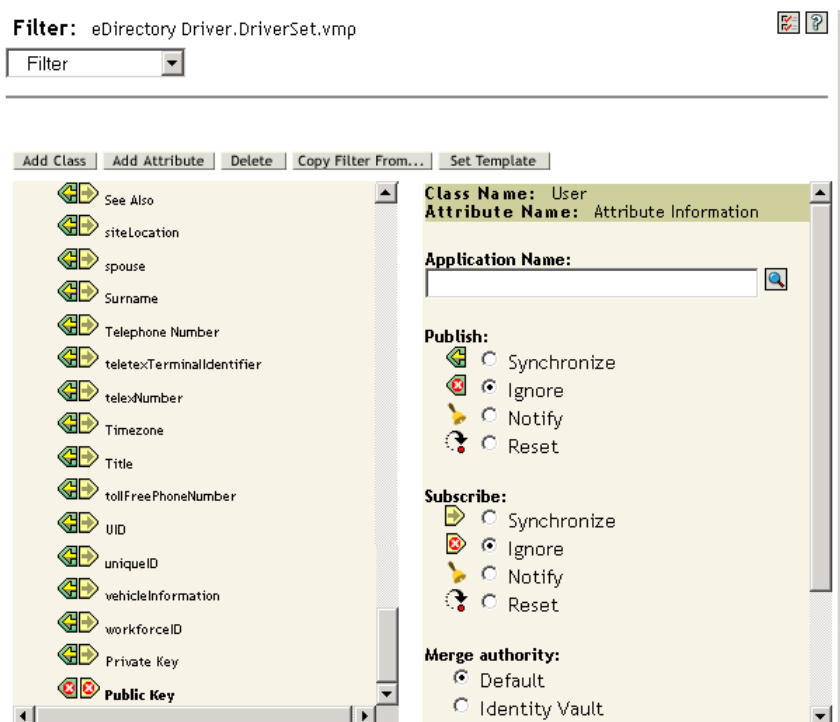
The policies must be in the correct location and the correct order in the driver configuration. For the list of policies, see [Section 5.3.4, “Policies Required in the Driver Configuration,”](#) on page 91.

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 100.

- 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.



- 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*.



- 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 127](#)
- ◆ [“Trouble Logging in to eDirectory” on page 128](#)
- ◆ [“Trouble Logging in to Another Connected System that Subscribes to Passwords” on page 129](#)
- ◆ [“E-Mail Not Generated on Password Failure” on page 130](#)
- ◆ [“Error When Using Check Password Status” on page 130](#)
- ◆ [“Helpful DTrace Commands” on page 130](#)

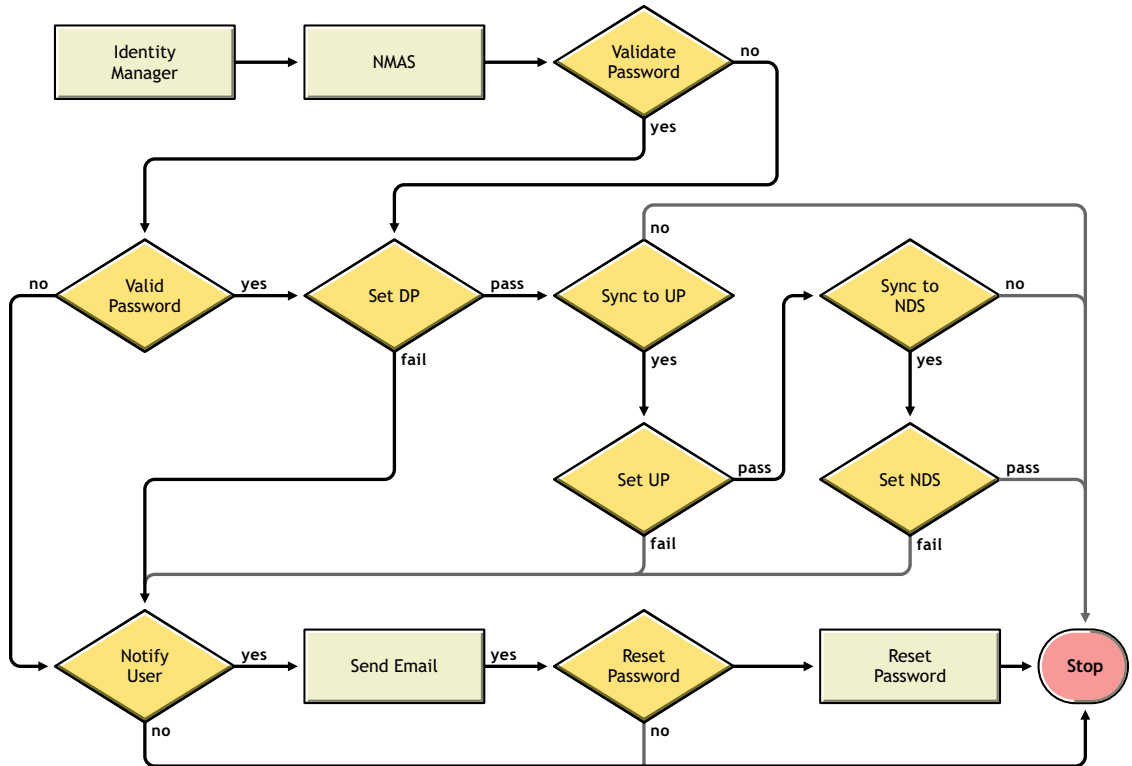
Also see the tips in [Section 5.13, “Troubleshooting Password Synchronization,” on page 155](#).

Flowchart for Scenario 3

The following flowchart 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

```

151.156.165.10 - PTY
TAGS: Show Event Tags          TIME: Show Event Times        ABUF: Agent Buffers
ALOC: Memory Allocation        AREQ: Agent Requests         AUMN: Audit
AUMC: 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
SVNC: 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 \(http://www.novell.com/documentation/nmas30/index.html\)](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 91.
- ♦ 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 143.

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 84.
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—Synchronizing Connected Systems but not an Identity Vault, with Identity Manager Updating the Distribution Password

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 122](#). 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 131](#)
- ◆ [“Setting Up Scenario 4” on page 132](#)
- ◆ [“Troubleshooting Scenario 4” on page 134](#)

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.	If Universal Password or Advanced Password Rules are not enabled, password policies are not enforced, and passwords on connected systems cannot be reset.
Password policies are not required.	
If you are using a password policy, the policy does not need to have Universal Password enabled. However, 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.	

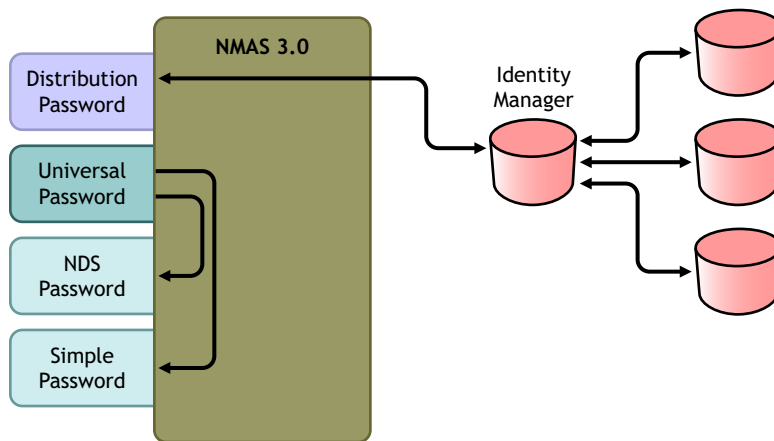
The figure in this scenario illustrates the following flow:

1. Passwords come in through Identity Manager.
2. Identity Manager goes through NMAS to directly update 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 132
- ◆ “Password Policy Configuration” on page 132
- ◆ “Password Synchronization Settings” on page 133
- ◆ “Driver Configuration” on page 133

Universal Password Deployment

Although you don't need to have password policies with Universal Password enabled, your environment must still 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 95.

Password Policy Configuration

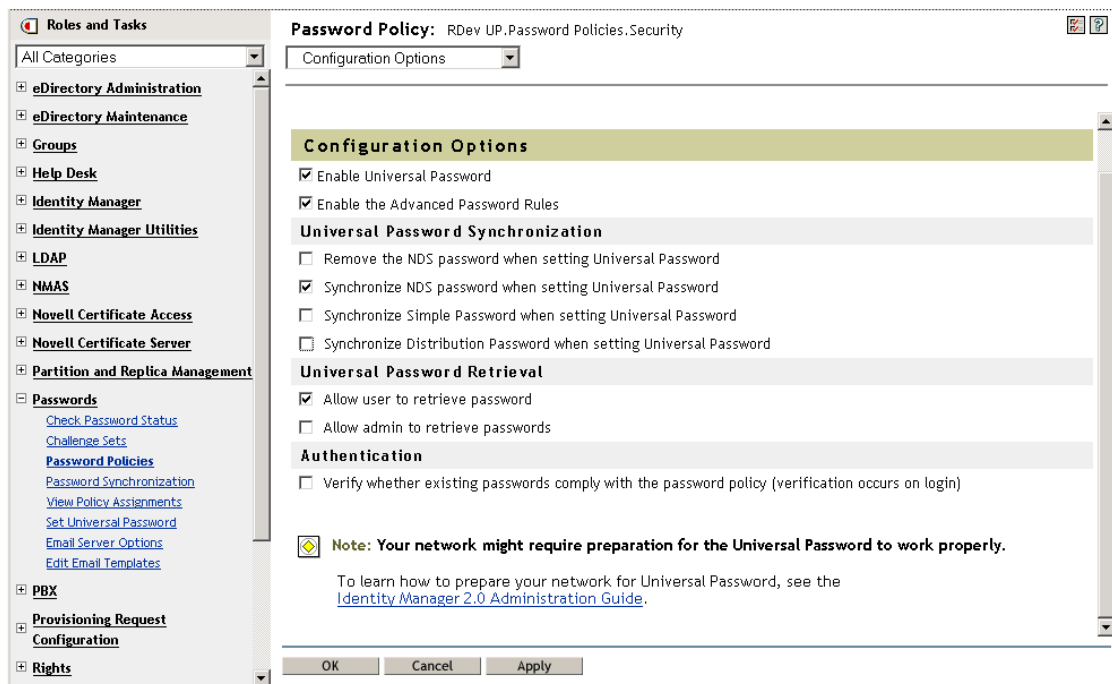
No password policy is required for Identity Vault users for this method.

However, if you use a password policy, you must do the following:

1 Make sure the following is not selected:

- ◆ *Synchronize Distribution Password when setting Universal Password*

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.



2 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 122.

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 122.

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 134
- ◆ “E-Mails Not Generated on Password Failure” on page 134
- ◆ “Error When Using Check Password Status” on page 135
- ◆ “Helpful DSTrace Commands” on page 135

See also the tips in [Section 5.13, “Troubleshooting Password Synchronization,”](#) on page 155.

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 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 91.
- ◆ 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 143](#).

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 84](#).

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 DTrace 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.

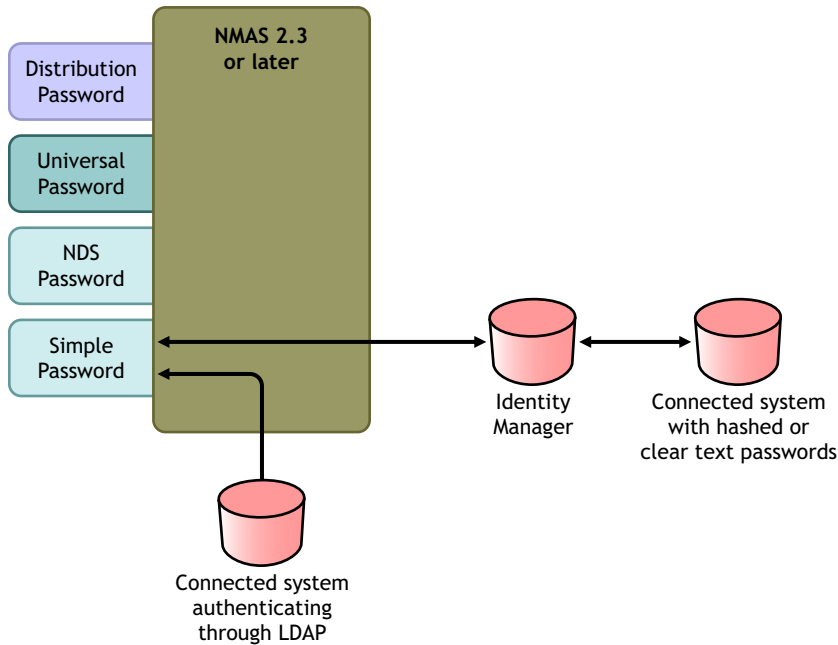
- ◆ [“Advantages of Synchronizing to the NDS Password” on page 136](#)
- ◆ [“Setting Up Scenario 5” on page 137](#)

Advantages of Synchronizing to the NDS Password

Table 5-15 *Advantages of Synchronizing to the NDS Password*

Advantages	Disadvantages
<ul style="list-style-type: none">◆ 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.	<ul style="list-style-type: none">◆ 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 137](#)
- ◆ [“Password Synchronization Settings” on page 137](#)
- ◆ [“Driver Configuration” on page 138](#)

Password Policy Configuration

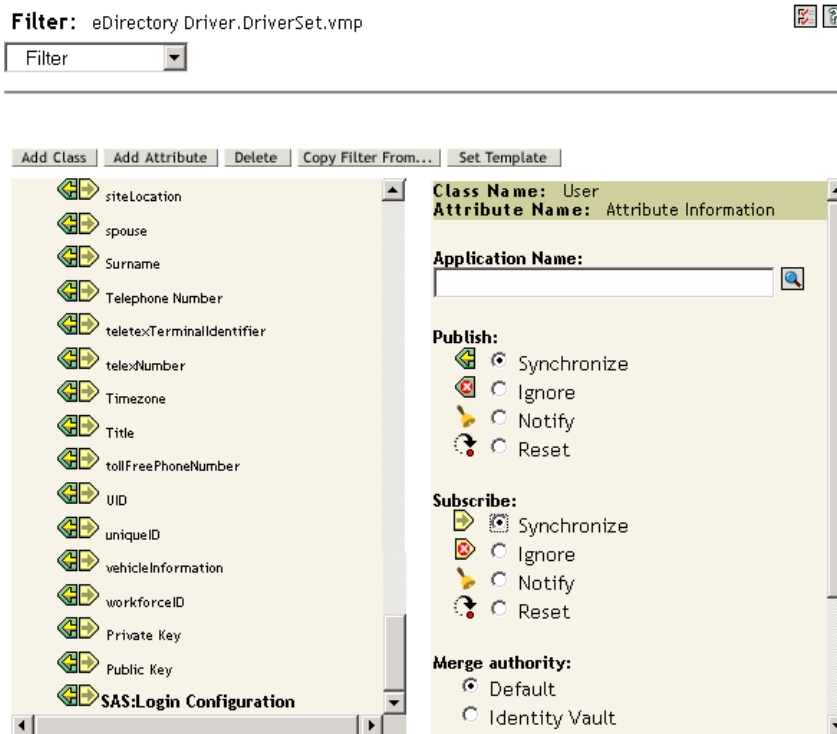
No password policy is required for users for this scenario. Universal Password cannot be used.

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.



- 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 Base 64 encoded.
- ♦ *{SHA}hashed_password*
This password is Base 64 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 139](#)
- [Section 5.9.2, “Setting Up Password Synchronization Filters for NIS,” on page 139](#)

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 \(http://www.novell.com/documentation/dirxml/drivers/index.html\)](http://www.novell.com/documentation/dirxml/drivers/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 \(http://www.novell.com/documentation/lg/dirxml/drivers/index.html\)](http://www.novell.com/documentation/lg/dirxml/drivers/index.html).

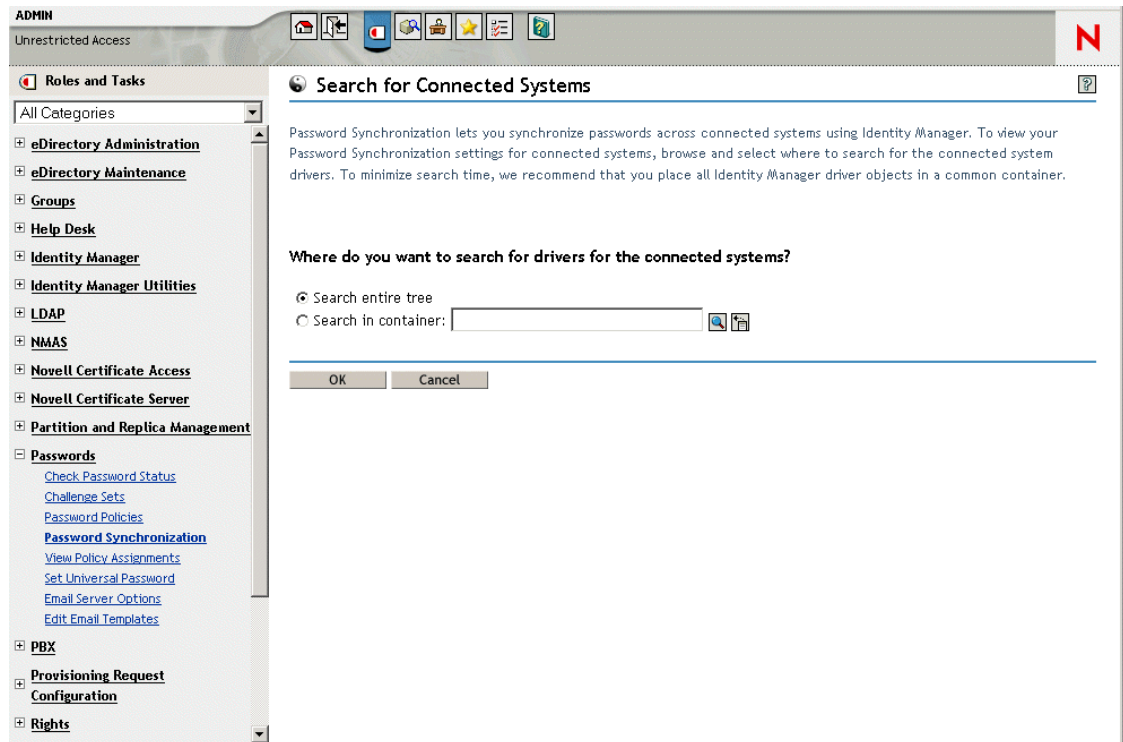
5.10 Managing Password Synchronization

- ♦ “Setting the Flow of Passwords Across Systems” on page 140
- ♦ “Enforcing Password Policies on Connected Systems” on page 141
- ♦ “Keeping the eDirectory Password Separate from the Synchronized Password” on page 141

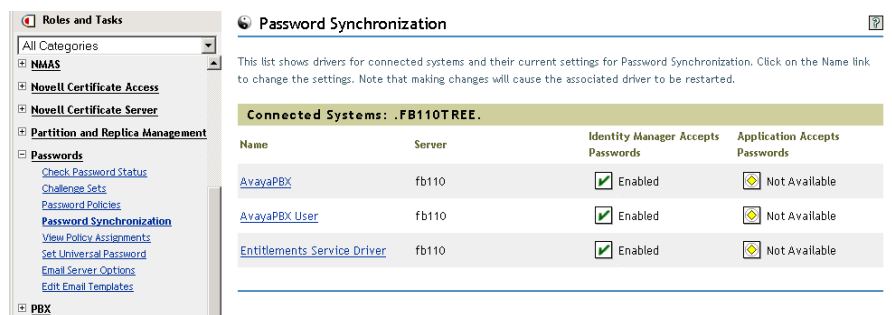
5.10.1 Setting the Flow of Passwords Across Systems

To view how your systems are set up to accept or publish passwords:

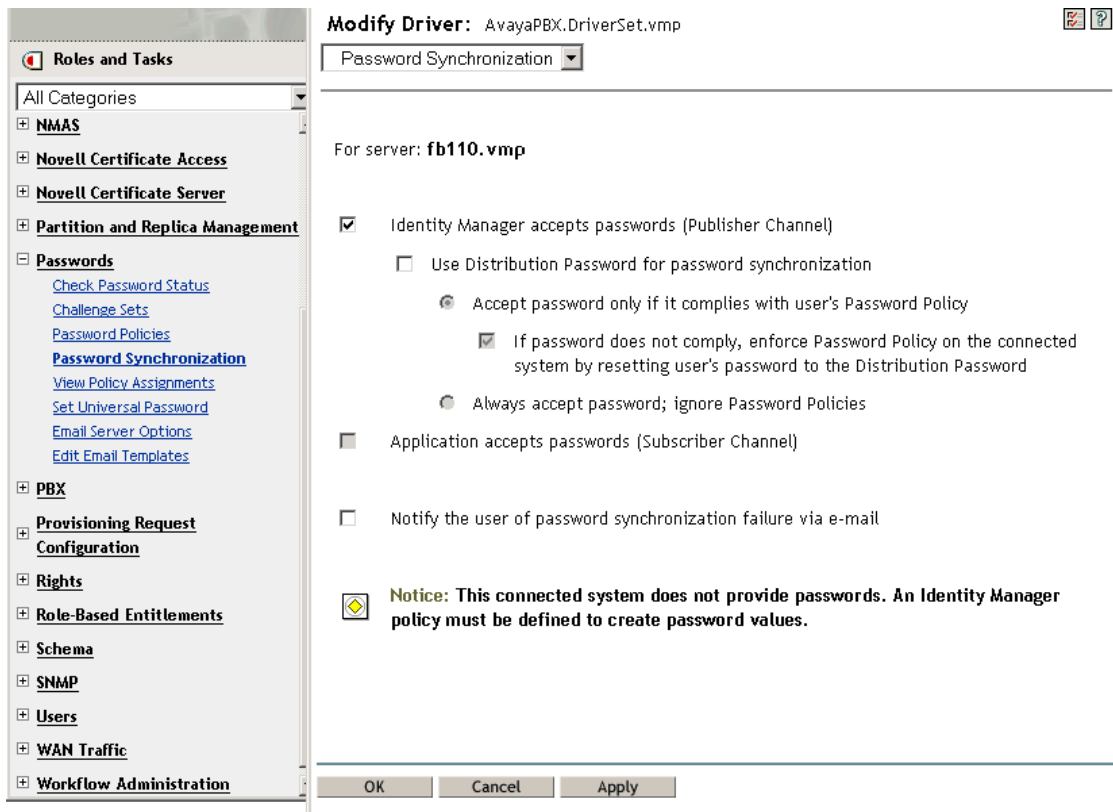
- 1 In iManager, select *Passwords > Password Synchronization*.
- 2 Search for drivers for the connected systems.



The search results show the settings for password flow to and from Identity Manager and the connected systems.



To make changes to these settings, click a connected system driver name.



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, “Controlling Password Synchronization by Using Global Configuration Values,” on page 88.](#)

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.

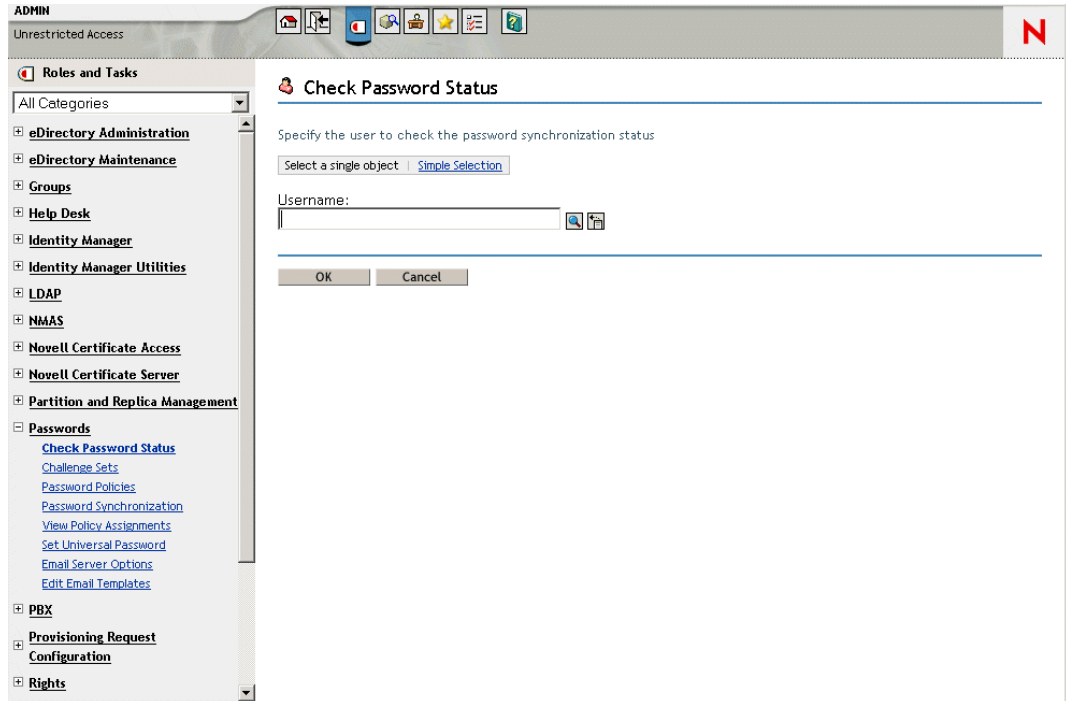
5.10.3 Keeping the eDirectory Password Separate from the Synchronized Password

This scenario is described in [Section 5.8.5, “Scenario 4: Tunneling—Synchronizing Connected Systems but not an Identity Vault, with Identity Manager Updating the Distribution Password,” on page 131.](#)

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*.



- 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 110.
- You are synchronizing Universal Password (as in [Section 5.8.3, “Scenario 2: Synchronizing by Using Universal Password,”](#) on page 112), 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.

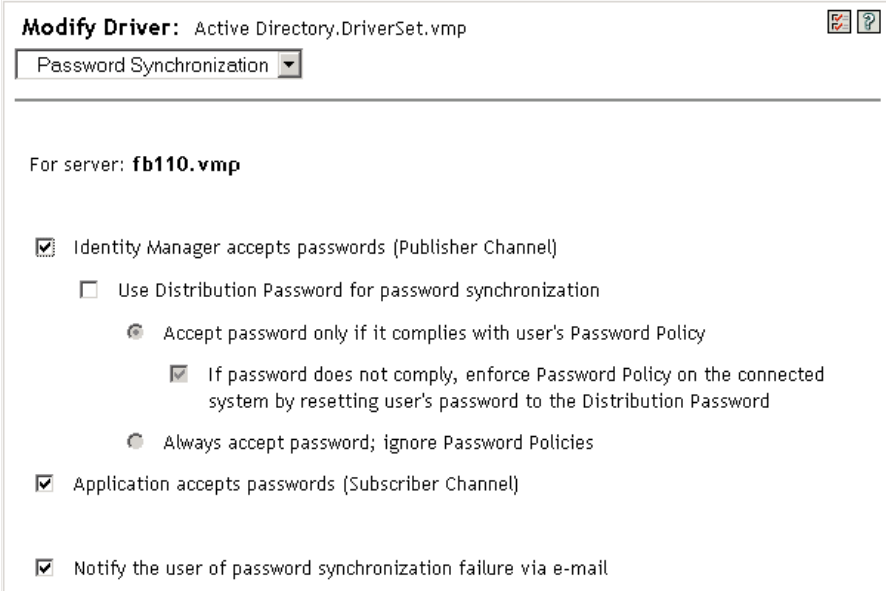
You don't create the templates. They are provided by the application that uses them. The e-mail templates are Template objects in the Identity Vault, and they are placed in the Security container, usually found at the root of your tree. Although they are Identity Vault objects, you should edit them only 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



The screenshot shows a dialog box titled "Modify Driver: Active Directory.DriverSet.vmp". At the top, there is a dropdown menu set to "Password Synchronization". Below this, it says "For server: fb110.vmp". The main area contains several checkboxes and radio buttons:

- 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 144](#)
- ◆ [Section 5.12.2, “Setting Up the SMTP Server To Send E-Mail Notification,” on page 145](#)
- ◆ [“Setting Up E-Mail Templates for Notification” on page 146](#)
- ◆ [Section 5.12.4, “Providing SMTP Authentication Information in Driver Policies,” on page 146](#)
- ◆ [Section 5.12.5, “Adding Your Own Replacement Tags to E-Mail Notification Templates,” on page 148](#)
- ◆ [Section 5.12.6, “Sending E-Mail Notifications to the Administrator,” on page 154](#)
- ◆ [Section 5.12.7, “Localizing E-Mail Notification Templates,” on page 154](#)

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 146.](#)
- ❑ If you are concerned that some users might not have the e-mail address populated, or 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 154.](#)

- ❑ 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 would need 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*.

The screenshot shows the iManager interface. On the left, the 'Roles and Tasks' pane is expanded to 'Passwords', where 'Email Server Options' is selected. The main window displays the 'Email Server Options' dialog box. It includes a title bar, a toolbar, and a main content area with the following elements:

- Host Name:** A text input field with a placeholder example: (for example: mail.novell.com or 137.89.119.5)
- From:** A text input field with a placeholder example: (for example: admin@novell.com)
- Authenticate to server using credentials:** A checkbox that is currently unchecked.
- User Name:** A text input field.
- Password:** A text input field.
- Retype password:** A text input field.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom.

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 146](#) 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.

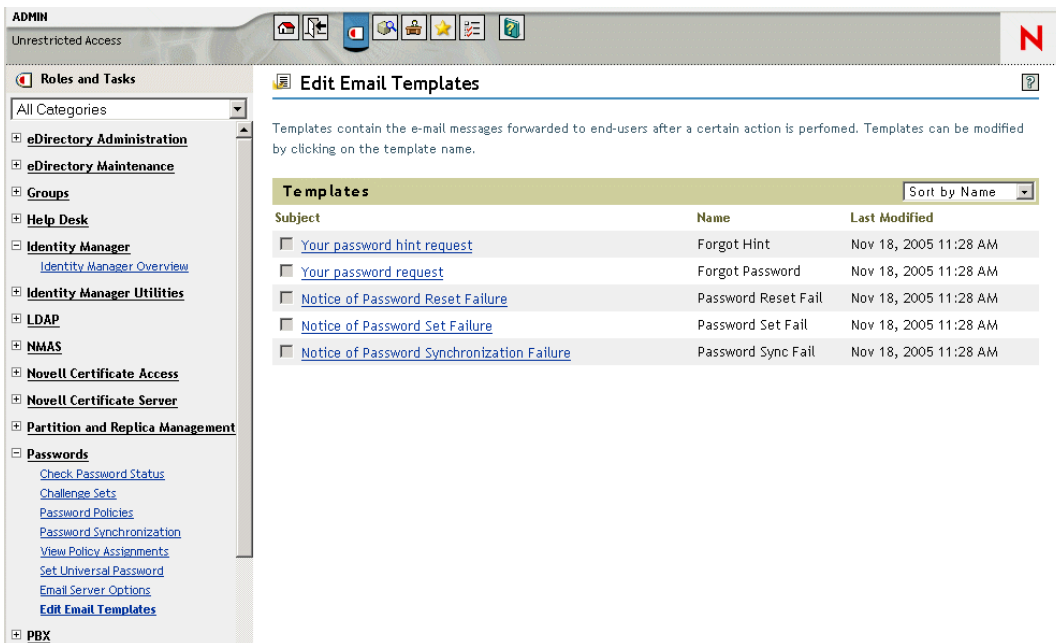
5 Customize the e-mail templates as described in [“Setting Up E-Mail Templates for Notification” on page 146](#).

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*.



- 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 148.

- 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 145. 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 password, the driver policy must provide it.

You must complete this procedure if the following conditions exist:

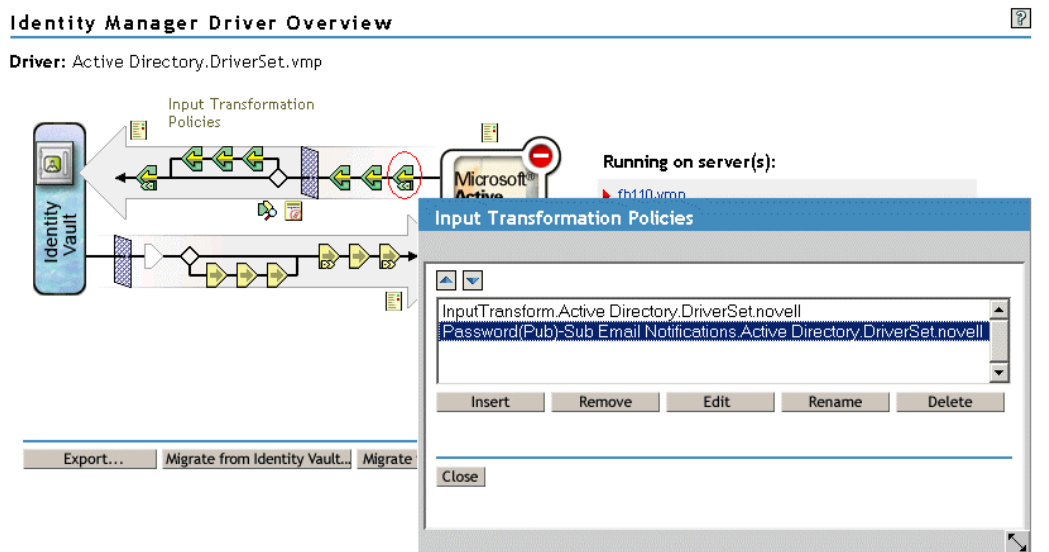
- ◆ The SMTP server is secured and requires authentication before sending e-mail.
- ◆ You are using Identity Manager Password Synchronization with an Identity Manager driver
- ◆ In the Password Synchronization settings for the driver, you have selected *Notify the user of password synchronization failure 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 100.

- 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.

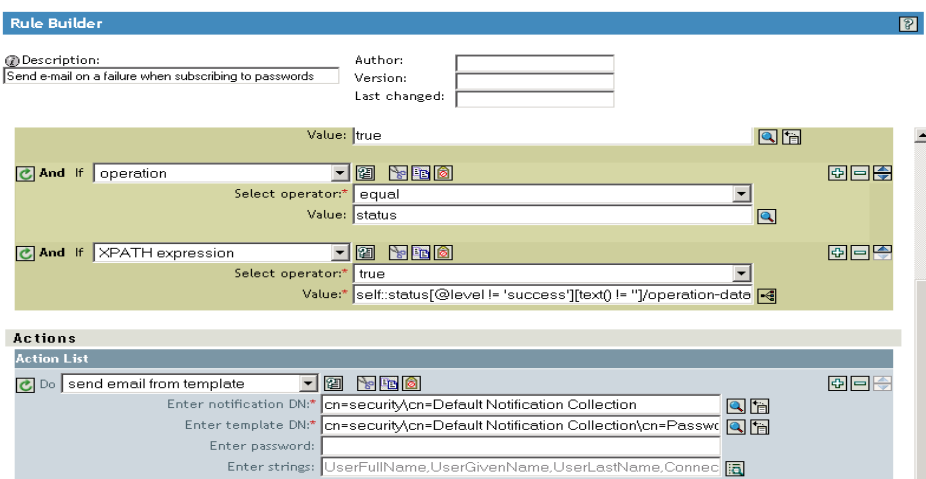


- 6 Select a policy, then click *Edit*.
- 7 Click a rule.
- 8 Specify the password for the SMTP server in the rules that include Do Send E-mail from Template actions.

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	<ul style="list-style-type: none"> ◆ 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
Output Transformation	Password(Sub)-Pub Email Notifications	<ul style="list-style-type: none"> ◆ 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.



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 149](#)
- ◆ [“Adding Replacement Tags to Forgotten Password E-Mail Notification Templates” on page 154](#)

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.

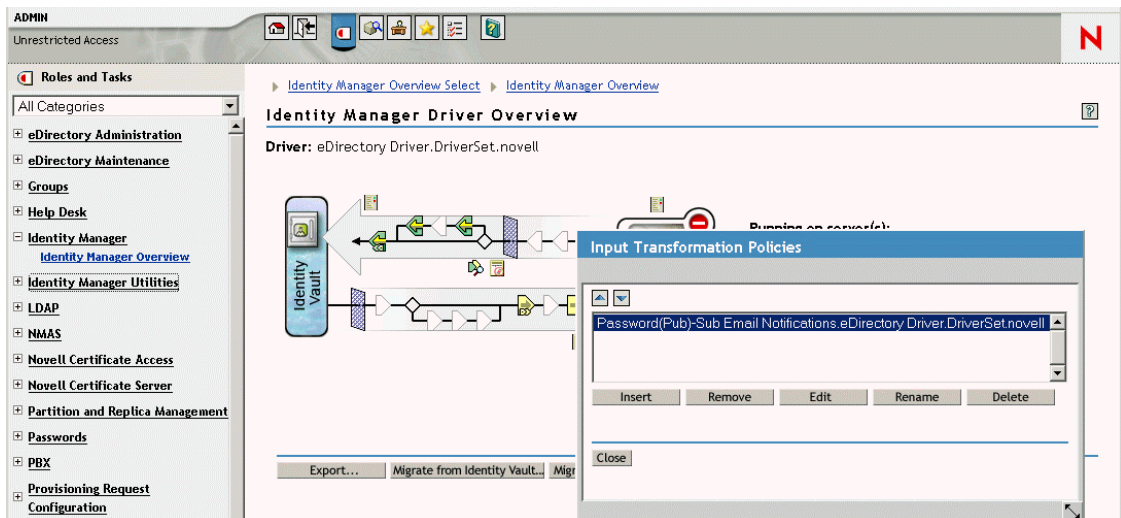
One way to make sure that you find all the policies that refer to the e-mail notification templates is to export your driver configurations, then search the XML for a do-send-e-mail action that has the template equal to the name of the e-mail notification template.

- 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.

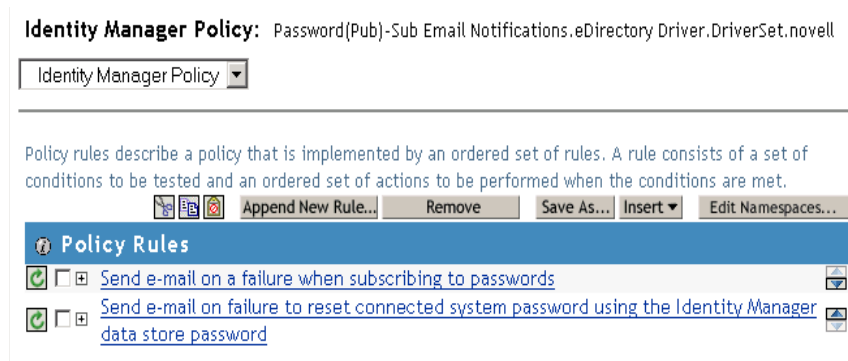
- 5 Click the policy, then click *Edit*.

The following figure illustrates how to edit the Password(Pub)-Sub Email Notifications policy for the eDirectory driver:



- 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.



If you click the first rule, the following page appears:

Rule Builder ?

Description: **Author:**
Version:
Last changed:

Conditions

Select condition structure:

OR Conditions, AND Groups
 AND Conditions, OR Groups

Append Condition Group * Required

Condition Group 1 ✖

If + - ↕

Enter name:* 🔍

Select operator:* ⌵

Compare mode: ⌵

Value: 🔍 📄

And If + - ↕

Select operator:* ⌵

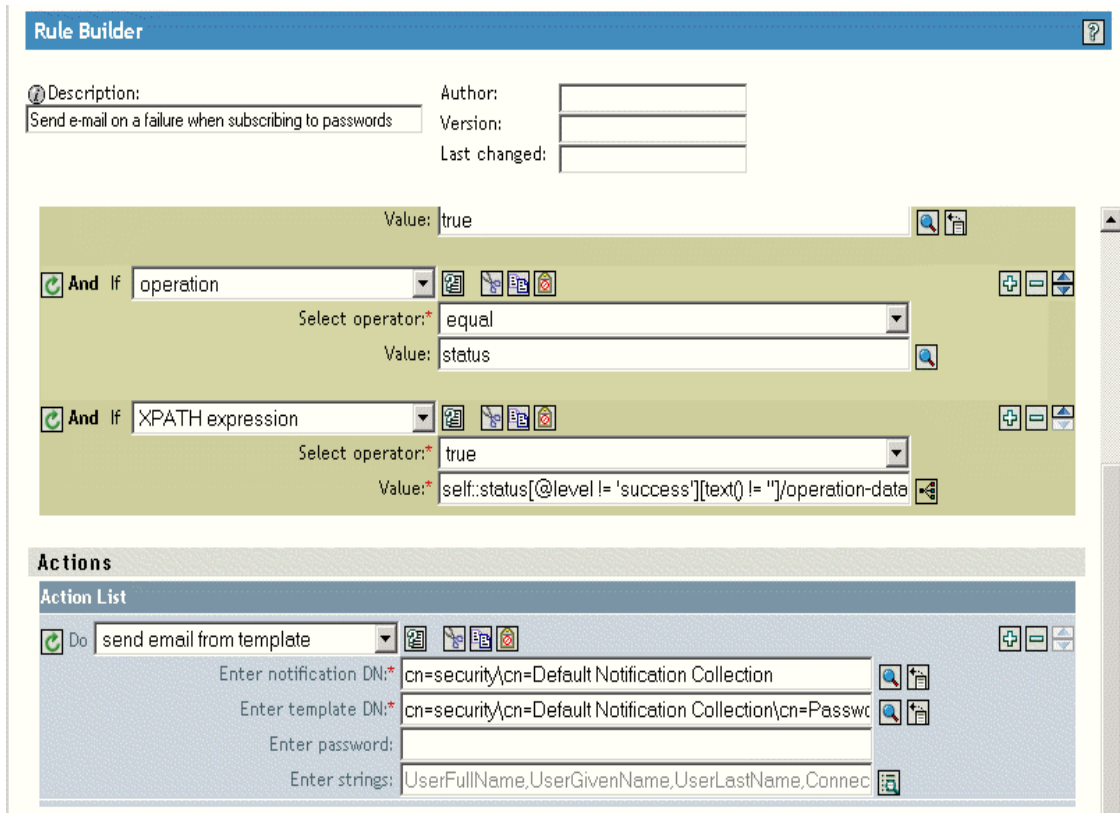
Value: 🔍


And If + - ↕

Select operator:* ⌵

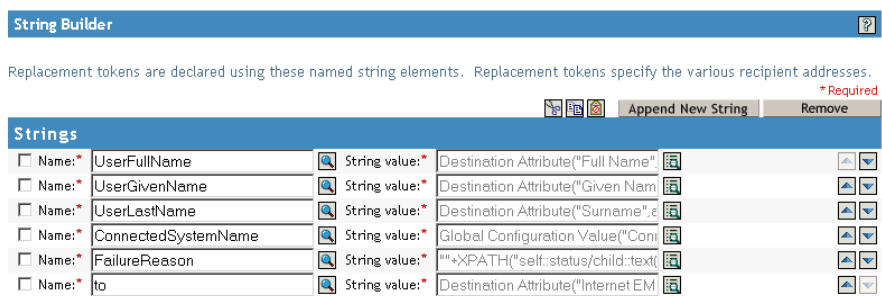
Value:* 🔍

7 Scroll to the *Actions* section.



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. Note that 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.



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 In the Argument Builder page, specify what value should be brought in when this tag is used in an e-mail notification template.

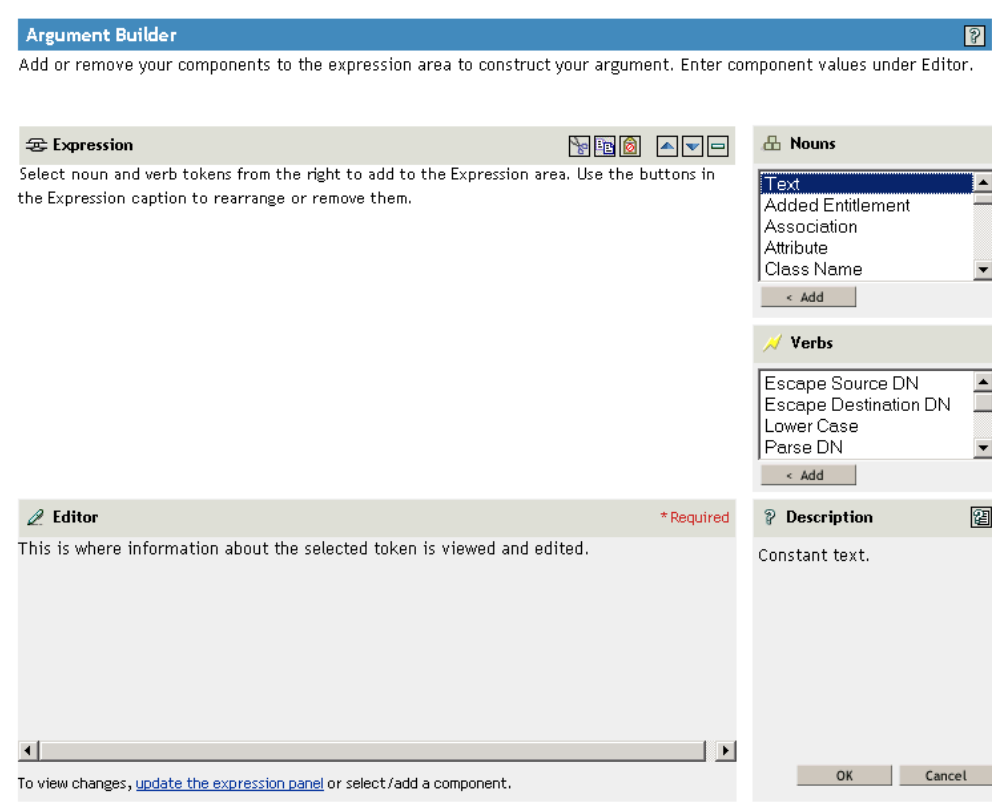
You can define the tag to be any of the following:

- ◆ 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:



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/dirxml/drivers/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 108](#).
- ◆ 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 than 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, in this case 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 following. 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 DTrace 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 162.](#)

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 [“Introduction to Workflow-Based Provisioning.”](#) For information on Role-Based Entitlements, see [Section 6.5, “Managing Role-Based Entitlements Overview,” on page 177.](#)

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.

- ◆ [Section 6.1, “Terminology,” on page 158](#)
- ◆ [Section 6.2, “Creating Entitlements: Overview,” on page 158](#)
- ◆ [Section 6.3, “Entitlement Prerequisites,” on page 161](#)
- ◆ [Section 6.4, “Writing Entitlements in XML through iManager,” on page 162](#)
- ◆ [Section 6.5, “Managing Role-Based Entitlements Overview,” on page 177](#)
- ◆ [Section 6.6, “Creating an Entitlements Service Driver Object,” on page 179](#)
- ◆ [Section 6.7, “Creating Entitlement Policies,” on page 180](#)
- ◆ [Section 6.8, “Conflict Resolution between Role-Based Entitlement Policies,” on page 187](#)
- ◆ [Section 6.9, “Troubleshooting Role-Based Entitlements,” on page 192](#)
- ◆ [Section 6.10, “Entitlement Elements that Apply To Role-Based Entitlements and Workflow-Based Provisioning Entitlements,” on page 193](#)

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 Preconfigurations that Support Entitlements,” on page 159](#)
- ♦ [Section 6.2.2, “Enabling Entitlements on Other Identity Manager Drivers,” on page 160](#)

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 172.](#)
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 Preconfigurations that Support Entitlements

Identity Manager comes with a number of drivers with preconfigurations 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 preconfiguration elements part of the driver. The following drivers have preconfigurations 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

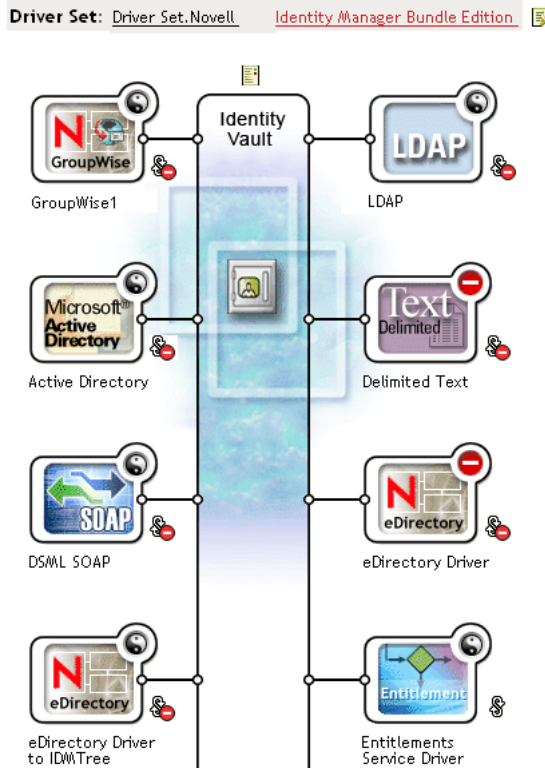
These are example entitlements and policies that you can use as is (if they meet your needs); you can also tweak them to meet your needs, or you can use them as examples and make your own through iManager or Designer. Again, if you want to use the preconfigured driver's entitlements, you must enable 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 3.0.1, run the *Upgrade Entitlements* option under *Identity Manager Utilities*.

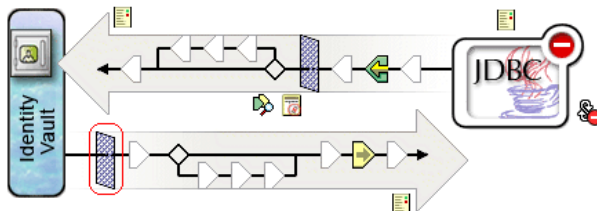
6.2.2 Enabling Entitlements on Other Identity Manager Drivers

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:

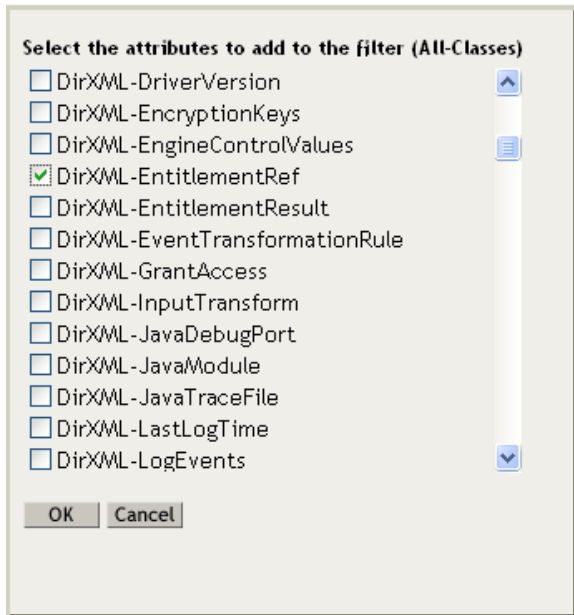
1. Select *Identity Manager > Identity Manager Overview*.
2. Browse to the driver set where the driver resides and click *Search*.
3. From the Identity Manager Overview screen, select the Driver object from the presented Driver Set.



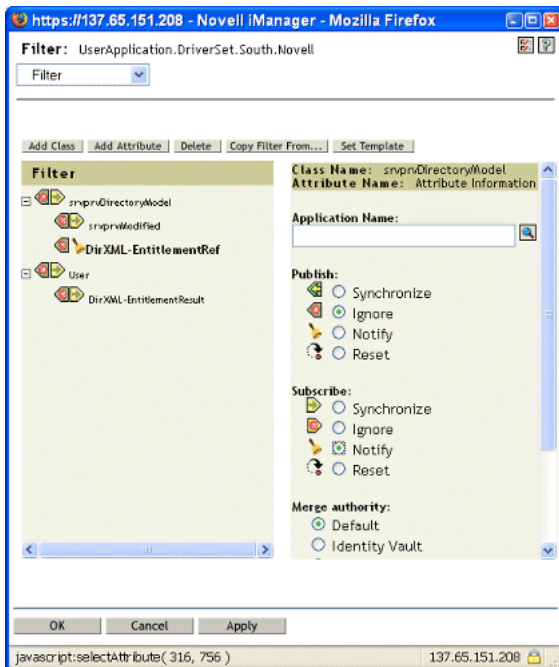
4. Double-click the driver from the Driver Set to bring up the driver screen. Click the *Driver Filter* icon right of the Identity Vault (circled in red).



5. On the Filter page, select *Add Attribute*, then scroll to the bottom and select *Show all attributes*. Select the `DirXML-EntitlementRef` attribute, and click *OK*.



6. Select *DirXML-EntitlementRef* in the Filter page. Under the Subscribe heading, select *Notify*. Click *OK*.



7. 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 or 3

❑ 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:
 - a. Create entitlements using iManager or Designer (Designer is preferred).
 - b. 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 160.](#)
 - c. 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 (AD), 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 162](#)
- ◆ [Section 6.4.2, “Using Novell’s Entitlement Document Type Definition \(DTD\),” on page 166](#)
- ◆ [Section 6.4.3, “Explaining the Entitlement DTD,” on page 168](#)
- ◆ [Section 6.4.4, “Creating Entitlements Through Designer,” on page 171](#)
- ◆ [Section 6.4.5, “Creating and Editing Entitlements in iManager,” on page 171](#)
- ◆ [Section 6.4.6, “Example Entitlements To Help You Create Your Own Entitlements,” on page 172](#)
- ◆ [Section 6.4.7, “Completing the Creating Entitlements Steps,” on page 176](#)

6.4.1 What the Active Directory Driver Adds When Entitlements Are Enabled

The AD 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 AD 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.

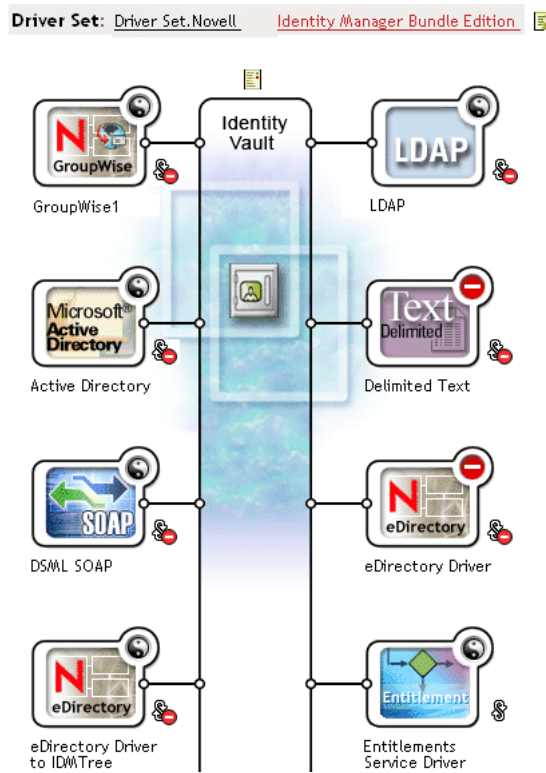
The following policies contain additional rules that allow entitlements to work properly:

- ◆ 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:

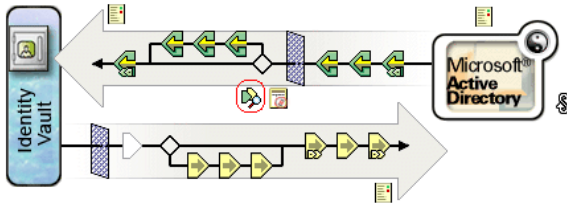
1. Select the *Identity Manager > Identity Manager Overview*.
2. Browse to the driver set where the driver resides and click *Search*.
3. From the Identity Manager Overview page, select the Driver object from the presented Driver Set.



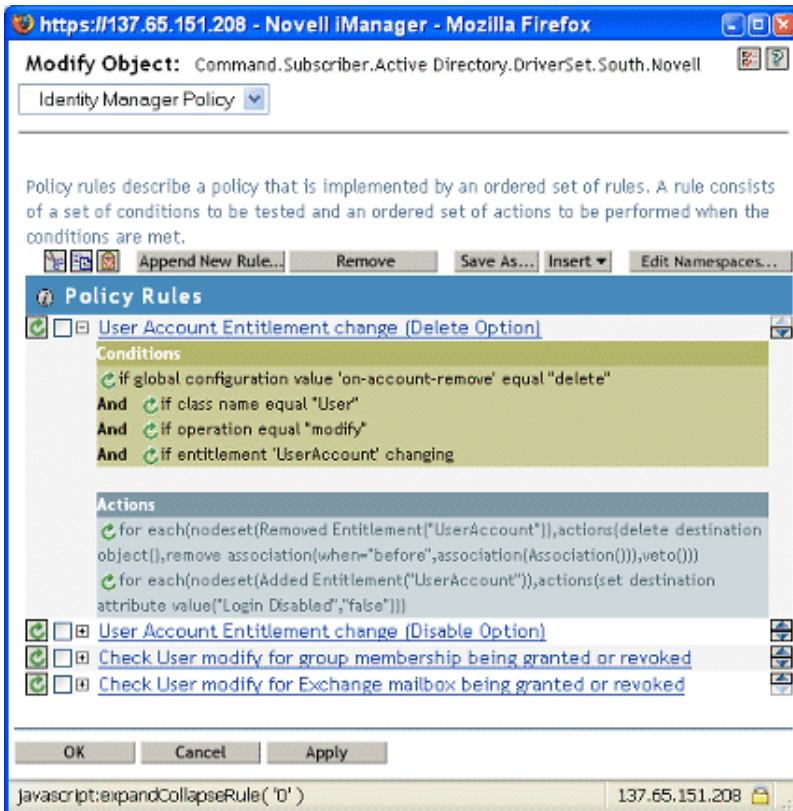
4. Double-click the driver from the Driver Set to bring up the driver page. Click the *View All Policies* icon in the center of the driver (circled in red).

Identity Manager Driver Overview

Driver: Active Directory.DriverSet.South.Novell



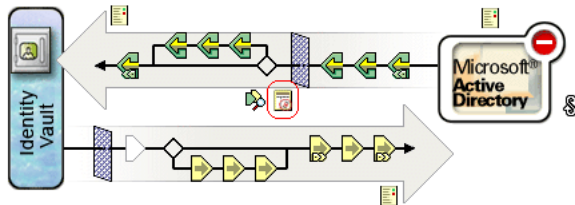
- Once you select a policy from the Show All Policies screen, you can view the conditions and actions that make up the policy.



- To view the actual XML code behind the policies, Select *Edit XML* from the drop-down menu (this menu defaults to Identity Manager Policy).
For information on creating and editing policies, see the *Policy Builder and Driver Customization Guide* and the selected *Identity Manager driver guide* (<http://www.novell.com/documentation/dirxml/drivers/index.html>) for building policies specific to that driver.
- To view the entitlements that come with preconfigured drivers (our example is Active Directory) with entitlements enabled, follow Step 1 through Step 4. However, select the *View All Entitlements* icon in the center of the driver (circled in red).

Identity Manager Driver Overview

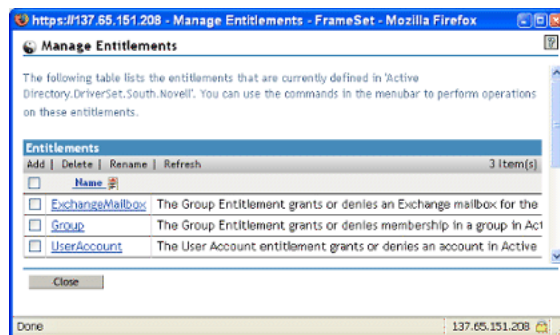
Driver: Active Directory.DriverSet.South.Novell



8. On the Manage Entitlements page, click the entitlement name to bring up the entitlement in the XML viewer. To edit the entitlement's code, click *Enable XML Editing*.

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 AD Driver



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 172.

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 173](#).

```
<!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 \(http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/ref/ndsstd/query.html\)](http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/ref/ndsstd/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 171](#) 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 the Create Entitlements option under the Identity Manager Utilities heading.
2. On the Create Entitlements page, type the name you want the entitlement to be, then use the Object Browser to find the Identity Manager Driver object to which the entitlement belongs.

Create Entitlement

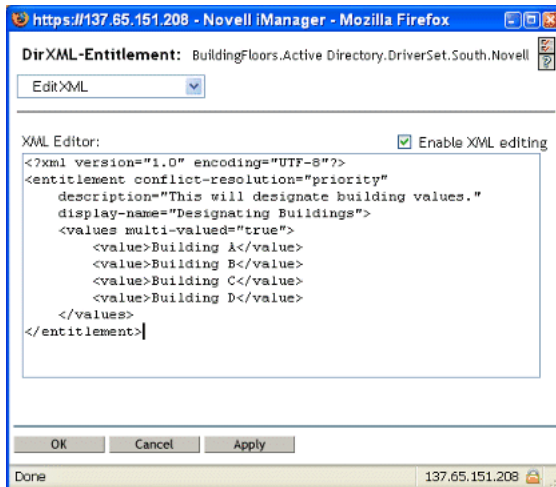
Name:*

Context:*  

(An Entitlement object may only be created within a 'Driver' object.)

Define additional properties

3. If Define Additional Properties is selected, you see the XML Editor page where you define the elements you want for this entitlement.



4. Check Enable XML Editing to add your elements to 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.

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 ActiveDirectory 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>
  </result-set>
</values>

```

```

        <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 158](#). 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 the *Policy Builder and Driver Customization Guide* and the appropriate *Identity Manager driver guide* (<http://www.novell.com/documentation/idmdrivers/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 “[Introduction to Workflow-Based Provisioning](#).” 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 177](#)

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 160](#).
2. Install the Entitlements Service driver (`Entitlement.xml`) as described in [Section 6.6, “Creating an Entitlements Service Driver Object,” on page 179](#).
3. Create Role-Based Entitlement Policies in iManager, as described in [Section 6.7, “Creating Entitlement Policies,” on page 180](#).

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 Preconfigurations that Support Entitlements,” on page 159](#), 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.

- ◆ 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 180**.

- 2 In the No Entitlements Service Driver page, click *Yes*.

The Create Driver Wizard opens.

You can also click *DirXML Utilities > Import Drivers*.

- 3 In the Create Driver Wizard page, Select *In an Existing Driver Set*, then click *Next*.

- 4 In the *Import a Driver Configuration from the Server (.XML file)* drop-down list, select *Entitlement.xml*.

Import or create a new Application Driver for this driver set.

Import a driver configuration from the server (.XML file)
Entitlement.xml

Import a driver configuration from the client (.XML file)
File:

Create a new driver
Name:

- 5 Name the 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: * Existing drivers:

The correct driver configuration file is chosen automatically. Just specify a name for the driver object or use the default.

- 6 We recommend that you define security equivalences and exclude administrative roles. Add user Admin to both of these selections, then click *Next*.
- 7 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

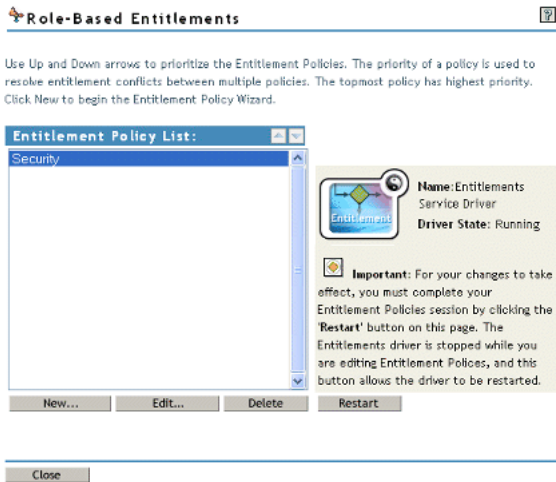
- ♦ [Section 6.7.1, “Defining Membership for an Entitlement Policy,” on page 182](#)
- ♦ [Section 6.7.2, “Choosing Entitlements for an Entitlement Policy,” on page 183](#)

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.



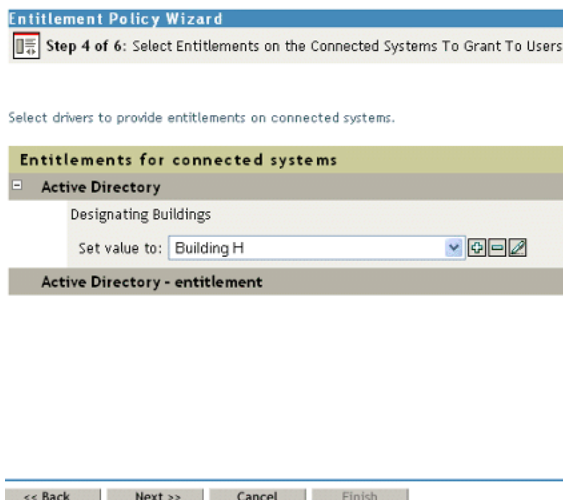
4 Click *New*.

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 1 through Step 6 in the wizard to create a new policy. Refer to the online help for information about each step in the wizard.

- 5a** In Step 1, give the policy a name and description.
- 5b** In Step 2, define the membership filter the search parameters.
- 5c** In Step 3, define static members by including and excluding members in the search criteria.
- 5d** In Step 4, select an Identity Manager driver and provide entitlements for inclusion. You created entitlements in [Section 6.4, “Writing Entitlements in XML through iManager,” on page 162](#). Click *Add Driver*, then select an entitlement to add.



- 5e** In Step 5, browse for objects for which you want this entitlement policy to be a trustee.
- 5f** In Step 6, read the summary to ensure that the entitlement policy does what you want it to. If it does, click *Finish*; if not, click *Back*.
- 6** Entitlement policy creation turns off the Entitlements Service driver. Click *Restart* to complete the session.

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.

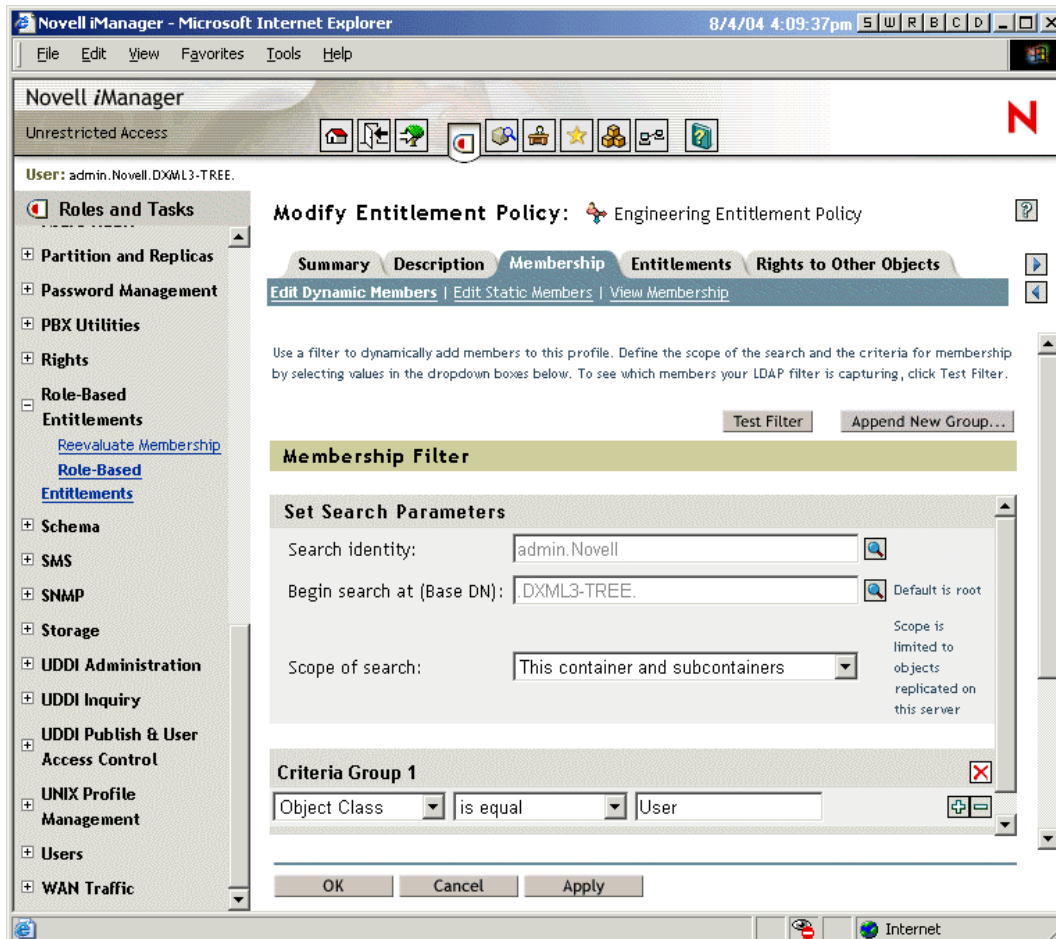
An Entitlement policy is a dynamic group object. You can define membership for an Entitlement policy by using two methods, dynamic and static. You can use both methods in the same Entitlement policy.

- ♦ **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



- ♦ **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 184
- ♦ “Membership in E-Mail Distribution Lists and NOS Lists” on page 185
- ♦ “Attribute Values on Connected Systems” on page 186

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:

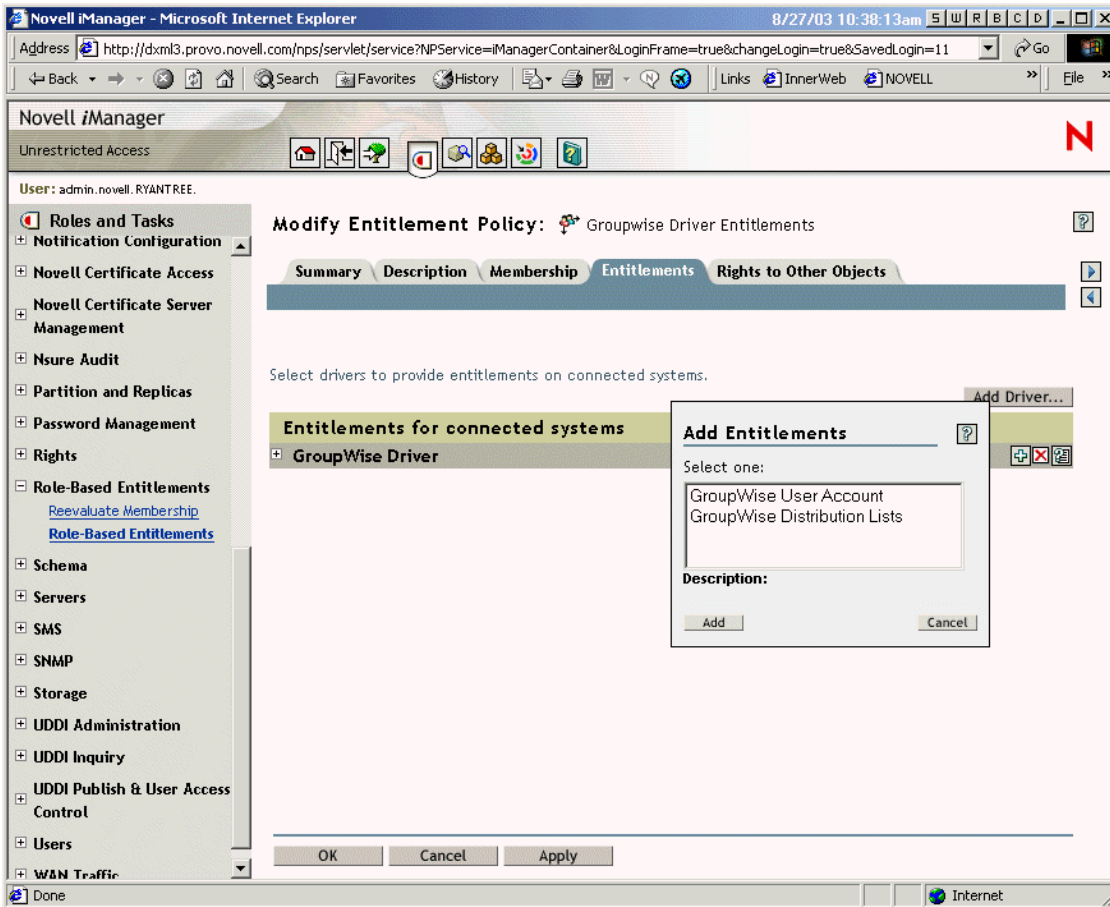
- ◆ Accounts
- ◆ Membership in e-mail distribution lists
- ◆ Group membership in NOS lists
- ◆ Attributes for the corresponding objects in connected systems, populated with values you specify
- ◆ Other entitlements that you customize

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

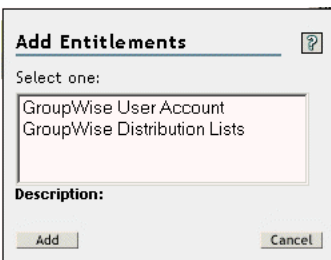


Membership in E-Mail Distribution Lists and NOS Lists

To assign membership in groups on connected systems, you choose the membership entitlement from the list of entitlements offered by a driver.

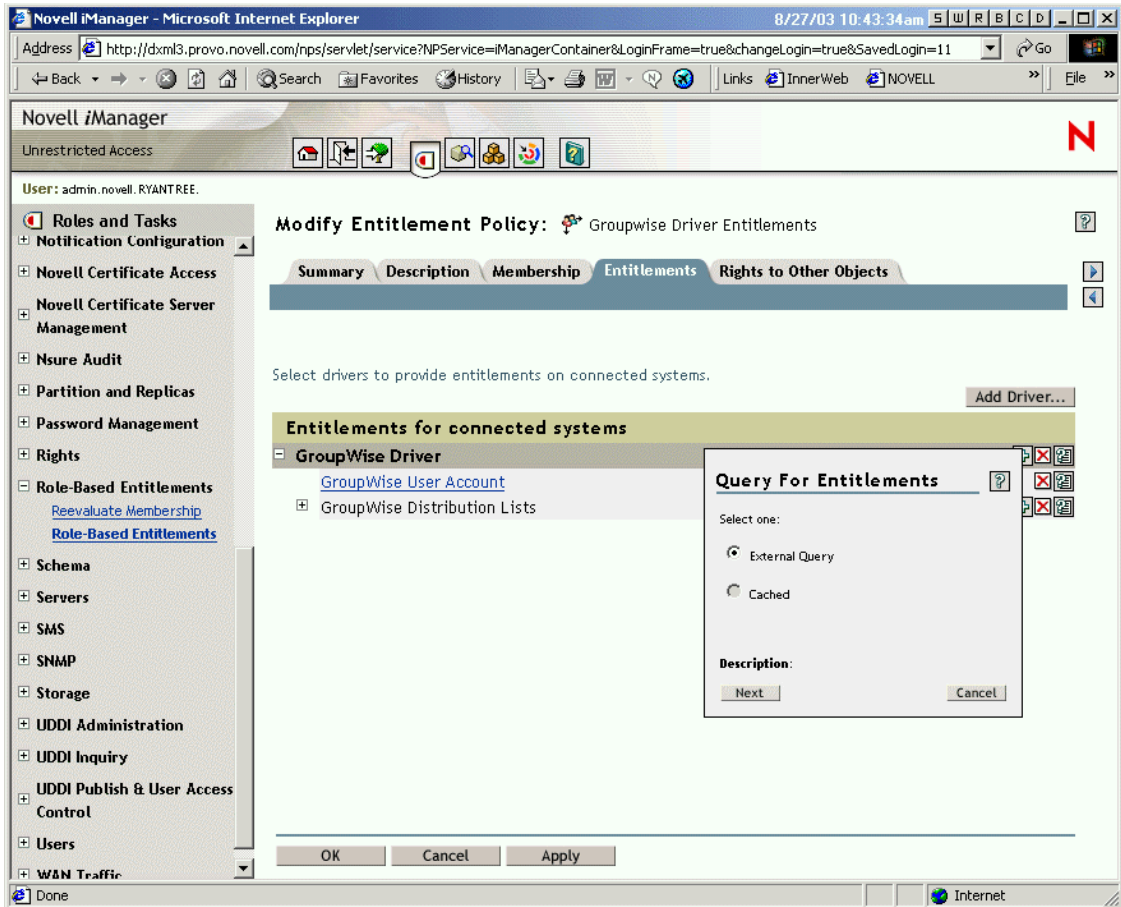
The following figure shows an example, with GroupWise Distribution Lists shown second in the list.

Figure 6-4 Selecting GroupWise Distribution Lists



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



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.

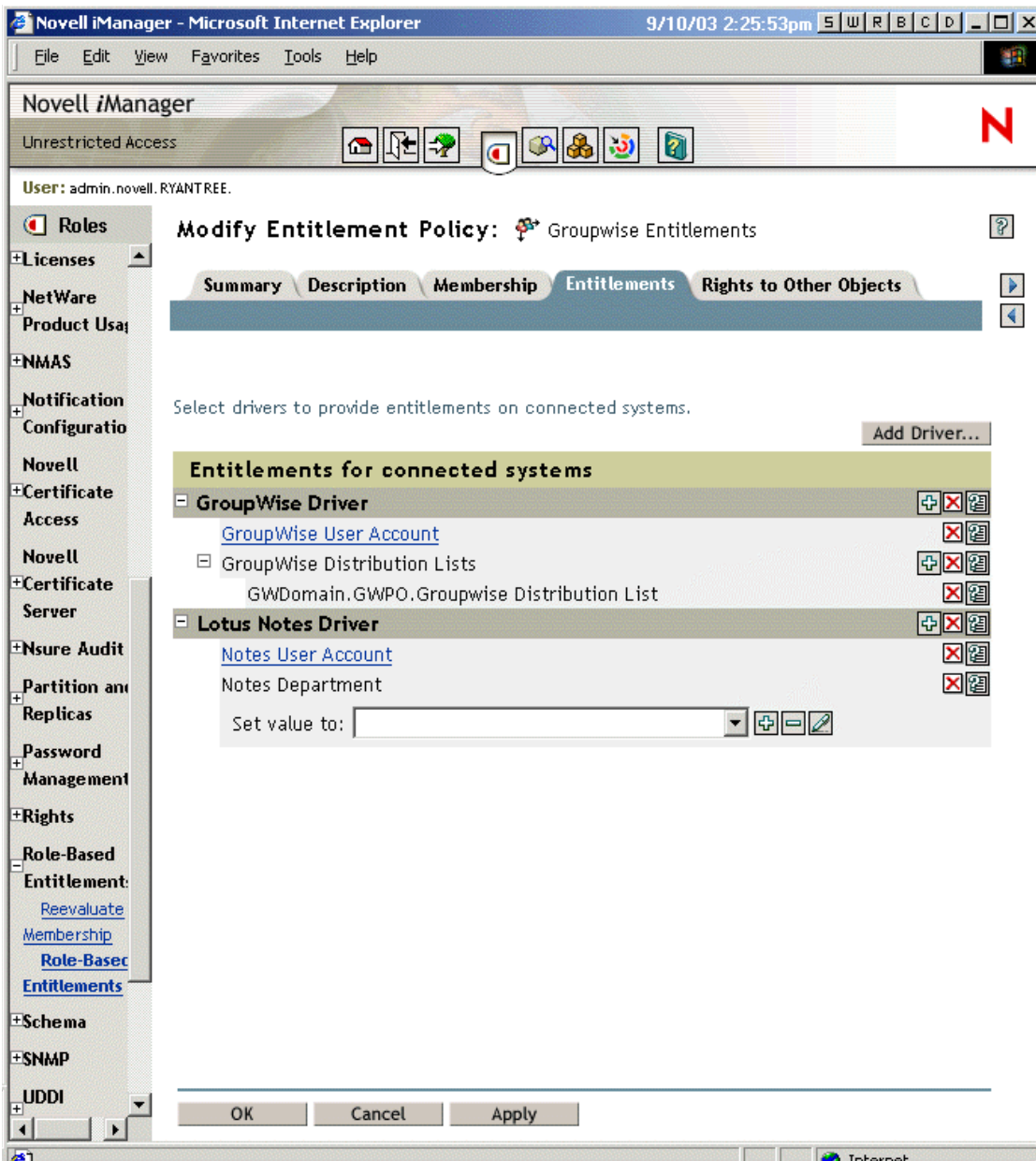
NOTE: A driver could be customized to limit the list to group names you specify, rather than a query that returns the complete list.

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



6.8 Conflict Resolution between Role-Based Entitlement Policies

- ◆ Section 6.8.1, “Conflict Overview,” on page 188
- ◆ Section 6.8.2, “Changing the Conflict Resolution Method for an Individual Entitlement,” on page 189
- ◆ Section 6.8.3, “Prioritizing Entitlement Policies,” on page 191

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 AD group named Mailroom Staff by the Mailroom policy, and also granted membership in the AD group named Emergency Response by the Emergency Volunteers policy, she is granted membership in both groups in AD.

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 AD 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.

A page appears with a graphical representation of all the drivers in the driver set.

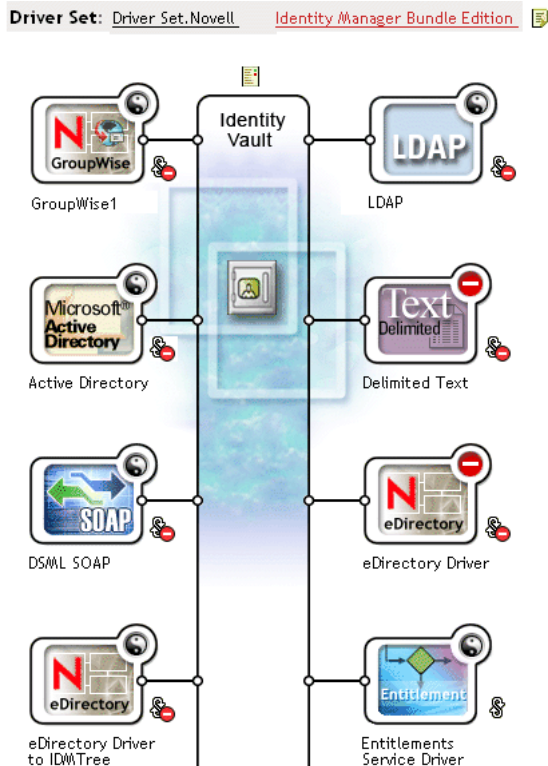
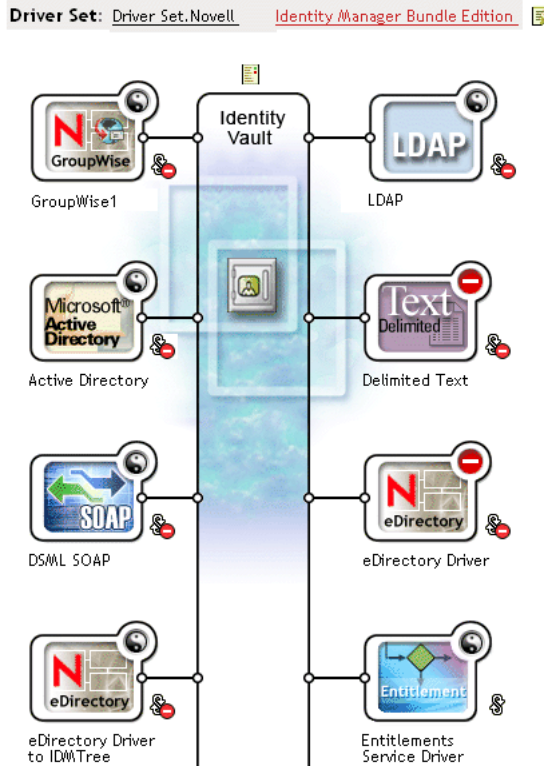


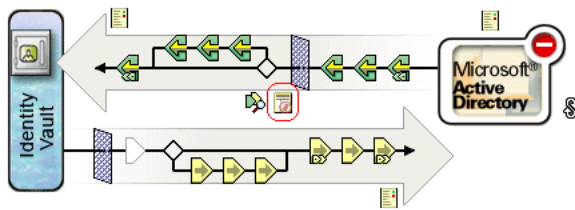
Figure 6-7 Driver Set



- 2 Click the Driver status button and select *Stop driver*.
- 3 Click the driver icon for the driver that offers the entitlement you want to change.
A page appears showing icons for the driver's policies and the driver. Select the *View All Entitlements* icon in the middle of the screen (circled in red).

Identity Manager Driver Overview

Driver: Active Directory.DriverSet.South.Novell



- 4 At the Manage Entitlements page, click on the entitlement name to bring up the entitlement in the XML viewer.
- 5 Select the check box for *Enable XML editing*.
- 6 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">
```

7 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 187](#).

8 Click *Restart* to restart the Entitlements Service 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 187](#).

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.

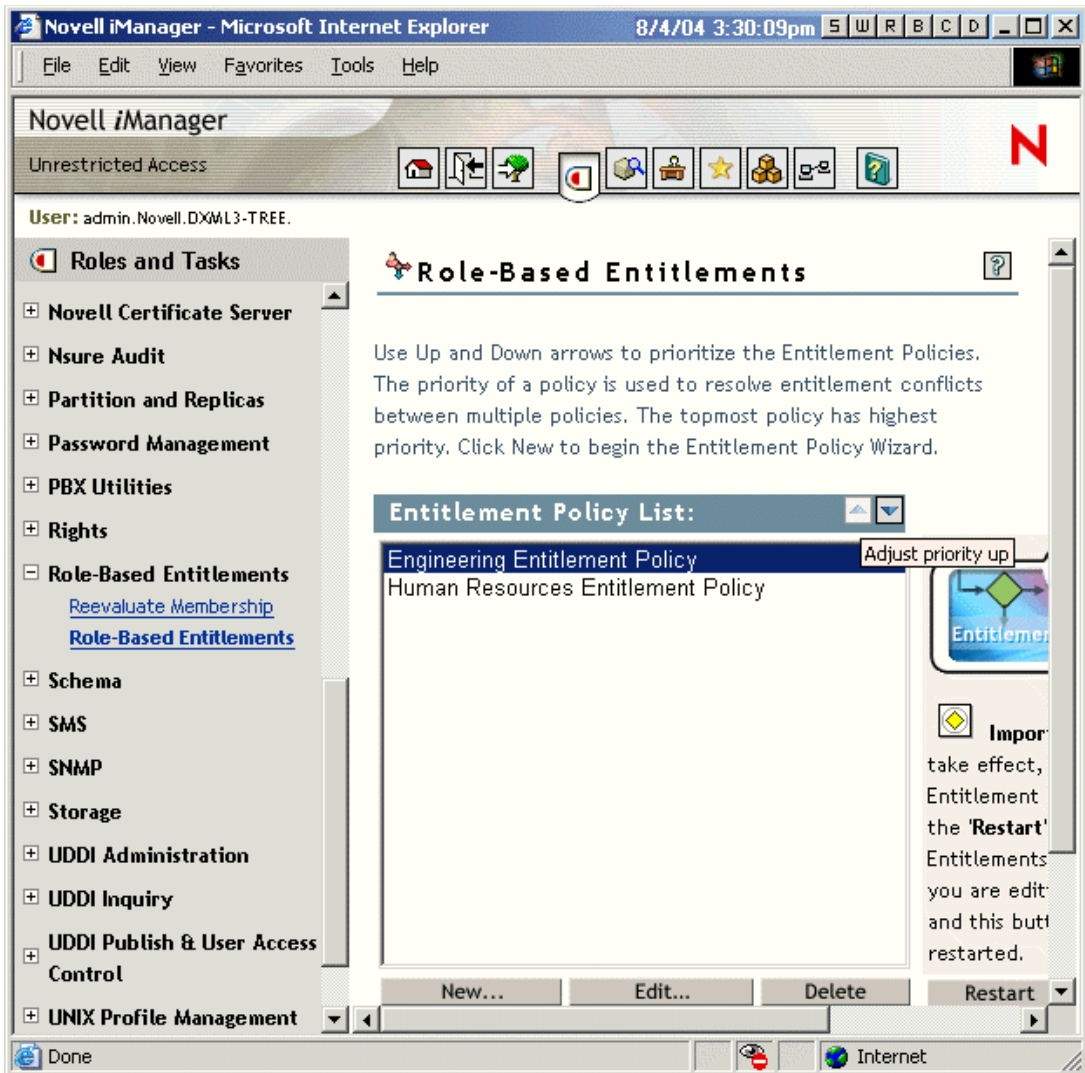
1 In iManager, click *Role-Based Entitlements > Role-Based Entitlements*.

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.



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.

- ♦ [Section 6.10.1, “Controlling the Meaning of Granting or Revoking Entitlements,” on page 193](#)
- ♦ [Section 6.10.2, “Preventing Data Loss,” on page 193](#)
- ♦ [Section 6.10.3, “Password Synchronization and Entitlements,” on page 194](#)

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 75](#).

- ♦ [Section 7.1, “Using SSL,” on page 195](#)
- ♦ [Section 7.2, “Securing Access,” on page 195](#)
- ♦ [Section 7.3, “Managing Passwords,” on page 195](#)
- ♦ [Section 7.4, “Creating Strong Password Policies,” on page 196](#)
- ♦ [Section 7.5, “Securing Connected Systems,” on page 197](#)
- ♦ [Section 7.7, “Industry Best Practices for Security,” on page 198](#)
- ♦ [Section 7.8, “Tracking Changes to Sensitive Information,” on page 199](#)

7.1 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 47](#)), and between the Metadirectory engine or Remote Loader and the connected systems.

If you don't enable SSL, you are sending information such as passwords in the clear.

7.2 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.

Access Rights. Administrative rights are needed 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

7.3 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 \(http://www.novell.com/documentation/password_management/index.html\)](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 \(http://www.novell.com/documentation/password_management/index.html\)](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 \(http://www.novell.com/documentation/password_management/index.html\)](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.

7.4 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 you have in place. If your 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 \(http://www.novell.com/documentation/password_management/index.html\)](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 108.

Using Universal Password and password policies allows you to enforce strong password requirements for your users. Use the Advanced Password Rules in password policies to follow industry best practices for passwords.

For example, 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 8.7.3 Administration Guide* (<http://www.novell.com/documentation/edir873/edir873/data/afxkmdi.html#amm7bjv>).

7.5 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.

7.6 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.

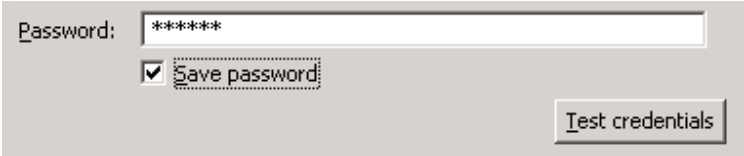
To save a password for a session, but not save it to the project:

- a. In an expanded Outline view, right-click an Identity Vault.
- b. Select Properties.
- c. 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 7-1 Save Password



The screenshot shows a dialog box with a 'Password:' label and a text input field containing '*****'. Below the input field is a checked checkbox labeled 'Save password'. At the bottom right of the dialog is a button labeled 'Test credentials'.

7.7 Industry Best Practices for Security

Follow industry best practices for security measures, such as blocking unused ports on the server.

7.8 Tracking Changes to Sensitive Information

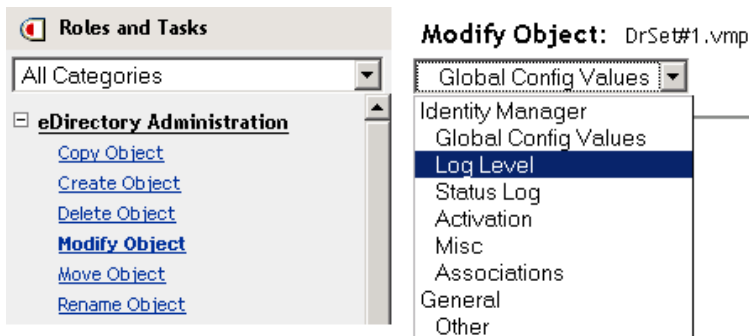
- ♦ Section 7.8.1, “Logging Events by Using iManager,” on page 199
- ♦ Section 7.8.2, “Logging Events by Using Designer,” on page 200

7.8.1 Logging Events by Using iManager

You can use Novell Audit to log events that you consider important for security. For information on Novell Audit, see [Chapter 10, “Logging and Reporting Using Novell Audit,”](#) on page 227.

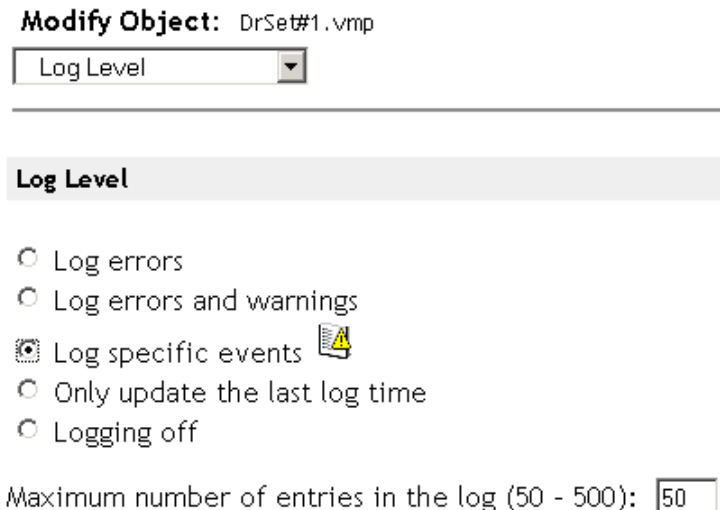
For example, you could log password changes for a particular Identity Manager driver (or driver set) by doing the following:


- 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*.



- 3 To select the specific events, click the log events icon .

4 On the Events page, select the following:

Operation Events		
<input type="checkbox"/> Search	<input type="checkbox"/> Add	<input type="checkbox"/> Remove
<input type="checkbox"/> Modify	<input type="checkbox"/> Rename	<input type="checkbox"/> Move
<input type="checkbox"/> Add Association	<input type="checkbox"/> Remove Association	<input type="checkbox"/> Query Schema
<input type="checkbox"/> Check Password	<input type="checkbox"/> Check Object Password	<input checked="" type="checkbox"/> Change Password
<input type="checkbox"/> Sync	<input type="checkbox"/> Clear Attribute	<input type="checkbox"/> Add Value
<input type="checkbox"/> Remove Value	<input type="checkbox"/> Merge Entry	

Transformation Events		
<input type="checkbox"/> Initial Document	<input type="checkbox"/> Input	<input type="checkbox"/> Output
<input type="checkbox"/> Event	<input type="checkbox"/> Placement	<input type="checkbox"/> Create
<input type="checkbox"/> Input Mapping	<input type="checkbox"/> Output Mapping	<input type="checkbox"/> Matching
<input type="checkbox"/> Command	<input type="checkbox"/> Driver Filter	<input type="checkbox"/> User Agent Request
<input type="checkbox"/> Resync Request	<input type="checkbox"/> Migrate Request	<input checked="" type="checkbox"/> Password Sync
<input checked="" type="checkbox"/> Password Set		

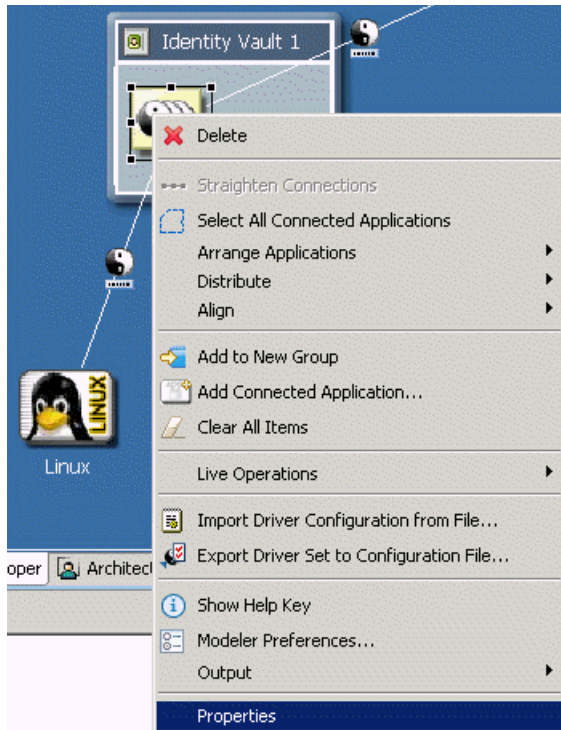
- ◆ In Operation Events, select *Change Password*.
This item monitors direct changes to the NDS password.
- ◆ In Transformation Events, select *Password Set* and *Password Sync*. These two items monitor events for the Universal Password and Distribution Password.

5 Click *OK* twice.

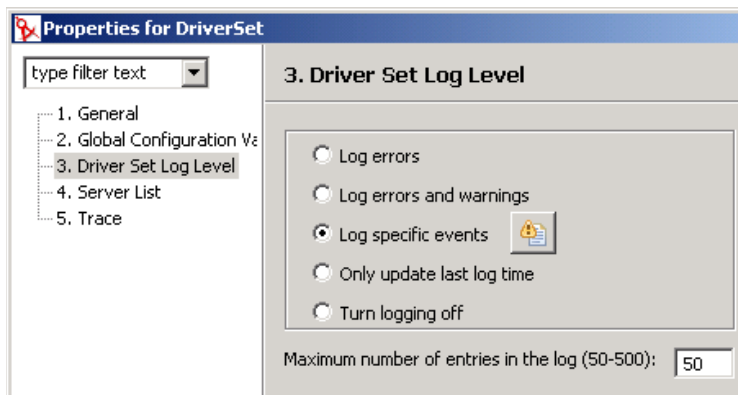
7.8.2 Logging Events by Using Designer

You can log events that apply to a driver set or to a driver.


Logging Events for a Driver Set

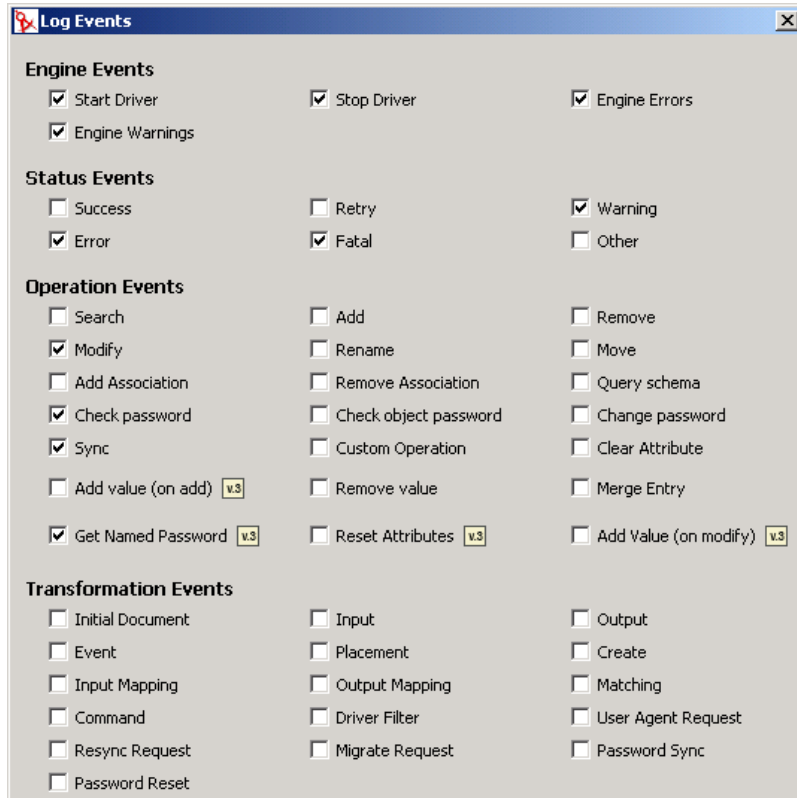


1 In Designer, right-click a driver set, then select *Properties*.



2 Select *Driver Set Log Level*, then select *Log Specific Events*.

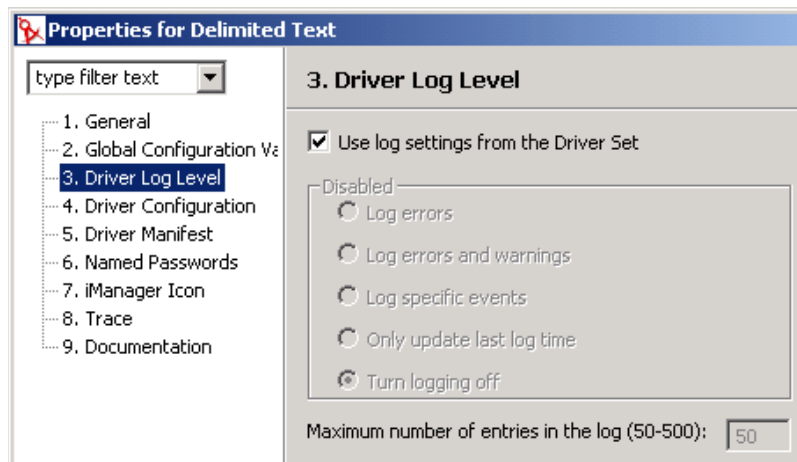
3 Click the *Select Events to Log* icon .



4 Select events to log, then click *OK*.

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*.

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.
- 4** Select events to log, then click *OK*.

Managing Engine Services

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 8.1, “Entitlements Service Driver,” on page 205](#)
- ♦ [Section 8.2, “Manual Task Service Driver,” on page 205](#)

8.1 Entitlements Service Driver

See [Chapter 6, “Creating and Using Entitlements,” on page 157](#).

8.2 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, Web server port number, etc., must be configured.

In this section:

- ♦ [Section 8.2.1, “Installing,” on page 205](#)
- ♦ [Section 8.2.2, “Overview,” on page 206](#)
- ♦ [Section 8.2.3, “Configuring,” on page 213](#)
- ♦ [Section 8.2.4, “Additional Information,” on page 220](#)

8.2.1 Installing

- ♦ **Installation:** The Driver for Manual Task Service 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.

8.2.2 Overview

In this section, you will find information about how various driver functionality works.

- ♦ [“Modes of Operation” on page 206](#)
- ♦ [“How E-Mail Messages and Web Pages Are Created by the Manual Task Service Driver” on page 207](#)
- ♦ [“Templates” on page 208](#)
- ♦ [“Replacement Tokens” on page 210](#)
- ♦ [“Replacement Data” on page 210](#)
- ♦ [“Template Action Elements” on page 211](#)
- ♦ [“Subscriber Channel E-Mail” on page 211](#)
- ♦ [“Publisher Channel Web Server” on page 212](#)

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 DirXML driver for the company's HR system).
2. The Manual Task Service Driver Subscriber 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 (for example, by the DirXML driver for the company's HR system).
2. The Manual Task Service Driver Subscriber 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 a DirXML 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 DirXML 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 DirXML 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 if desired.

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

This section describes document creation templates as used in the Manual Task Service Driver.

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 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 and appear in bold.

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 299](#). In the following examples action elements also appear in bold.

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 291](#).

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 297.](#)

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`.

The elements appearing in bold in the above examples are action elements.

Action elements are described in detail in [Appendix F, “Manual Task Service Driver: Template Action Elements Reference,” on page 299.](#)

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 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 303.](#)

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 submits the URL from the e-mail message to the Web server using a Web browser. 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 authenticates the user using the user's DN and password. 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, then 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.

8.2.3 Configuring

This section describes configuring the Manual Task Service Driver's parameters and templates.

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

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 Documents" 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 on which of multiple, local IP addresses the Publisher channel Web server will 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.

Note that 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, then 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 below).

Using SSL is recommended for security reasons because eDirectory passwords are passed in HTTP POST data when using the Publisher channel Web serve

Name of Keystore File

This parameter, together with 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, then 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

Settings for the Subscriber channel are described in this section.

SMTP Server

This parameter specifies the name of the SMTP server that the Subscriber channel will use to send e-mail messages.

SMTP Account Name

If the SMTP server specified using the SMTP server parameter requires authentication then 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, then 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 319.

Publisher Settings

Settings for the Publisher channel are described in this section.

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.HttpServlet`. 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 321.

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 DirXML 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, then 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, then 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, then queries must be made to obtain information not contained in the modify command.

Blocking Commands from Reaching the Subscriber

If e-mail messages are to be generated from events other than add events, then add events must be allowed to reach the Subscriber for those objects that are to be monitored. Allowing add events to reach the Subscriber results in a generated association value being returned to Identity Manager from the Subscriber.

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 will 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. The Subscriber handles only <add> commands and <mail> commands. Other commands result in the Subscriber 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 303](#) 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 should contain a URL that points to the Publisher channel Web server and is used to solicit information from a user, then 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 [Section , “Replacement Data,” on page 210](#)) 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>) then 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.xml. If the MIME type is anything other than text/plain, the default style sheet name is process_template.xml.

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 [Section , “Templates,” on page 208](#) 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\$ then 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, then undesirable XML escaping will occur. The default Manual Task Service Driver style sheet, `process_text_template.xml`, 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 it is possible to construct the Web page and XDS templates so they result in exactly the XDS required and the XDS need not be further processed by policies.

If policies are required they will be 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 [Section , “Templates,” on page 208](#) 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 307](#).

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 under the Overview section 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=some 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=some 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 which 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 (DirXML Engine traces XML documents at this level and above)
3	No additional messages
4	Messages relating to document construction from templates and style sheets
5	Replacement data documents traced

8.2.4 Additional Information

For additional information on Manual Task Service driver settings, refer to the following appendix sections:

- ◆ [Appendix D, “Manual Task Service Driver: Replacement Data,” on page 291](#)
- ◆ [Appendix E, “Manual Task Service Driver: Automatic Replacement Data Items,” on page 297](#)
- ◆ [Appendix F, “Manual Task Service Driver: Template Action Elements Reference,” on page 299](#)
- ◆ [Appendix G, “Manual Task Service Driver: <mail> Element Reference,” on page 303](#)
- ◆ [Appendix H, “Manual Task Service Driver: Data Flow Scenario for New Employee,” on page 307](#)
- ◆ [Appendix I, “Manual Task Service Driver: Custom Element Handlers for the Subscriber Channel,” on page 319](#)
- ◆ [Appendix J, “Manual Task Service Driver: Custom Servlets for the Publisher Channel,” on page 321](#)

High Availability

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 9.1, “Configuring eDirectory and Identity Manager for Use with Shared Storage on Linux and UNIX,” on page 221](#)
- ♦ [Section 9.2, “Case Study for SuSE Linux,” on page 225](#)

9.1 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 9.1.1, “Installing eDirectory,” on page 222](#)

- ♦ [Section 9.1.2, “Installing Identity Manager,” on page 222](#)
- ♦ [Section 9.1.3, “Sharing NICI Data,” on page 222](#)
- ♦ [Section 9.1.4, “Sharing eDirectory and Identity Manager Data,” on page 223](#)
- ♦ [Section 9.1.5, “Identity Manager Driver Considerations,” on page 224](#)

9.1.1 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.

9.1.2 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 Server* 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.

9.1.3 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.

9.1.4 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 223](#)
- ♦ [“On the Secondary Node” on page 224](#)

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
<code>/shared/var/nds/class16.conf</code>	<code>/etc/class16.conf</code>
<code>/shared/var/nds/class32.conf</code>	<code>/etc/class32.conf</code>
<code>/shared/var/nds/help.conf</code>	<code>/etc/help.conf</code>
<code>/shared/var/nds/ndsmonhealth.conf</code>	<code>/etc/ndsmonhealth.conf</code>
<code>/shared/var/nds/miscicon.conf</code>	<code>/etc/miscicon.conf</code>
<code>/shared/var/nds/ndsmon.conf</code>	<code>/etc/ndsmon.conf</code>
<code>/shared/var/nds/macaddr</code>	<code>/etc/macaddr</code>

- 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 `nds` and verify that `nds` runs with the shared storage.
- 10 Stop `nds`.
- 11 Place `nds` in the cluster manager's list of resources to be hosted.
- 12 Remove `nds` 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 `nds` in the cluster manager's list of resources to be hosted.
- 7 Remove `nds` 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.

9.1.5 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.

9.2 Case Study for SuSE Linux

For a description of running Identity Manager on shared storage with SUSE LINUX Enterprise Server 8, see [TID10093317 \(http://support.novell.com/cgi-bin/search/searchtid.cgi?10093317.htm\)](http://support.novell.com/cgi-bin/search/searchtid.cgi?10093317.htm).

Logging and Reporting Using Novell Audit

Identity Manager is instrumented to use Novell[®] Audit for auditing and reporting.

10.1 Overview

Novell Audit is a collection of technologies providing monitoring, logging, reporting and notification capabilities. Through integration with Novell Audit, Identity Manager provides detailed information about the current and historical status of driver and engine activity. This information is provided by a set of preconfigured reports, standard notification services, and user-defined data logging.

You can monitor real-time Identity Manager events, send e-mail notifications for any Identity Manager event, and generate reports of Identity Manager activity using Novell Audit.

The types of messages sent to Novell Audit are controlled using plug-ins similar to those provided with the Reporting and Notification Service (RNS). Additional levels are added to these plug-ins, to select the type of operations or debug information you would like to track, such as status, add entry, search, and so on.

Reporting and Notification Services

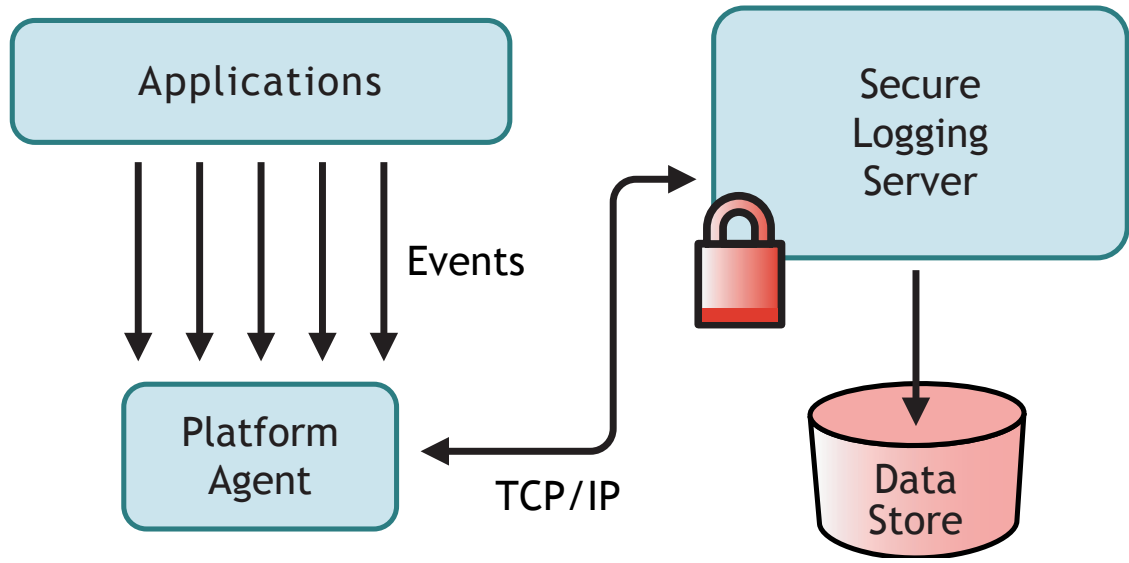
Reporting and Notification Service (RNS) is deprecated, although the Metadirectory engine continues to process RNS functions if you are currently using RNS. You should plan to move to Novell Audit, because Novell Audit expands the functionality provided by RNS, and RNS might no longer be supported in a future release of Identity Manager. For RNS documentation, see the *DirXML 1.1a Administration Guide* (<http://www.novell.com/documentation/lg/dirxml11a/dirxml/data/afae8bz.html>).

10.2 Novell Audit

Novell Audit is a centralized, cross-platform logging service that can log data from multiple applications to a centralized data store. After event data is logged, you can run detailed reports, custom queries, and trigger notifications based on the logged events.

The following figure illustrates the high-level architecture of Novell Audit:

Figure 10-1 Architecture Overview



In this illustration, Identity Manager is one of the applications that uses the Platform Agent to report events to the Novell Audit Secure Logging Server.

10.3 Setting Up Novell Audit

As described in the Overview, Novell Audit consists of two fundamental components:

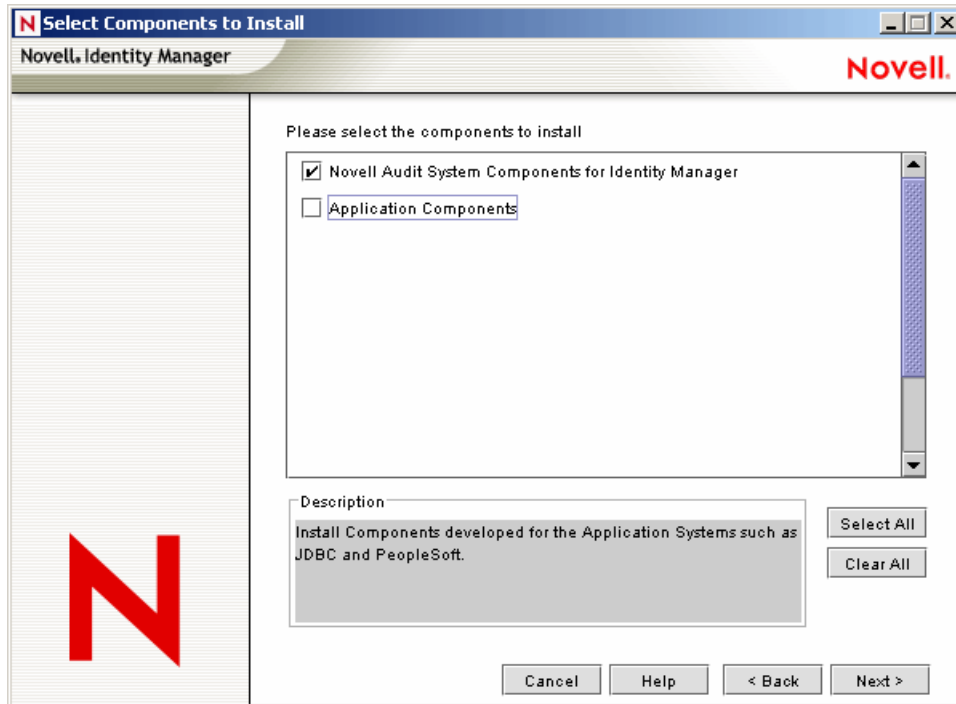
- ◆ Platform Agent
- ◆ Secure Logging Server

The Platform Agent is the component that runs with Identity Manager to communicate events to the Secure Logging Server. It is installed with Identity Manager. The Secure Logging Server is the component that receives event data from Identity Manager and other applications, and is installed separately from Identity Manager as part of Novell Audit 1.0.3.

10.3.1 Setting Up the Platform Agent

The Platform Agent is installed by selecting the Novell Audit System Components for the Identity Manager option during install.

Figure 10-2 Installation of Identity Manager



The Platform Agent can be installed during the installation Identity Manager, or it can be installed later.

NOTE: If you install the Platform Agent after the Metadirectory engine has been started, Identity Manager must be restarted before the Platform Agent and Identity Manager are linked. Identity Manager attempts to connect to the Platform Agent only during startup.

After the Platform Agent is installed, complete the following steps to configure it:

- 1 Open the Novell Audit configuration file, `logevent.cfg`, in a text editor. The default location for this file is:

Operating System	Path
NetWare®	<code>sys:\etc\logevent.cfg</code>
Windows	<code>windows_directory\logevent.cfg</code>
Linux\Solaris	<code>/etc/logevent.conf</code>

- 2 Change the value of the LogHost parameter to the IP address or DNS name of your Secure Logging Server.
- 3 Restart Identity Manager.

10.3.2 Setting Up the Secure Logging Server

NOTE: The Novell Audit Secure Logging Server is not included with Identity Manager. The Secure Logging Server is part of Novell Audit 1.0.3. For information on downloading Novell Audit 1.0.3, see the [Novell Audit Product Page \(http://www.novell.com/products/nsureaudit\)](http://www.novell.com/products/nsureaudit).

The Secure Logging Server runs on NetWare 5.1 or later, Windows* NT 4.0, Windows 2000 Server, Windows 2003 Server, Solaris* 8 or 9, and several versions of Linux*, including SUSE® Enterprise Linux Server 8 and SUSE 9.0.

The Secure Logging Server can log events to MySQL*, Oracle*, Microsoft* SQL Server, Java* Applications, and several other locations, including a flat file. Novell Audit includes a custom application designed to query databases for event data, called Novell Audit Report. A datastore that has an ODBC connector is required to use this advanced reporting tool.

A Quick Start Guide containing Secure Logging Server setup instructions is available for each platform, and is included with the Novell Audit 1.0.3 installation. The Quick Start guides are also viewable on the Web along with the *Novell Audit 1.0.3 Administration Guide* on the [Novell Audit Documentation Web site \(http://www.novell.com/documentation/nsureaudit\)](http://www.novell.com/documentation/nsureaudit).

10.4 Logging Configuration

Identity Manager enables you to configure the events that are logged by using several predefined levels, or by individually selecting each event you want to log. Changes to the configuration settings are also logged.

User-defined events, discussed in [Section 10.4.2, “User-Defined Events,” on page 235](#), are logged any time logging is enabled, and are never filtered by the Metadirectory engine.

Logging is configured on a driver set or on an individual driver. Drivers can inherit logging configuration from the driver set. For information on the eDirectory™ attributes containing log information, see [Section 10.4.3, “eDirectory Objects,” on page 237](#).

By default, only critical and user-defined events are logged.

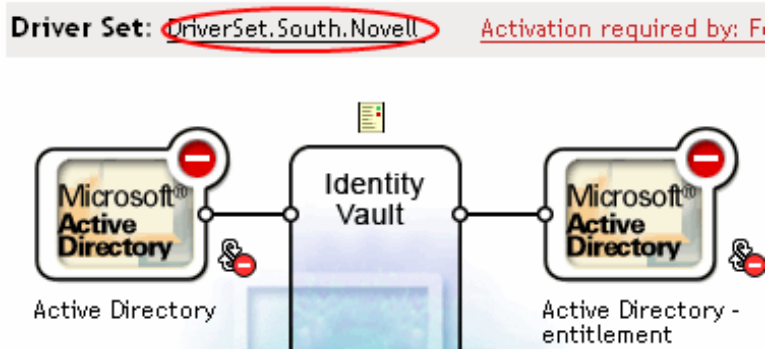
10.4.1 Selecting Events to Log

You can select events for a driver set or for a specific driver.

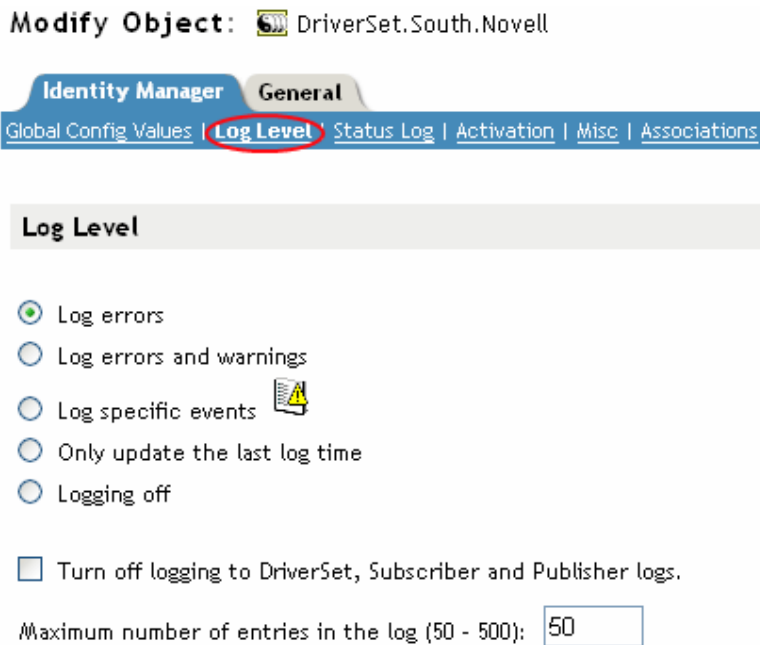
Logging Events for the Driver Set:

- 1 In iManager, select *Identity Manager > Identity Manager Overview*, then click *Next*.
- 2 Browse to and select the Driver Set object, then click *Search*.

3 Click the Driver Set name. The Modify Object page appears.




4 Select *Log Level* on the *Identity Manager* tab.



5 Select the logging option that is required for your environment.

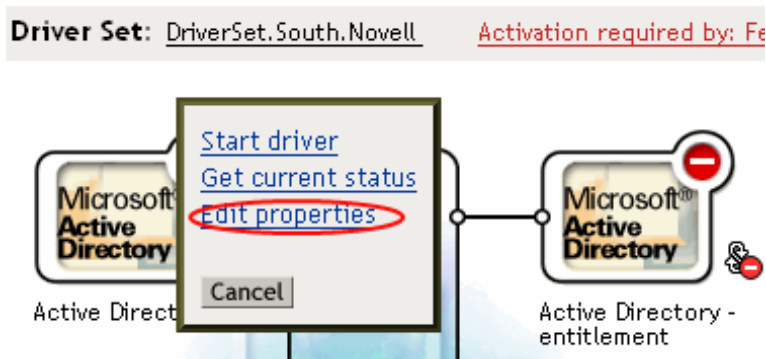
Option	Description
Log errors	<p>This is the default log level. This option logs all events with an error status, and also user-defined events.</p> <p>With this option selected, you receive only events with a decimal ID of 196646, with an error message stored in the first text field.</p>
Log errors and warnings	<p>This option logs all events with an error or warning status, and also user-defined events.</p> <p>With this option selected, you receive only events with a decimal ID of 196646 and 196647, with an error or warning message stored in the first text field.</p>

Option	Description
Log specific events	This option enables you to select specific events to log from a list. Click the  icon to select events. User-defined events are always logged. To log any event other than an error or warning, you must select it from this list. If you select this option, you must also select errors and warnings if you want to continue to log them. For a list of all available events, see "Identity Manager Events" on page 234 .
Only update the last log time	Only user-defined events are logged. When an event occurs, the last log time is updated, so you can view the time and date of the last error in the status log.
Logging off	Only user-defined events are logged.
Turn off logging to DriverSet, Subscriber and Publisher logs	Turns off logging to the Driver Set object log, and to the Subscriber and Publisher logs.
Maximum Number of Entries in the Log	This setting allows you to specify the maximum number of entries to log in the status logs. See Section 10.7.2, "Viewing Status Logs," on page 241 for details.

6 After you have selected the events you want to log, click *OK*.

Logging Events for the Driver:

- 1 In iManager select *Identity Manager > Identity Manager Overview*, then click *Next*.
- 2 Browse to and select the Driver Set object, then click *Search*.
- 3 Click the upper right corner of the driver icon, then select *Edit properties*.



4 Select *Log Level* on the *Identity Manager* tab.

Modify Object: Active Directory.DriverSet.South.Novell

Identity Manager | **Server Variables** | **General**

[Driver Configuration](#) | [Global Config Values](#) | [Named Passwords](#) | [Engine Control Values](#) | [Linkage](#) | **Log Level** | [Driver Image](#) | [Security Equals](#) | [Filter](#) | [Edit Filter XML](#) | [Misc](#) | [Excluded Users](#) | [Driver Manifest](#) | [Associations](#)

Log Level

Use log settings from the DriverSet, DriverSet.South.Novell
The following log settings are from the DriverSet and cannot be changed on this page. To modify the DriverSet's settings, [click here](#).

Log errors

Log errors and warnings

Log specific events

Only update the last log time

Logging off

Turn off logging to DriverSet, Subscriber and Publisher logs.


Maximum number of entries in the log (50 - 500):

5 (Optional) By default, the Driver object is configured to inherit log settings from the Driver Set object. To select logged events for this driver only, deselect use log settings from the Driver Set.

Use log settings from the DriverSet, DriverSet.South.Novell
The following log settings are from the DriverSet and cannot be changed on this page. To modify the DriverSet's settings, [click here](#).

6 Select the logging option that is required for your environment.

Option	Description
Log errors	This is the default log level. This option logs all events with an error status, and also user-defined events. With this option selected, you receive only events with a decimal ID of 196646, with an error message stored in the first text field.
Log errors and warnings	This option logs all events with an error or warning status, and also user-defined events. With this option selected, you receive only events with a decimal ID of 196646 and 196647, with an error or warning message stored in the first text field.

Option	Description
Log specific events	This option enables you to select specific events to log from a list. Click the  icon to select events. User-defined events are always logged. To log any event other than an error or warning, you must select it from this list. If you select this option, you must also select errors and warnings if you want to continue to log them. For a list of all available events, see “Identity Manager Events” on page 234 .
Only update the last log Time	Only user-defined events are logged. When an event occurs, the last log time is updated, so you can view the time and date of the last error in the status log.
Logging Off	Only user-defined events are logged.
Turn off logging to DriverSet, Subscriber and Publisher logs.	Turns off logging to the Driver Set object log, and to the Subscriber and Publisher logs.
Maximum Number of Entries in the Log	This setting allows you to specify the maximum number of entries to log in the status logs. See Section 10.7.2, “Viewing Status Logs,” on page 241 for details.

7 After you have selected the events you want to log, click *OK*.

Identity Manager Events

A listing of all events logged by Identity Manager is contained in the [Appendix C, “Identity Manager Events and Reports,” on page 263](#).

Driver Start and Stop Events

Identity Manager can generate an event whenever a driver starts or stops. The following table contains details about these events:

Table 10-1 *Driver Start and Stop Events*

Event	Log Level	Information
EV_LOG_DRIVER_START	LOG_INFO	To log driver starts, you must use the <i>Log Specific Events</i> option and select this event.
EV_LOG_DRIVER_STOP	LOG_WARNING	To log driver stops, select <i>Log Errors and Warnings</i> , or use the <i>Log Specific Events</i> option and select this event.

For details on creating Novell Audit notifications based on these events, see [Section 10.6, “Sending Notifications Based on Events,” on page 239](#).

Error and Warning Events

Identity Manager generates an event whenever an error or warning is encountered. The following table contains details on these events:

Table 10-2 *Error and Warning Events*

Event	Log Level	Information
DirXML_Error	LOG_ERROR	All Identity Manager errors log this event. The actual error code encountered is stored in the event. To log errors, select <i>Log Errors</i> , <i>Log Errors and Warnings</i> , or use the <i>Log Specific Events</i> option and select this event.
DirXML_Warning	LOG_WARNING	All Identity Manager warnings log this event. The actual warning code encountered is stored in the event. To log warnings, select <i>Log Errors and Warnings</i> , or use the <i>Log Specific Events</i> option and select this event.

For details on creating Novell Audit notifications based on these events, see [Section 10.6, “Sending Notifications Based on Events,”](#) on page 239.

Remote Loader Events

The following events are logged from the remote loader:

Table 10-3 *Remote Loader Events*

Event	Log Level	Information
Remote Loader Start	LOG_INFO	To log when the Remote Loader starts, you must use the <i>Log Specific Events</i> option and select this event.
Remote Loader Stop	LOG_INFO	To log when the Remote Loader stops, you must use the <i>Log Specific Events</i> option and select this event.
Remote Loader Connection Established	LOG_INFO	To log when the Remote Loader connections are established, you must use the <i>Log Specific Events</i> option and select this event.
Remote Loader Connection Dropped	LOG_INFO	To log when the Remote Loader connections are dropped, you must use the <i>Log Specific Events</i> option and select this event.

For details on creating Novell Audit notifications based on these events, see [Section 10.6, “Sending Notifications Based on Events,”](#) on page 239.

10.4.2 User-Defined Events

Identity Manager enables you to configure your own events to log to Novell Audit. Events can be logged using an action in Policy Builder, or within a style sheet. Any information you have access to when defining policies can be logged.

Event IDs

Event IDs between 1000 and 1999 are allotted for user-defined events. You must specify a value within this range for the event ID when defining your own events. In Novell Audit, this ID is combined with the Identity Manager application ID of 003.

Log Levels


Log levels enable you to group events based on the type of event being logged. The following predefined log levels are available:

Table 10-4 Log Levels

Log Level	Description
log-emergency	Events that cause the Metadirectory engine or driver to shut down.
log-alert	Events that require immediate attention.
log-critical	Events that can cause parts of the Metadirectory engine or driver to malfunction.
log-error	Events describing errors that can be handled by the Metadirectory engine or driver.
log-warning	Negative events not representing a problem.
log-notice	Positive or negative events an administrator can use to understand or improve use and operation.
log-info	Positive events of any importance.
log-debug	Events of relevance for support or for engineers to debug operation of the Metadirectory engine or driver.

Generating Events Using Policy Builder

In Policy Builder, events are logged by selecting the *Generate Event* action.

- 1 Select the condition to be met before the event is generated, then select the *Generate Event* action.
- 2 Specify an **event ID**.
- 3 Select a **log level**.
- 4 Click the  icon next to the *Enter Strings* field to launch the Named String Builder.
- 5 Use the Named String Builder to construct named strings corresponding to the custom data fields:



Strings	
<input type="checkbox"/> Name: * text1	String tokens: * Operation Attribute("Given Name") 
<input type="checkbox"/> Name: * text2	String tokens: * Operation() 
<input type="checkbox"/> Name: * value	String tokens: * "1000" 

- 6 Click *OK* to return to the Policy Builder to construct the remainder of your policy.

See “**Generate Event**” in the *Policy Builder and Driver Customization Guide* for information on how to configure a policy to log events.

Generating Events Using Status Documents

Status documents generated through style sheets using the `<xsl:message>` element are sent to Novell Audit with an event ID corresponding to the level attribute of the status document as specified in the following table:

Table 10-5 *Status Documents*

Status Level	Status Event ID
Success	EV_LOG_STATUS_SUCCESS (1)
Retry	EV_LOG_STATUS_RETRY (2)
Warning	EV_LOG_STATUS_WARNING (3)
Error	EV_LOG_STATUS_ERROR (4)
Fatal	EV_LOG_STATUS_FATAL (5)
User Defined	EV_LOG_STATUS_OTHER (6)

The following example generates an Novell Audit event 0x004 and value1=7777, with a level of EV_LOG_STATUS_ERROR:

```
<xsl:message>
  <status level="error" text1="This would be text1" value="7777">This
data would be in the blob and in text 2, since no value is specified
for text2 in the attributes.</status>
</xsl:message>
```

The following example generates an Novell Audit event 0x004 and value1=7778, with a level of EV_LOG_STATUS_ERROR:

```
<xsl:message>
  <status level="error" text1="This would be text1" text2="This would
be text2" value1="7778">This data would be in the blob only for this
case, since a value for text2 is specified in the attributes.</status>
</xsl:message>
```

10.4.3 eDirectory Objects

This section provides details on the Novell eDirectory attributes that store log data. You do not need to modify these attributes directly, because these objects are automatically configured based on your selections in iManager.

The Identity Manager events you want to log are stored in the DirXML-LogEvent attribute on the Driver Set object or Driver object. The attribute is a multivalue integer with each value identifying an event ID to be logged.

Before logging an event, the engine checks the current event type against the contents of this attribute to determine whether the event should be logged.

Previous versions of Identity Manager used the `DirXML-DriverTraceLevel` attribute to set up logging levels. The logging level was specified on each Driver object, and did not support inheritance. In versions after Identity Manager 2, Driver objects can inherit this information from the Driver Set object. The `DirXML-DriverTraceLevel` attribute of a driver object has the highest precedence when determining log settings. If a Driver object does not contain a `DirXML-DriverTraceLevel` attribute, the engine uses the log settings from the parent driver set object.

10.5 Querying and Reporting

Novell Audit provides two tools to query for events in the Novell Audit database: The Novell Audit iManager plug-in, and Novell Audit Report (LReport).

The Novell Audit iManager plug-in is a Web-based JDBC database querying application that enables you to quickly create and store queries using drop-down lists and macros.

Novell Audit Report is a Windows-based ODBC-compliant application that can use SQL query statements or Crystal Decisions Reports to query Oracle and MySQL data stores (or any other database that has ODBC driver support).

Follow the instructions in the *Novell Audit Administration Guide* to access the Novell Audit iManager plug-in, or to set up Novell Audit Report. This guide is available on the [Novell Audit Documentation Web site](http://www.novell.com/documentation/nsureaudit) (<http://www.novell.com/documentation/nsureaudit>).

10.5.1 Identity Manager Reports

Identity Manager provides a number of Crystal Decisions Reports (*.rpt) that simplify gathering information on common operations performed in Identity Manager. These reports are included on the Identity Manager installation CD.

After you have configured Novell Audit Report, these reports, along with any custom queries and reports you have defined, can be executed. See [Working with Reports in Novell Audit Report](http://www.novell.com/documentation/nsureaudit/nsureaudit/data/alsn2fj.html) (<http://www.novell.com/documentation/nsureaudit/nsureaudit/data/alsn2fj.html>) in the *Novell Audit 1.0.3 Administration Guide* for information on using these reports in Novell Audit Report. For examples of these reports see [Section C.11, “Reports,” on page 282](#) in the [Appendix C, “Identity Manager Events and Reports,” on page 263](#).

10.5.2 Viewing Identity Manager Events

- 1 In the Novell Audit Report Workspace, click the *Events* tab, then expand the *DirXML* folder.

This list contains all predefined Identity Manager events. Double-click any event in the list to view event properties.

- 2 To query for an Identity Manager event, right-click the event in the Workspace, then select *Define Query*.
- 3 When the Query Expert appears, specify a time frame and verify the event.
- 4 To run this query, select the *Query* tab in the Workspace, right-click the query name, then select *Run*.

Queries can also be created using SQL statements. All Identity Manager events have a decimal Event ID between 109608 and 262144.

10.6 Sending Notifications Based on Events

Novell Audit provides the ability to send a notification when a specific event occurs or does not occur. Notifications can be sent based on one or more events and any values contained within these events. Notifications can be sent to any logging channel, enabling you to log notifications to a database, a Java application or SNMP management system, or several other locations.

For information on creating notifications, see “[Configuring Filters and Event Notifications](http://www.novell.com/documentation/nsureaudit/nsureaudit/data/al0lg08.html#al0lg08)” in the *Novell Audit 1.0.3 Administration Guide* (<http://www.novell.com/documentation/nsureaudit/nsureaudit/data/al0lg08.html#al0lg08>)

10.7 Using Status Logs

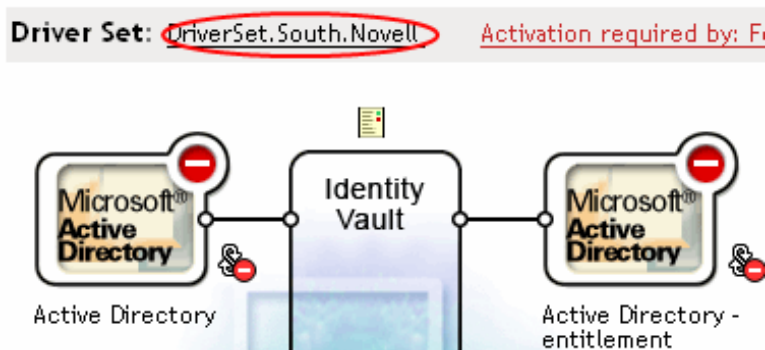
In addition to the functionality provided by Novell Audit, Identity Manager logs a specified number of events on the Driver Set object and Driver object. These status logs provide a view of recent Identity Manager activity. After the log reaches the set size, the oldest half of the log is permanently removed to clear room for more recent events. Therefore, any events you want to track over time should be logged to Novell Audit or the Report and Notification Service.

10.7.1 Setting the Maximum Log Size

Status logs can be configured to hold between 50 and 500 events. This setting can be configured on the Driver Set object to be inherited by all drivers in the set, or configured for each driver in the set. The maximum log size operates independently of the events you have selected to log, so you can configure the events you want to log on the Driver Set, then specify a different log size for each driver in the set.

Setting the Log Size on the Driver Set

- 1 In iManager, select *Identity Manager > Identity Manager Overview*, then click *Next*.
- 2 Browse to and select the Driver Set object, then click *Search*.
- 3 Click the Driver Set name. The Modify Object window appears.




- 4 Select *Log Level* on the *Identity Manager* tab.

Modify Object:  DriverSet.South.Novell



Log Level

- Log errors
 - Log errors and warnings
 - Log specific events 
 - Only update the last log time
 - Logging off
- Turn off logging to DriverSet, Subscriber and Publisher logs.
- Maximum number of entries in the log (50 - 500):

- 5 Specify the maximum log size in the *Maximum number of entries in the log* field:

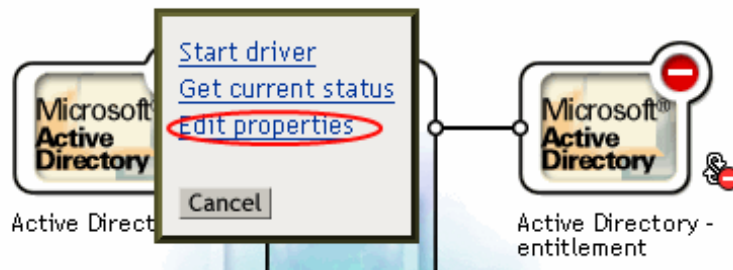
Maximum number of entries in the log (50 - 500):

- 6 After you have specified the maximum number, click *OK*.

Setting the Log Size on the Driver

- 1 In iManager select *Identity Manager > Identity Manager Overview*, then click *Next*.
- 2 Browse to and select the Driver Set object, then click *Search*.
- 3 Click the upper right corner of the driver icon, then select *Edit properties*.

Driver Set: DriverSet.South.Novell Activation required by: Fe



4 Select *Log Level* on the *Identity Manager* tab.

Modify Object: Active Directory.DriverSet.South.Novell

Identity Manager | Server Variables | General

Driver Configuration | Global Config Values | Named Passwords | Engine Control Values | Linkage | **Log Level** | Driver Image | Security Equals | Filter | Edit Filter XML | Misc | Excluded Users | Driver Manifest | Associations

Log Level

Use log settings from the DriverSet, DriverSet.South.Novell
The following log settings are from the DriverSet and cannot be changed on this page. To modify the DriverSet's settings, [click here](#).

Log errors
 Log errors and warnings
 Log specific events
 Only update the last log time
 Logging off

Turn off logging to DriverSet, Subscriber and Publisher logs.

Maximum number of entries in the log (50 - 500):

5 Specify the maximum log size in the *Maximum number of entries in the log* field:

Maximum number of entries in the log (50 - 500):

6 After you have specified the maximum number, click *OK*.

10.7.2 Viewing Status Logs

Status log entries are represented in iManager with a status log icon . Anywhere you see this icon in iManager, you can view a short-term log. The following status logs are available:

- ♦ On the driver set.
- ♦ On the Publisher channel for each driver in the set.
- ♦ On the Subscriber channel for each driver in the set.

The status logs for the Publisher and Subscriber channels report channel-specific messages generated by the driver, such as an operation veto for an unassociated object.

The status log for the driver set contains only messages generated by the engine, such as state changes for any drivers in the driver set. All engine messages are logged.

DirXML Command Line Utility

A

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 243](#)
- ♦ [Section A.2, “Command Line Mode,” on page 251](#)

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.
Example: `admin.novell`
- 3 Enter the user’s password specified above.
Example: `novell`

```
DirXML commands
1: Start driver
2: Stop driver
3: Driver operations...
4: Driver set operations...
5: Log events operations...
6: Get DirXML version
99: Quit
Enter choice: █
```

- 4 Enter the number of the command you would like to perform.
The [Table A-1 on page 244](#) 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 errors 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: <i>Start Driver</i>	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: <i>Stop Driver</i>	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: <i>Driver operations</i>	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 244 for the operations available.
4: <i>Driver set operations</i>	Lists the operations available for the driver set. <ul style="list-style-type: none"> ◆ 1: Associate driver set with server ◆ 2: Disassociate driver set from server ◆ 99: Exit
5: <i>Log events operations</i>	Lists the operations available for logging events through Novell Audit. See Table A-5 on page 249 for a description of these options.
6: <i>Get DirXML version</i>	Lists the version of the Identity Manager installed.
99: <i>Quit</i>	Exits the DirXML Command Line Utility

Figure A-1 *Driver Options*

```

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: Check object password
10: Initialize new driver object
11: Passwords operations
12: Cache operations
99: Exit
Enter choice: █

```

Table A-2 *Driver Options*

Options	Description
1: <i>Start driver</i>	Starts driver.

Options	Description
2: <i>Stop driver</i>	Stops driver.
3: <i>Get driver state</i>	Lists the state of the driver. <ul style="list-style-type: none"> ◆ 0 - Driver is stopped ◆ 1 - Driver is starting ◆ 2 - Driver is running ◆ 3 - Driver is stopping
4: <i>Get driver start option</i>	Lists the current driver start option. <ul style="list-style-type: none"> ◆ 1 - Disabled ◆ 2 - Manual ◆ 3 - Auto
5: <i>Set driver start option</i>	Changes the start option of the driver. <ul style="list-style-type: none"> ◆ 1 - Disabled ◆ 2 - Manual ◆ 3 - Auto ◆ 99 - Exit
6: <i>Resync driver</i>	<p>Forces a resynchronization the driver. It prompts for a time delay: <i>Do you want to specify a minimum time for resync? (yes/no)</i>.</p> <p>If you enter yes, specify the date and time the resynchronization occurs: <i>Enter a date/time (format 9/27/05 3:27 PM)</i>.</p> <p>If you enter no, the resynchronization occurs immediately.</p>
7: <i>Migrate from application into DirXML</i>	<p>Processes an XML document that contains a query command: <i>Enter filename of XDS query document:</i></p> <p>Create the XML document that contains a query command by using the Novell nds.dtd (http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/ref/ndsstdtd/query.html).</p> <p>Examples:</p> <p>NetWare: <code>sys:\files\query.xml</code></p> <p>Windows: <code>c:\files\query.xml</code></p> <p>Linux: <code>/files/query.xml</code></p>

Options	Description
8: <i>Submit XDS command document to driver</i>	<p>Processes an XDS command document:</p> <p><i>Enter filename of XDS command document:</i></p> <p>Examples:</p> <p>NetWare: sys:\files\user.xml</p> <p>Windows: c:\files\user.xml</p> <p>Linux: /files/user.xml</p> <p><i>Enter name of file for response:</i></p> <p>Examples:</p> <p>NetWare: sys:\files\user.log</p> <p>Windows: c:\files\user.log</p> <p>Linux: /files/user.log</p>
9: <i>Check object password</i>	<p>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).</p> <p>Enter user name:</p>
10: <i>Initialize new driver object</i>	<p>Performs an internal initialization of data on a new Driver object. This is only for testing purposes.</p>
11: <i>Password operations</i>	<p>There are nine Password options. See Table A-3 on page 247 for a description of these options.</p>
12: <i>Cache operations</i>	<p>There are five Cache operations. See Table A-4 on page 248 for a descriptions of these options.</p>
99: <i>Exit</i>	<p>Exits the driver options.</p>

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: <i>Set shim password</i>	Sets the application password. This is the password of the user account you are using to authenticate into the connected system with.
2: <i>Clear shim password</i>	Clears the application password.
3: <i>Set Remote Loader password</i>	<p>The Remote Loader password is used to control access to the Remote Loader instance. See Chapter 3, “Setting Up a Connected System,” on page 45 for more information.</p> <p>Enter Remote Loader password, then confirm the password by typing it again.</p>
4: <i>Clear Remote Loader password</i>	Clears the Remote Loader password so no Remote Loader password is set on the Driver object.
5: <i>Set named password</i>	<p>Allows you to store a password or other pieces of information security on the driver. See Section 2.9, “Using Named Passwords,” on page 31 for more information.</p> <p>There are four prompts to fill in:</p> <ul style="list-style-type: none">◆ Enter password name:◆ Enter password description:◆ Enter password:◆ Confirm password
6: <i>Clear named passwords</i>	<p>Clears a specified named password or all named passwords that are stored on the driver object: <i>Do you want to clear all named passwords? (yes/no)</i>.</p> <p>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.</p>
7: <i>List named passwords</i>	Lists all named passwords that are stored on the driver object. It lists the password name and the password description.
8: <i>Get password state</i>	<p>Lists if a password is set for:</p> <ul style="list-style-type: none">◆ Driver Object password:◆ Application password:◆ Remote loader password: <p>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 it has been set or not.</p>
99: <i>Exit</i>	Exits the current menu and takes you back to Driver options.

Figure A-3 Cache Operations

```
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: <i>Get driver cache limit</i>	Displays the current cache limit that is set for the driver.
2: <i>Set driver cache limit</i>	Sets the driver cache limit in kilobytes. A value of 0 is unlimited.
3: <i>View cached transactions</i>	A text file is created with the events that are stored in cache. You can select the number of transactions to view. <ul style="list-style-type: none">◆ Enter option token (default=0):◆ Enter maximum transactions records to return (default=1):◆ Enter name of file for response:
4: <i>Delete cached transactions</i>	Deletes the transactions stored in cache. <ul style="list-style-type: none">◆ 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: <i>Exit</i>	Exits the current menu and takes you back to 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: <i>Set driver set log events</i>	Allows you to log driver set events through Novell Audit. There are 49 items to select to log. See Table A-6 on page 249 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: <i>Reset driver set log events</i>	Resets all of the log event options.
3: <i>Set driver log events</i>	Allows you to log driver events through Novell Audit. There are 49 items to select to log. See Table A-6 on page 249 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: <i>Reset driver log events</i>	Resets all of the log event options.
99: <i>Exit</i>	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

Options

- 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
 - 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: Unknown
 - 47: Unknown
 - 48: User defined IDs
-

Options

99: Accept checked items

A.2 Command Line Mode

The command line mode allows you to use script or batch files. The [Table A-7 on page 251](#) 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`

The command starts the driver.

Table A-7 *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.
-? <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 if 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.

Option	Description
-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 <code>nds.dtd</code> (http://developer.novell.com/ndk/doc/dirxml/dirxmlbk/ref/ndsstd/query.html).
-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. See Chapter 3, "Setting Up a Connected System," on page 45 for more information.
<clearremoteloaderpassword <driver dn>	Clears the Remote Loader password.
-sendcommand <driver dn> <input filename> <output filename>	Processes an XDS command document. Specify the XDS command document as the input file. Examples: NetWare: <code>sys:\files\user.xml</code> Windows: <code>c:\files\user.xml</code> Linux: <code>/files/user.log</code> Specify the output file name to see the results. Examples: NetWare: <code>sys:\files\user.log</code> Windows: <code>c:\files\user.log</code> Linux: <code>/files/user.log</code>
-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 249 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 installed.
-initdriver object <dn>	Performs an internal initialization of data on a new Driver object. This is only for testing purposes.

Option	Description
-setnamedpassword < <i>driver dn</i> > < <i>name</i> > < <i>password</i> > [<i>description</i>]	Sets named passwords on the driver object. You specify the name, the password and the description of the named password.
-clearnamedpassword < <i>driver dn</i> > < <i>name</i> >	Clears a specified named password.
-clearallnamedpasswords < <i>driver dn</i> >	Clears all named passwords set on a specific driver.

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	<p>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.</p> <p>You have three options: address=<i>address number</i> address='localhost' Don't use this parameter.</p> <p>If you don't use the -address, the Remote Loader listens on all local IP addresses.</p> <p>Example: address=137.65.134.83</p>
-class	-cl	Java class name	<p>Specifies the Java class name of the Identity Manager application shim that is to be hosted.</p> <p>For example, for a Java driver, type one of the following:</p> <pre>-class com.novell.nds.dirxml.driver.Idap.LDAPDriverShim -cl com.novell.nds.dirxml.driver.Idap.LDAPDriverShim</pre> <p>Java uses a keystore to read certificates. The -class option and the -module option are mutually exclusive.</p> <p>To see a list of the Java class name see Table B-2 on page 262.</p>

Option	Secondary Name	Parameter	Description
-commandport	-cp	port number	<p>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.</p> <p>Example:</p> <pre>-commandport 8001 -cp 8001</pre>
-config	None	filename	<p>Specifies a configuration file. The configuration file can contain any command line options except <code>config</code>. Options specified on the command line override options specified in the configuration file.</p> <p>Example:</p> <pre>-config config.txt</pre>
-connection	-conn	connection configuration string	<p>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.</p> <p>Example:</p> <pre>-connection "port=8091 rootfile=server1.pem" -conn "port=8091 rootfile=server1.pem"</pre>

Option	Secondary Name	Parameter	Description
-description	-desc	short description	<p>Specify a short description string (for example, SAP) to be used for the trace window title and for Novell® Audit logging.</p> <p>Example:</p> <pre>-description SAP -desc SAP</pre> <p>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).</p>
-help	-?	None	<p>Displays help.</p> <p>Example:</p> <pre>-help -?</pre>
-java	-j	None	<p>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.</p>
-javadebugport	-jdp	Port number	<p>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.</p> <p>Example:</p> <pre>-javadebugport 8080 -jdp 8080</pre>
keystore			<p>Conditional parameters. Used only for Identity Manager application shims contained in .jar files.</p> <p>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.</p> <p>If you are running SSL and need the Remote Loader to communicate with a Java driver, type a key-value pair:</p> <pre>keystore='keystorename' storepass='password'</pre>

Option	Secondary Name	Parameter	Description
-module	-m	modulename	<p>Specifies the module containing the Identity Manager application shim that is to be hosted.</p> <p>For example, for a native driver, type one of the following:</p> <pre>-module "c:\Novell\RemoteLoader\Exchange5Shim.dll" -m "c:\Novell\RemoteLoader\Exchange5Shim.dll"</pre> <p>or</p> <pre>-module "usr/lib/dirxml/NISDriverShim.so" -m "usr/lib/dirxml/NISDriverShim.so"</pre> <p>The -module option uses a rootfile certificate. The -module option and the -class option are mutually exclusive.</p>
-password	-p	password	<p>Specifies the password for command authentication. This password must be the same as the first password specified with <code>setpasswords</code> for the loader instance being commanded. If a command option (for example, <code>unload</code> or <code>tracechange</code>) is specified and the <code>password</code> option isn't specified, the user is prompted to enter the password for the loader that is the target of the command.</p> <p>Example:</p> <pre>-password novell4 -p novell4</pre>
port		decimal port number	<p>A required parameter. It specifies the TCP/IP port on which the Remote Loader listens for connections from the remote interface shim.</p> <p>Example:</p> <pre>port=8090</pre>
rootfile			<p>A conditional parameter. If you are running SSL and need the Remote Loader to communicate with a native driver, type</p> <pre>rootfile='trusted certname'</pre>

Option	Secondary Name	Parameter	Description
-service	-serv	None, or install/ uninstall	<p>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.</p> <p>This option installs the Win32 service but doesn't start the service.</p> <p>To uninstall an instance running as a service, use the uninstall argument together with any other arguments necessary to host the application shim.</p> <p>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.</p> <p>Example:</p> <pre>-service install -serv uninstall</pre> <p>This option isn't available on rdxml or the Java Remote Loader.</p>
-setpasswords	-sp	password password	<p>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.</p> <p>Example:</p> <pre>-setpasswords novell4 staccato3 -sp novell4 staccato3</pre>
-storepass		storepass	<p>Used only for Identity Manager application shims contained in .jar files. Specifies the password for the Java keystore specified by the keystore parameter.</p> <p>Example:</p> <pre>storepass=mypassword</pre> <p>This option applies only to the Java Remote Loader.</p>

Option	Secondary Name	Parameter	Description
-trace	-t	integer	<p>Specifies the trace level. This is only used when hosting an application shim. Trace levels correspond to those used on the metadirectory server.</p> <p>Example:</p> <pre>-trace 3 -t 3</pre>
-tracechange	-tc	integer	<p>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.</p> <p>Example:</p> <pre>-tracechange 1 -tc 1</pre>
-tracefile	-tf	filename	<p>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.</p> <p>Example:</p> <pre>-tracefile c:\temp\trace.txt -tf c:\temp\trace.txt</pre>
-tracefilechange	-ffc	None, or filename	<p>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.</p> <p>Example:</p> <pre>-tracefilechange c:\temp\newtrace.txt ffc c:\temp\newtrace.txt</pre>

Option	Secondary Name	Parameter	Description
-tracefilemax	-tfm	size	<p>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.</p> <p>The size parameter is the number of bytes. Specify the size by using the suffixes K, M, or G for kilobytes, megabytes, or gigabytes.</p> <p>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</p> <p>Example:</p> <pre>-tracefilemax 1000M -tfm 1000M</pre> <p>In this example, the trace file can be only 1 GB.</p>
-unload	-u	None	<p>Unloads the Remote Loader instance. If the Remote Loader is running as a Win32 Service, this command stops the service.</p> <p>Example:</p> <pre>-unload -u</pre>
-window	-w	On/Off	<p>Turns the trace window on or off in a Remote Loader instance.</p> <p>Example:</p> <pre>-window on -w off</pre> <p>This option is available only on Windows platforms. It isn't available on the Java Remote Loader.</p>

Option	Secondary Name	Parameter	Description
-wizard	-wiz	None	<p>Launches the Configuration Wizard. Running <code>dirxml_remote.exe</code> 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.</p> <p>Example:</p> <p>-wizard</p> <p>-wiz</p> <p>This option is available only on Windows platforms. It isn't available on the Java Remote Loader.</p>

Table B-2 *Java Class Names*

Java Class Name	Driver
<code>com.novell.nds.dirxml.driver.avaya.PBXDriverShim</code>	Avaya PBX Driver
<code>com.novell.nds.dirxml.driver.delimitedtext.DelimitedTextDriver</code>	Delimited Text Driver
<code>com.novell.nds.dirxml.driver.nds.DriverShimImpl</code>	eDirectory Driver
<code>com.novell.nds.dirxml.driver.entitlement.EntitlementServiceDriver</code>	Entitlement Services Driver
<code>com.novell.gw.dirxml.driver.gw.GWdriverShim</code>	GroupWise Driver
<code>com.novell.nds.dirxml.driver.jdbc.JDBCdriverShim</code>	JDBC Driver
<code>com.novell.nds.dirxml.driver.ldap.LDAPDriverShim</code>	LDAP Driver
<code>com.novell.nds.dirxml.driver.loopback.LoopbackDriverShim</code>	Loopback Driver
<code>com.novell.nds.dirxml.driver.manualtask.driver.ManualTaskDriver</code>	Manual Task Driver
<code>com.novell.nds.dirxml.driver.nisdriver.NISDriverShim</code>	NIS Driver
<code>com.novell.nds.dirxml.driver.notes.NotesDriverShim</code>	Notes Driver
<code>com.novell.nds.dirxml.driver.psoftshim.PSOFTDriverShim</code>	PeopleSoft Driver
<code>com.novell.nds.dirxml.driver.SAPShim.SAPDriverShim</code>	SAP HR Driver
<code>com.novell.nds.dirxml.driver.sapusershim.SAPDriverShim</code>	SAP User Management Driver
<code>com.novell.nds.dirxml.driver.sifagent.SIFShim</code>	SIF Driver
<code>com.novell.nds.dirxml.driver.soap.SOAPDriver</code>	Soap Driver
<code>com.novell.idm.driver.ComposerDriverShim</code>	User Application
<code>be.opns.dirxml.driver.ars.arsremedydrivershim.ARSDriverShim</code>	Driver for Remedy ARS

Identity Manager Events and Reports

The sections contains a listing of all Novell® Audit events logged by Identity Manager. It also contains examples of the reports that can be run with Novel Audit. [Section C.11, “Reports,” on page 282](#) contains the examples of the reports.

Each event has the follow information stored on it: EventID, Description, Originator Title, Target Title, Subtarget Title, Text1 Title, Text2 Title, Text3 Title, Value1 Title, Value1 Type, Value2 Title, Value2 Type, Value3 Title, Value3 Type, Group Title, Group Type, Data Title, Data Type, Display Schema.

The events for the following components are in the tables.

- ♦ [Section C.1, “Engine Events,” on page 263](#)
- ♦ [Section C.2, “Server Events,” on page 272](#)
- ♦ [Section C.3, “Remote Loader Events,” on page 274](#)
- ♦ [Section C.4, “Detail Portlets,” on page 275](#)
- ♦ [Section C.5, “Change Password Portlet,” on page 275](#)
- ♦ [Section C.6, “Forgot Password Change Password Portlet,” on page 276](#)
- ♦ [Section C.7, “Search List Portlet,” on page 276](#)
- ♦ [Section C.8, “Create Portlet,” on page 277](#)
- ♦ [Section C.9, “Security Context,” on page 277](#)
- ♦ [Section C.10, “Workflow,” on page 279](#)
- ♦ [Section C.11, “Reports,” on page 282](#)

C.1 Engine Events

The tables contains the list of engine events that can be audited through Novell Audit.

Table C-1 *Engine Events Fields: Originator Title, Target Title, and Subtarget Title*

EventID	Description	Originator Title	Target Title	Subtarget Title
30001	Status Success	Channel	src-dn (dest-dn)	Level
30002	Status Retry	Channel	src-dn (dest-dn)	Level
30003	Status Warning	Channel	src-dn (dest-dn)	Level
30004	Status Error	Channel	src-dn (dest-dn)	Level
30005	Status Fatal	Channel	src-dn (dest-dn)	Level
30006	Status Other	Channel	src-dn (dest-dn)	Level

EventID	Description	Originator Title	Target Title	Subtarget Title
30007	Search	Channel	dest-dn or association	Scope
30008	Add Entry	Channel	dest-dn or association	Attribute name
30009	Delete Entry	Channel	dest-dn or association	Attribute name
3000A	Modify Entry	Channel	dest-dn or association	Attribute name
3000B	Rename Entry	Channel	dest-dn or association	Object type
3000C	Move Entry	Channel	dest-dn or association	Move Destination
3000D	Add Association	Channel	dest-dn	Attribute name
3000E	Remove Association	Channel		Attribute name
3000F	Query Schema	Channel		
30010	Check Password	Channel	Driver	
30011	Check Object Password	Channel	dest-dn or association	
30012	Change Password	Channel	dest-dn or association	
30013	Sync	Channel	dest-dn or association	Attribute name
30014	Input XML Document	Channel		Attribute name
30015	Input Transformation Document	Channel		
30016	Output Transformation Document	Channel		
30017	Event Transformation Document	Channel		
30018	Placement Rule Transformation Document	Channel		
30019	Create Rule Transformation Document	Channel		
3001A	Input Mapping Rule Transformation Document	Channel		
3001B	Output Mapping Rule Transformation Document	Channel		

EventID	Description	Originator Title	Target Title	Subtarget Title
3001C	Matching Rule Transformation Document	Channel		
3001D	Command Transformation Document	Channel		
3001E	Publisher Filter Transformation Document	Channel		
3001F	User Agent Request	Channel		
30020	Resync Driver	Channel	Driver	
30021	Migrate	Channel	Association	Attribute name
30022	Driver Start	Driver Set	Driver	
30023	Driver Stop	Driver Stop	Driver	
30024	Password Sync	Channel	Object	Attribute name
30025	Password Reset	Channel	dest-dn or association	Attribute name
30026	DirXML Error	Channel	Object	
30027	DirXML Warning	Channel	Object	
30028	Custom Operation	Channel		
30029	Clear Attribute	Channel	dest-dn or association	Attribute name
3002A	Add Value - Modify Entry	Channel	dest-dn or association	Attribute name
3002B	Remove Value	Channel	dest-dn or association	Attribute name
3002C	Merge Entries	Channel	Object	Attribute name
3002D	Get Named Password	Driver or Channel	Object	
3002E	Reset Attributes	Channel	Object	Channel
3002F	Add Value - Add Entry	Channel	dest-dn or association	Attribute name

Table C-2 Engine Events Fields: Text1 Title, Text2 Title, and Text3 Title

EventID	Description	Text1 Title	Text2 Title	Text3 Title
30001	Status Success	Type	Status Document	Event ID
30002	Status Retry	Type	Status Document	Event ID
30003	Status Warning	Type	Status Document	Event ID

EventID	Description	Text1 Title	Text2 Title	Text3 Title
30004	Status Error	Type	Status Document	Event ID
30005	Status Fatal	Type	Status Document	Event ID
30006	Status Other	Type	Status Document	Event ID
30007	Search	Object type		Event ID
30008	Add Entry	Object type	src-dn	Event ID
30009	Delete Entry	Object type	src-dn	Event ID
3000A	Modify Entry	Object type	src-dn	Event ID
3000B	Rename Entry	New name	src-dn	Event ID
3000C	Move Entry	Move Association	src-dn	Event ID
3000D	Add Association	Association		Event ID
3000E	Remove Association	Association		Event ID
3000F	Query Schema			Event ID
30010	Check Password			
30011	Check Object Password			Event ID
30012	Change Password	Object type	src-dn	Event ID
30013	Sync	Object type	association	Type
30014	Input XML Document			Warning message
30015	Input Transformation Document			Warning message
30016	Output Transformation Document			Warning message
30017	Event Transformation Document			Warning message
30018	Placement Rule Transformation Document			Warning message
30019	Create Rule Transformation Document			Warning message
3001A	Input Mapping Rule Transformation Document			Warning message
3001B	Output Mapping Rule Transformation Document			Warning message
3001C	Matching Rule Transformation Document			Warning message

EventID	Description	Text1 Title	Text2 Title	Text3 Title
3001D	Command Transformation Document			Warning message
3001E	Publisher Filter Transformation Document			Warning message
3001F	User Agent Request			
30020	Resync Driver			Error message
30021	Migrate	Object type		Warning message
30022	Driver Start			Driver message
30023	Driver Stop			Driver message
30024	Password Sync			
30025	Password Reset		src-dn	
30026	DirXML Error	Error Message		
30027	DirXML Warning	Warning Message		
30028	Custom Operation			
30029	Clear Attribute		src-dn	Event ID
3002A	Add Value - Modify Entry	Value	src-dn	Event ID
3002B	Remove Value	Value	src-dn	Event ID
3002C	Merge Entries	Object type	Channel	Association
3002D	Get Named Password	Password Name		Event ID
3002E	Reset Attributes			
3002F	Add Value - Add Entry	Value	src-dn	Event ID

Table C-3 Engine Events Fields: Value1 Title, Value2 Title, and Value3 Title

EventID	Description	Value1 Title	Value2 Title	Value3 Title
30001	Status Success			
30002	Status Retry			
30003	Status Warning			
30004	Status Error			
30005	Status Fatal			
30006	Status Other			
30007	Search			Result
30008	Add Entry			Result

EventID	Description	Value1 Title	Value2 Title	Value3 Title
30009	Delete Entry			Result
3000A	Modify Entry			Result
3000B	Rename Entry			Result
3000C	Move Entry			Result
3000D	Add Association			Result
3000E	Remove Association			Result
3000F	Query Schema			Result
30010	Check Password			
30011	Check Object Password			
30012	Change Password			Result
30013	Sync			Result
30014	Input XML Document			
30015	Input Transformation Document			
30016	Output Transformation Document			
30017	Event Transformation Document			
30018	Placement Rule Transformation Document			
30019	Create Rule Transformation Document			
3001A	Input Mapping Rule Transformation Document			
3001B	Output Mapping Rule Transformation Document			
3001C	Matching Rule Transformation Document			
3001D	Command Transformation Document			
3001E	Publisher Filter Transformation Document			
3001F	User Agent Request			Result

EventID	Description	Value1 Title	Value2 Title	Value3 Title
30020	Resync Driver			Result
30021	Migrate			
30022	Driver Start	State		
30023	Driver Stop	State		
30024	Password Sync			Result
30025	Password Reset			
30026	DirXML Error	Code		
30027	DirXML Warning	Code		
30028	Custom Operation			
30029	Clear Attribute			Result
3002A	Add Value - Modify Entry			Result
3002B	Remove Value			Result
3002C	Merge Entries			
3002D	Get Named Password			Result
3002E	Reset Attributes			
3002F	Add Value - Add Entry			Result

Table C-4 *Engine Events Fields: Data Type and Triggers*

EventID	Description	Data Type	Triggers
30001	Status Success	XML Document	Many different events can cause the status success event to occur. It usually signifies an operation was successfully completed.
30002	Status Retry	XML Document	Many different events can cause the status retry event to occur. It signifies an operation was not completed and the operation must be tried again later.
30003	Status Warning	XML Document	Many different events can cause the status warning event to occur. It usually signifies an operation was completed with minor problems.
30004	Status Error	XML Document	Many different events can cause the status error event to occur. It usually signifies an operation was not completed successfully.

EventID	Description	Data Type	Triggers
30005	Status Fatal	XML Document	Many different events can cause the status fatal event to occur. It usually signifies an operation was not completed successfully and the engine or driver could not continue.
30006	Status Other	XML Document	Any status document processed with a level other than the five previously defined will create a status other event. These events can only be generated within a style sheet or rule.
30007	Search	XML Document	Occurs when a query document is sent to the IDM engine or driver.
30008	Add Entry	XML Document	Occurs when an object is added.
30009	Delete Entry	XML Document	Occurs when an object is deleted.
3000A	Modify Entry	XML Document	Occurs when an object is modified.
3000B	Rename Entry	XML Document	Occurs when an object is renamed.
3000C	Move Entry	XML Document	Occurs when an object is moved.
3000D	Add Association	XML Document	Occurs when an association is added. It can happen on an add or a match.
3000E	Remove Association	XML Document	When an object is deleted, there is no remove association event. The remove association occurs when a User object is deleted in the disparate application, and the delete is then converted into a modify that removes the association.
3000F	Query Schema	XML Document	Occurs when a query schema operation is sent to the IDM engine or driver.
30010	Check Password		Manual function that is initiated via iManager.
30011	Check Object Password	XML Document	Occurs when a request is issued to check an objects password, other than the driver.
30012	Change Password	XML Document	Occurs when a request is issued to check the Drivers password.
30013	Sync	XML Document	Occurs when a Sync event is requested.

EventID	Description	Data Type	Triggers
30014	Input XML Document	XML Document	Generated whenever an input document is created by the engine or driver.
30015	Input Transformation Document	XML Document	Generated after the input transformation policies are processed, allowing the user to view the transformed document.
30016	Output Transformation Document	XML Document	Generated after the output transformation policies are processed, allowing the user to view the transformed document.
30017	Event Transformation Document	XML Document	Generated after the event transformation policies are processed, allowing the user to view the transformed document.
30018	Placement Rule Transformation Document	XML Document	Generated after the placement rule policies are processed, allowing the user to view the transformed document.
30019	Create Rule Transformation Document	XML Document	Generated after the create rule policies are processed, allowing the user to view the transformed document.
3001A	Input Mapping Rule Transformation Document	XML Document	Generated after the schema mapping rules are processed which convert the document to the eDirectory schema
3001B	Output Mapping Rule Transformation Document	XML Document	Generated after the schema mapping rules are processed which convert the document to the applications schema.
3001C	Matching Rule Transformation Document	XML Document	Generated after the matching rule policies are processed, allowing the user to view the transformed document.
3001D	Command Transformation Document	XML Document	Generated after the command transformation policies are processed, allowing the user to view the transformed document.
3001E	Publisher Filter Transformation Document	XML Document	Generated after the processing the notify filter on the publisher channel, allowing the user to view the transformed document.
3001F	User Agent Request	XML Document	Occurs when a User Agent XDS command document is sent to the Driver on the subscriber channel.

EventID	Description	Data Type	Triggers
30020	Resync Driver		Occurs when a resync request is issued.
30021	Migrate		Occurs when a migrate request is issued.
30022	Driver Start	XML Document	Occurs when a driver is started.
30023	Driver Stop	XML Document	Occurs when a driver is stopped.
30024	Password Sync		Generated when setting the distribution or simple password on an object.
30025	Password Reset		Generated when resetting the connected application password after a failed password sync operation.
30026	DirXML Error		Generated whenever the engine throws an internal error.
30027	DirXML Warning		Generated whenever the engine throws an internal warning.
30028	Custom Operation	XML Document	Occurs when an unknown operation appears in an input document. An example of known operations would be an add, delete or modify.
30029	Clear Attribute		Occurs when a modify operation contains a remove-all-value element.
3002A	Add Value - Modify Entry	Value	Occurs when a value is added during the modification of an object.
3002B	Remove Value	Value	Occurs when a modify operation contains a remove-value element.
3002C	Merge Entries	XML Document	Occurs when two objects are being merged.
3002D	Get Named Password	XML Document	Generated on a Get Named Password operation.
3002E	Reset Attributes	XML Document	Occurs when a Reset document is issued on the publisher or subscriber channels.
3002F	Add Value - Add Entry	Value	Occurs when a value is added during the creation of an object.

C.2 Server Events

The tables contains the list of server events that can be audited through Novell Audit.

Table C-5 *Server Events Fields: Originator Title, Target Title, and Subtarget Title*

EventID	Description	Originator Title	Target Title	Subtarget Title
307D0	Config:Log Events	Server	Driver	Attribute name
307D1	Config:Driver Cache Limit	Server	Driver	Attribute name
307D2	Config:Driver Set	Server	Server	Attribute name
307D3	Config:Driver Start Option	Server	Driver	Attribute name
307D4	Driver Resync	Server	Driver	
307D5	Migrate Application Server	Server	Driver	
307D6	Shim Password Set	Server	Driver	Attribute name
307D7	Keyed Password Set	Server	Driver	
307D8	Remote Loader Password Set	Server	Driver	Attribute name

Table C-6 *Server Event Fields: Text1 Title, Text2 Title, and Text3 Title*

EventID	Description	Text1 Title	Text2 Title	Text3 Title
307D0	Config:Log Events			Operation
307D1	Config:Driver Cache Limit			
307D2	Config:Driver Set	Driver Set	Type	
307D3	Config:Driver Start Option			Message
307D4	Driver Resync			
307D5	Migrate Application Server			
307D6	Shim Password Set			
307D7	Keyed Password Set		Type	
307D8	Remote Loader Password Set			

Table C-7 *Server Event Field: Value1 Title, Value2 Title, and Value3 Title*

EventID	Description	Value1 Title	Value2 Title	Value3 Title
307D0	Config:Log Events			Result

EventID	Description	Value1 Title	Value2 Title	Value3 Title
307D1	Config:Driver Cache Limit	Limit		Result
307D2	Config:Driver Set			Result
307D3	Config:Driver Start Option	Start option		Result
307D4	Driver Resync			Result
307D5	Migrate Application Server			Result
307D6	Shim Password Set		Version	Result
307D7	Keyed Password Set			Result
307D8	Remote Loader Password Set		Version	Result

Table C-8 *Server Event Fields: Data Type and Triggers*

EventID	Description	Data Type	Triggers
307D0	Config:Log Events	Input buffer	Occurs when the log events attribute is changed on the Driver or Driver Set object.
307D1	Config:Driver Cache Limit		Occurs when the Driver Cache Limit attribute is changed on a Driver object.
307D2	Config:Driver Set	Input buffer	Occurs when the Driver Set/Server association is changed.
307D3	Config:Driver Start Option	Input buffer	Occurs when the Driver Start Option is changed for a Driver object.
307D4	Driver Resync		Occurs when a resynchronization is issued for the driver.
307D5	Migrate Application Server	XML Document	Occurs when the migration of the application server happens.
307D6	Shim Password Set		Occurs when the Application password is set.
307D7	Keyed Password Set		
307D8	Remote Loader Password Set		Occurs when the Remote Loader password is set.

C.3 Remote Loader Events

The tables contains the list of Remote Loader events that can be audited through Novell Audit.

Table C-9 Remote Loader Events Fields: Originator Title, Target Title, and Subtarget Title

EventID	Description	Originator Title	Triggers
30BB8	Remote Loader Start	Instance	Occurs when the Remote Loader starts.
30BB9	Remote Loader Stop	Instance	Occurs when the Remote Loader stops.
30BBA	Remote Loader Connection Established	Instance	Occurs when the Remote Loader connection is established.
30BBB	Remote Loader Connection Dropped	Instance	Occurs when the Remote Loader connection is dropped.

C.4 Detail Portlets

Table C-10 Detail Portlets Fields: Originator Title, Target Title, and Subtarget Title

EventID	Description	Originator Title	Target Title	Subtarget Title
31400	Delete_Entity	User Name	Entity DN	Entity Definition
31401	Update_Entity	User Name	Entity DN	Entity Definition

Table C-11 Detail Portlets Fields: Group Title, Group Type, and Triggers

EventID	Description	Group Title	Group Type	Triggers
31400	Delete_Entity	Group Number	Number	Occurs when an object is deleted.
31401	Update_Entity	Group Number	Number	Occurs when an object is modified.

C.5 Change Password Portlet

Table C-12 Change Password Portlet Fields: Originator Title, Target Title, and Text3 Title

EventID	Description	Originator Title	Target Title	Text3 Title
31420	Change_Password_Failure	Initiator ID	Target DN	Error Message
31421	Change_Password_Success	Initiator ID	Target DN	

Table C-13 Change Password Portlet Fields: Value3 Title, Value3 Type, and Triggers

EventID	Description	Value3 Title	Value3 Type	Triggers
31420	Change_Password_Failure	Error Number	Boolean	Occurs when a password change fails.
31421	Change_Password_Success			Occurs when a password change is successful.

C.6 Forgot Password Change Password Portlet

Table C-14 *Forgot Password Change Password Portlet Fields: Originator Title, Target Title, and Text3 Title*

EventID	Description	Originator Title	Target Title	Text3 Title
31420	Forgot_Password_Change_Failure	Initiator ID	Target DN	Error Message
31421	Forgot_Password_Change_Success	Initiator ID	Target DN	

Table C-15 *Forgot Password Change Password Portlet Fields: Value3 Title, Value3 Type, and Group Title*

EventID	Description	Value3 Title	Value3 Type	Group Title
31420	Forgot_Password_Change_Failure	Error Number	Boolean	Group Number
31421	Forgot_Password_Change_Success			Group Number

Table C-16 *Forgot Password Change Password Portlet Fields: Group Type and Triggers*

EventID	Description	Group Type	Triggers
31420	Forgot_Password_Change_Failure	Number	Occurs when the Forgot Password change fails.
31421	Forgot_Password_Change_Success	Number	Occurs when the Forgot Password change is successful.

C.7 Search List Portlet

Table C-17 *Search List Portlet Fields: Originator Title, Target Title, and Group Title*

EventID	Description	Originator Title	Target Title	Group Title
31430	Search_Request	User ID	Search Key	User ID
31431	Search_Saved	User ID	Search Key	User ID

Table C-18 *Search List Portlet Fields: Group Type, Data Title, and Data Type*

EventID	Description	Group Type	Data Title	Data Type
31430	Search_Request	Number	Search XML	String
31431	Search_Saved	Number	Search XML	String

Table C-19 Search List Portlet Fields: Triggers

EventID	Description	Triggers
31430	Search_Request	Occurs when a user performs a search request.
31431	Search_Saved	Occurs when the user selects My Saved Searches.

C.8 Create Portlet

Table C-20 Create Portlet Fields: Originator Title, Target Title, and Subtarget Title

EventID	Description	Originator Title	Target Title	Subtarget Title
31440	Create_Entity	User Name	Entity DN	Entity Definition

Table C-21 Create Portlet Fields: Triggers

Event ID	Description	Triggers
31440	Create_Entity	Occurs when an object is created.

C.9 Security Context

The tables contains the list of security events that can be audited through Novell Audit.

Table C-22 Security Context Fields: Originator Title, Target Title, and Text1 Title

EventID	Description	Originator Title	Target Title	Text1 Title
31540	Create_Proxy_Definition_Success	Initiator ID	Definition	Detail
31541	Create_Proxy_Definition_Failure	Initiator ID	Definition	Detail
31542	Update_Proxy_Definition_Success	Initiator ID	Definition	Detail
31543	Update_Proxy_Definition_Failure	Initiator ID	Definition	Detail
31544	Delete_Proxy_Definition_Success	Initiator ID	Definition	Detail
31545	Delete_Proxy_Definition_Failure	Initiator ID	Definition	Detail
31546	Create_Delegatee_Definition_Success	Initiator ID	Definition	Detail
31547	Create_Delegatee_Definition_Failure	Initiator ID	Definition	Detail
31548	Update_Delegatee_Definition_Success	Initiator ID	Definition	Detail
31549	Update_Delegatee_Definition_Failure	Initiator ID	Definition	Detail
3154A	Delete_Delegatee_Definition_Success	Initiator ID	Definition	Detail
3154B	Delete_Delegatee_Definition_Failure	Initiator ID	Definition	Detail
3154C	Create_Availability_Success	Initiator ID	Target	

EventID	Description	Originator Title	Target Title	Text1 Title
3154D	Create_Availability_Failure	Initiator ID	Target	Detail
3154E	Delete_Availability_Success	Initiator ID	Target	Detail
3154F	Delete_Availability_Failure	Initiator ID	Target	Detail

Table C-23 Security Context Fields: Text3 Title, Data Title, and Data Type

EventID	Description	Text3 Title	Data Title	Data Type
31540	Create_Proxy_Definition_Success			
31541	Create_Proxy_Definition_Failure	Error Message	stacktrace	String
31542	Update_Proxy_Definition_Success			
31543	Update_Proxy_Definition_Failure	Error Message	stacktrace	String
31544	Delete_Proxy_Definition_Success			
31545	Delete_Proxy_Definition_Failure	Error Message	stacktrace	String
31546	Create_Delegatee_Definition_Success			
31547	Create_Delegatee_Definition_Failure	Error Message	stacktrace	String
31548	Update_Delegatee_Definition_Success			
31549	Update_Delegatee_Definition_Failure	Error Message	stacktrace	String
3154A	Delete_Delegatee_Definition_Success			
3154B	Delete_Delegatee_Definition_Failure	Error Message	stacktrace	String
3154C	Create_Availability_Success			
3154D	Create_Availability_Failure	Error Message	stacktrace	String
3154E	Delete_Availability_Success			
3154F	Delete_Availability_Failure	Error Message	stacktrace	String

Table C-24 Security Context Fields: Triggers

EventID	Description	Triggers
31540	Create_Proxy_Definition_Success	Occurs on successful creation of proxy definition.
31541	Create_Proxy_Definition_Failure	Occurs on failed creation of proxy definition.
31542	Update_Proxy_Definition_Success	Occurs on successful update of proxy definition.
31543	Update_Proxy_Definition_Failure	Occurs on failed update of proxy definition.
31544	Delete_Proxy_Definition_Success	Occurs on successful deletion of proxy definition.

EventID	Description	Triggers
31545	Delete_Proxy_Definition_Failure	Occurs on failed deletion of proxy definition.
31546	Create_Delegatee_Definition_Success	Occurs on successful creation of delegatee definition.
31547	Create_Delegatee_Definition_Failure	Occurs on failed creation of delegatee definition.
31548	Update_Delegatee_Definition_Success	Occurs on successful update of delegatee definition.
31549	Update_Delegatee_Definition_Failure	Occurs on failed update of delegatee definition.
3154A	Delete_Delegatee_Definition_Success	Occurs on successful deletion of delegatee definition.
3154B	Delete_Delegatee_Definition_Failure	Occurs on failed deletion of delegatee definition.
3154C	Create_Availability_Success	Occurs on successful creation of availability status.
3154D	Create_Availability_Failure	Occurs on failed creation of availability status.
3154E	Delete_Availability_Success	Occurs on successful deletion of availability status.
3154F	Delete_Availability_Failure	Occurs on failed deletion of availability status.

C.10 Workflow

The tables contains the list of User Application events that can be audited through Novell Audit.

Table C-25 *Workflow Fields: Originator Title, Target Title, and Subtarget Title*

EventID	Description	Originator Title	Target Title	Subtarget Title
31520	Workflow_Error	Initiator ID		
31521	Workflow_Started	Initiator ID		
31522	Workflow_Forwarded	Initiator ID	Recipient	Process Name
31523	Workflow_Reassigned	Initiator ID	Recipient	Process Name
31524	Workflow_Approved	Initiator ID	Recipient	Process Name
31525	Workflow_Refused	Initiator ID	Recipient	Process Name
31526	Workflow_Ended	Initiator ID	Recipient	Process Name
31527	Workflow_Claimed	Initiator ID	Recipient	Process Name
31528	Workflow_Unclaimed	Initiator ID	Recipient	Process Name
31529	Workflow_Denied	Initiator ID	Recipient	Process Name

EventID	Description	Originator Title	Target Title	Subtarget Title
3152A	Workflow_Completed	Initiator ID	Recipient	Process Name
3152B	Workflow_Timedout	Initiator ID	Recipient	Process Name
3152C	User_Message	Initiator ID	Author	
3152D	Provision_Error	Initiator ID	Recipient	Process Name
3152E	Provision_Submitted	Initiator ID	Recipient	Process Name
3152F	Provision_Success	Initiator ID	Recipient	Process Name
31530	Provision_Failure	Initiator ID	Recipient	Process Name
31531	Provision_Granted	Initiator ID	Recipient	Process Name
31532	Provision_Revoked	Initiator ID	Recipient	Process Name
31533	Workflow_Retracted	Initiator ID	Recipient	Process Name

Table C-26 *Workflow Fields: Text1 Title, Text2 Title, and Text3 Title*

EventID	Description	Text1 Title	Text2 Title	Text3 Title
31520	Workflow_Error	Activity	Process ID	Error Message
31521	Workflow_Started	Activity	Process ID	
31522	Workflow_Forwarded	Activity	Process ID	
31523	Workflow_Reassigned	Activity	Process ID	
31524	Workflow_Approved	Activity	Process ID	Secondary User
31525	Workflow_Refused	Activity	Process ID	Secondary User
31526	Workflow_Ended	Activity	Process ID	
31527	Workflow_Claimed	Activity	Process ID	Secondary User
31528	Workflow_Unclaimed	Activity	Process ID	Secondary User
31529	Workflow_Denied	Activity	Process ID	Secondary User
3152A	Workflow_Completed	Activity	Process ID	
3152B	Workflow_Timedout	Activity	Process ID	
3152C	User_Message		Message	
3152D	Provision_Error	Activity	Process ID	Error Message
3152E	Provision_Submitted	Activity	Process ID	
3152F	Provision_Success	Activity	Process ID	
31530	Provision_Failure	Activity	Process ID	
31531	Provision_Granted	Activity	Process ID	
31532	Provision_Revoked	Activity	Process ID	

EventID	Description	Text1 Title	Text2 Title	Text3 Title
31533	Workflow_Retracted	Activity	Process ID	Secondary User

Table C-27 Workflow Fields: Value3 Title, Value3 Type, and Data Title

EventID	Description	Value3 Title	Value3 Type	Data Title
31520	Workflow_Error	Error Number	Boolean	stacktrace
31521	Workflow_Started			
31522	Workflow_Forwarded			
31523	Workflow_Reassigned			
31524	Workflow_Approved			Secondary User Type
31525	Workflow_Refused			Secondary User Type
31526	Workflow_Ended			
31527	Workflow_Claimed			Secondary User Type
31528	Workflow_Unclaimed			Secondary User Type
31529	Workflow_Denied			Secondary User Type
3152A	Workflow_Completed			
3152B	Workflow_Timedout			
3152C	User_Message			
3152D	Provision_Error	Error Number	Boolean	stacktrace
3152E	Provision_Submitted			
3152F	Provision_Success			
31530	Provision_Failure			
31531	Provision_Granted			
31532	Provision_Revoked			
31533	Workflow_Retracted			Secondary User Type

Table C-28 Workflow Fields: Data Type and Triggers

EventID	Description	Data Type	Triggers
31520	Workflow_Error	String	Many items can cause this event.

EventID	Description	Data Type	Triggers
31521	Workflow_Started		Occurs when the workflow starts.
31522	Workflow_Forwarded		Occurs when the workflow is forwarded.
31523	Workflow_Reassigned		Occurs when the workflow is reassigned.
31524	Workflow_Approved	String	Occurs when the workflow is approved.
31525	Workflow_Refused	String	Occurs when the workflow is refused.
31526	Workflow_Ended		Occurs when the workflow ends.
31527	Workflow_Claimed	String	Occurs when the workflow is claimed.
31528	Workflow_Unclaimed	String	
31529	Workflow_Denied	String	Occurs when the workflow is denied.
3152A	Workflow_Completed		Occurs when the workflow is completed.
3152B	Workflow_Timedout		Occurs when the workflow timed out.
3152C	User_Message		
3152D	Provision_Error	String	Many items can cause this event.
3152E	Provision_Submitted		
3152F	Provision_Success		
31530	Provision_Failure		
31531	Provision_Granted		
31532	Provision_Revoked		
31533	Workflow_Retracted	String	Occurs when the workflow is retracted.

C.11 Reports

Below are examples of what the Novell Audit reports look like. Here is the list of reports that can be run.

- ◆ Administrative Action Report
- ◆ Historical Approval Flow Report
- ◆ Resource Provisioning Report
- ◆ Specific User Audit Trail
- ◆ Specific User Provisioning
- ◆ User Provisioning

Figure C-1 Administrative Action Report

Novell® Audit Report for Identity Manager			
Administrative Action Report			Report Last Modified: 10/13/2005 Report Generated On: 10/13/2005 Total pages: 5
Total # Events: 121			
Report Period: - 10/13/2005 8:43:50AM			
Date / Time	Administrator	Subject	Action
8/18/2005 5:45:17PM	cn=admin,ou=idm sample-cts10,ovenovell	cn=TestCreateGroup,ou=groups,ou=idm sample-cts10,ovenovell	Entity Deleted
8/18/2005 7:07:40PM	cn=admin,ou=idm sample-cts10,ovenovell	cn=testCreateUser11,ou=users,ou=idm sample-cts10,ovenovell	Entity Deleted
8/18/2005 7:09:05PM	cn=admin,ou=idm sample-cts10,ovenovell	cn=TestCreateGroup,ou=groups,ou=idm sample-cts10,ovenovell	Entity Deleted
8/18/2005 7:12:50PM	cn=admin,ou=idm sample-cts10,ovenovell	cn=testCreateUser11,ou=users,ou=idm sample-cts10,ovenovell	Entity Deleted
8/18/2005 7:13:39PM	cn=admin,ou=idm sample-cts10,ovenovell	cn=TestCreateGroup,ou=groups,ou=idm sample-cts10,ovenovell	Entity Deleted
8/23/2005 4:56:39PM	cn=admin,ou=idm sample,ovenovell	cn=TestCreateGroup,ou=groups,ou=idm sample,ovenovell	Entity Deleted
8/31/2005 12:01:55PM	cn=admin,ou=idm sample,ovenovell	cn=testCreateUser,ou=users,ou=idm sample,ovenovell	Entity Created
8/31/2005 12:02:18PM	cn=admin,ou=idm sample,ovenovell	cn=TestCreateGroup,ou=groups,ou=idm sample,ovenovell	Entity Created
8/31/2005 12:19:07PM	cn=admin,ou=idm sample,ovenovell	cn=testCreateUser,ou=users,ou=idm sample,ovenovell	Entity Created
8/31/2005 12:19:31PM	cn=admin,ou=idm sample,ovenovell	cn=TestCreateGroup,ou=groups,ou=idm sample,ovenovell	Entity Created
8/31/2005 12:27:58PM	cn=admin,ou=idm sample,ovenovell	cn=testCreateUser,ou=users,ou=idm sample,ovenovell	Entity Created
8/31/2005 12:28:22PM	cn=admin,ou=idm sample,ovenovell	cn=TestCreateGroup,ou=groups,ou=idm sample,ovenovell	Entity Created
8/31/2005 2:59:39PM	cn=admin,ou=idm sample,ovenovell	cn=testCreateUser,ou=users,ou=idm sample,ovenovell	Entity Created
8/31/2005 3:24:30PM	cn=admin,ou=idm sample,ovenovell	cn=testCreateUser,ou=users,ou=idm sample,ovenovell	Entity Created
8/31/2005 8:11:59PM	cn=admin,ou=idm sample-Jeff,ovenovell	cn=testCreateUser,ou=users,ou=idm sample-Jeff,ovenovell	Entity Deleted
8/31/2005 8:12:23PM	cn=admin,ou=idm sample-Jeff,ovenovell	cn=TestCreateGroup,ou=groups,ou=idm sample-Jeff,ovenovell	Entity Deleted
8/31/2005 8:12:55PM	cn=admin,ou=idm sample-Jeff,ovenovell	cn=admin,ou=idm sample-Jeff,ovenovell	Entity Updated
8/31/2005 8:13:03PM	cn=admin,ou=idm sample-Jeff,ovenovell	cn=admin,ou=idm sample-Jeff,ovenovell	Entity Updated
9/1/2005 10:29:53AM	cn=admin,ou=idm sample-Jeff,ovenovell	cn=aa,ou=users,ou=idm sample-Jeff,ovenovell	Entity Deleted
9/1/2005 11:31:45AM	cn=admin,ou=idm sample,ovenovell	cn=asoprano,ou=users,ou=idm sample,ovenovell	Entity Created

Figure C-2 Historical Approval Flow Report



Figure C-3 Resource Provisioning Report

Novell® Audit Report for Identity Manager

Resource Provisioning Report

Report Last Modified: 10/13/2005
 Report Generated On: 10/13/2005
 Total pages: 3

Total # Events: 42

Report Period: - 10/13/2005 8:47:18AM

Resource

Value Adder(Mgr Approve - 5 minute, 1 retry TD)	Date / Time	User Name	Action
Provision Granted	9/12/2005 4:38:35PM	cn=ablake,ou=users,ou=idm sample-Jeff,o=novell	Entitlement Provisioning Activity
Provision Success	9/12/2005 4:38:35PM	cn=ablake,ou=users,ou=idm sample-Jeff,o=novell	ENTITLEMENT
Provision Submitted	9/12/2005 4:38:35PM	cn=ablake,ou=users,ou=idm sample-Jeff,o=novell	Entitlement Provisioning Activity
Provision Success	9/12/2005 4:33:32PM	cn=ablake,ou=users,ou=idm sample-Jeff,o=novell	ENTITLEMENT
Provision Granted	9/12/2005 3:32:06PM	cn=ablake,ou=users,ou=idm sample-Jeff,o=novell	Entitlement Provisioning Activity
Provision Submitted	9/12/2005 3:32:06PM	cn=ablake,ou=users,ou=idm sample-Jeff,o=novell	Entitlement Provisioning Activity
Revoke Active Directory Account (Mgr Approve-No Timeout)	Date / Time	User Name	Action
Provision Revoked	9/9/2005 12:37:37PM	cn=ablake,ou=users,ou=idm sample-Jeff,o=novell	Entitlement Provisioning Activity
Provision Submitted	9/9/2005 12:37:37PM	cn=ablake,ou=users,ou=idm sample-Jeff,o=novell	Entitlement Provisioning Activity
Enable Active Directory Account 2 Parallel(Mgr, HR Group) No Timeout	Date / Time	User Name	Action
Provision Granted	9/28/2005 2:12:27PM	cn=ablake,ou=users,ou=idm sample-Jeff,o=novell	Entitlement Provisioning Activity
Provision Submitted	9/28/2005 2:12:27PM	cn=ablake,ou=users,ou=idm sample-Jeff,o=novell	Entitlement Provisioning Activity
Provision Granted	9/7/2005 4:52:02PM	cn=ablake,ou=users,ou=idm sample-Jeff,o=novell	Entitlement Provisioning Activity
Provision Submitted	9/7/2005 4:52:02PM	cn=ablake,ou=users,ou=idm sample-Jeff,o=novell	Entitlement Provisioning Activity
Enable Active Directory Account (Mgr Approve-No Timeout)	Date / Time	User Name	Action
Provision Granted	10/12/2005 1:03:28PM	cn=??,ou=users,ou=idm sample-qatest,o=novell	Entitlement Provisioning Activity
Provision Submitted	10/12/2005 1:03:28PM	cn=??,ou=users,ou=idm sample-qatest,o=novell	Entitlement Provisioning Activity
Provision Success	9/9/2005 4:12:02PM	cn=ablake,ou=users,ou=idm sample-Jeff,o=novell	ENTITLEMENT

Figure C-4 Specific User Audit Trail 1

Novell® Audit Report for Identity Manager

Specific User Audit Trail

Report Period: - 10/13/2005 8:51:32AM

User ID: ablake

Report Last Modified: 10/13/2005
 Report Generated On: 10/13/2005
 Total pages: 8

Approval Flow

Workflow Event: fecedbe80a3d4abd83c9476a1b576ea2			
Date / Time	Action	Initiator ID	
9/12/2005 3:20:42PM	Workflow Started	cn=ablake,ou=users,ou=idm sample-Jeff,o=novell	
9/12/2005 3:20:43PM	Workflow Forwarded	Workflow Administrator	
9/12/2005 3:25:43PM	Workflow Reassigned	Unclaimed	
9/12/2005 3:30:44PM	Workflow Forwarded	Workflow Administrator	
9/12/2005 3:30:44PM	Workflow Ended	Workflow Administrator	
9/12/2005 3:30:44PM	Workflow Denied	System	

Workflow Event: fc6d74b1268243b3beac52261439dea0			
Date / Time	Action	Initiator ID	
9/28/2005 1:12:19PM	Workflow Started	cn=ablake,ou=users,ou=idm sample-Jeff,o=novell	
9/28/2005 1:12:22PM	Workflow Forwarded	Workflow Administrator	
9/28/2005 2:12:23PM	Workflow Forwarded	Workflow Administrator	
9/28/2005 2:12:23PM	Workflow Forwarded	Workflow Administrator	
9/28/2005 2:12:23PM	Workflow Forwarded	Workflow Administrator	
9/28/2005 2:12:23PM	Workflow Forwarded	Workflow Administrator	
9/28/2005 2:12:23PM	Workflow Approved	System	
9/28/2005 2:12:23PM	Workflow Approved	System	
9/28/2005 2:12:23PM	Workflow Completed	Workflow Administrator	
9/28/2005 2:12:27PM	Workflow Forwarded	Workflow Administrator	
9/28/2005 2:12:27PM	Workflow Ended	Workflow Administrator	
9/28/2005 2:12:27PM	Provision Submitted	Workflow Administrator	
9/28/2005 2:12:27PM	Provision Granted	Workflow Administrator	

Workflow Event: efaa8304e07641edb9e6375a1a36e396			
Date / Time	Action	Initiator ID	
10/12/2005 11:58:13AM	Workflow Started	cn=ablake,ou=users,ou=idm sample-qatest,o=novell	
10/12/2005 11:58:13AM	Workflow Forwarded	Workflow Administrator	

Workflow Event: ea341eb11a824e669e356837745fe264			
Date / Time	Action	Initiator ID	
9/27/2005 4:24:44PM	Workflow Started	cn=mackenzie,ou=users,ou=idm sample-Jeff,o=novell	
9/27/2005 4:24:44PM	Workflow Forwarded	Workflow Administrator	

Page 1 of 8
Specific User Audit Trail

Figure C-5 Specific User Audit Trail 2

<u>Date / Time</u>	<u>Action</u>	<u>Target</u>	<u>Results</u>
9/12/2005 10:37:16AM	Search Request		Success
9/12/2005 10:37:39AM	Search Request		Success
9/12/2005 12:48:28PM	Change Password	cn=ablake ,ou=users,ou=idm sample- Jeff,o=novell	Success
9/12/2005 12:48:45PM	Change Password	cn=ablake ,ou=users,ou=idm sample- Jeff,o=novell	Success
9/15/2005 5:00:44PM	Search Request		Success
9/22/2005 2:00:49PM	Search Request		Success

Page 1 of 1 SelfServiceSub.rpt

Page 1 of 1 Specific User Audit Trail

Figure C-6 Specific User Audit Trail 3

Administrative Actions

<u>Date / Time</u>	<u>Administrator</u>	<u>Subject</u>	<u>Action</u>
9/28/2005 2:27:10PM	cn=admin,ou=idm sample,o=novell	cn=ablake,ou=users,ou=idm sample,o=novell	Entity Updated
10/5/2005 5:22:37PM	cn=admin,ou=idm sample,o=novell	cn=ablake,ou=users,ou=idm sample,o=novell	Entity Updated

Page 1 of 1 *AdministrativeActionSub.rpt*

Figure C-7 Specific User Provisioning Report

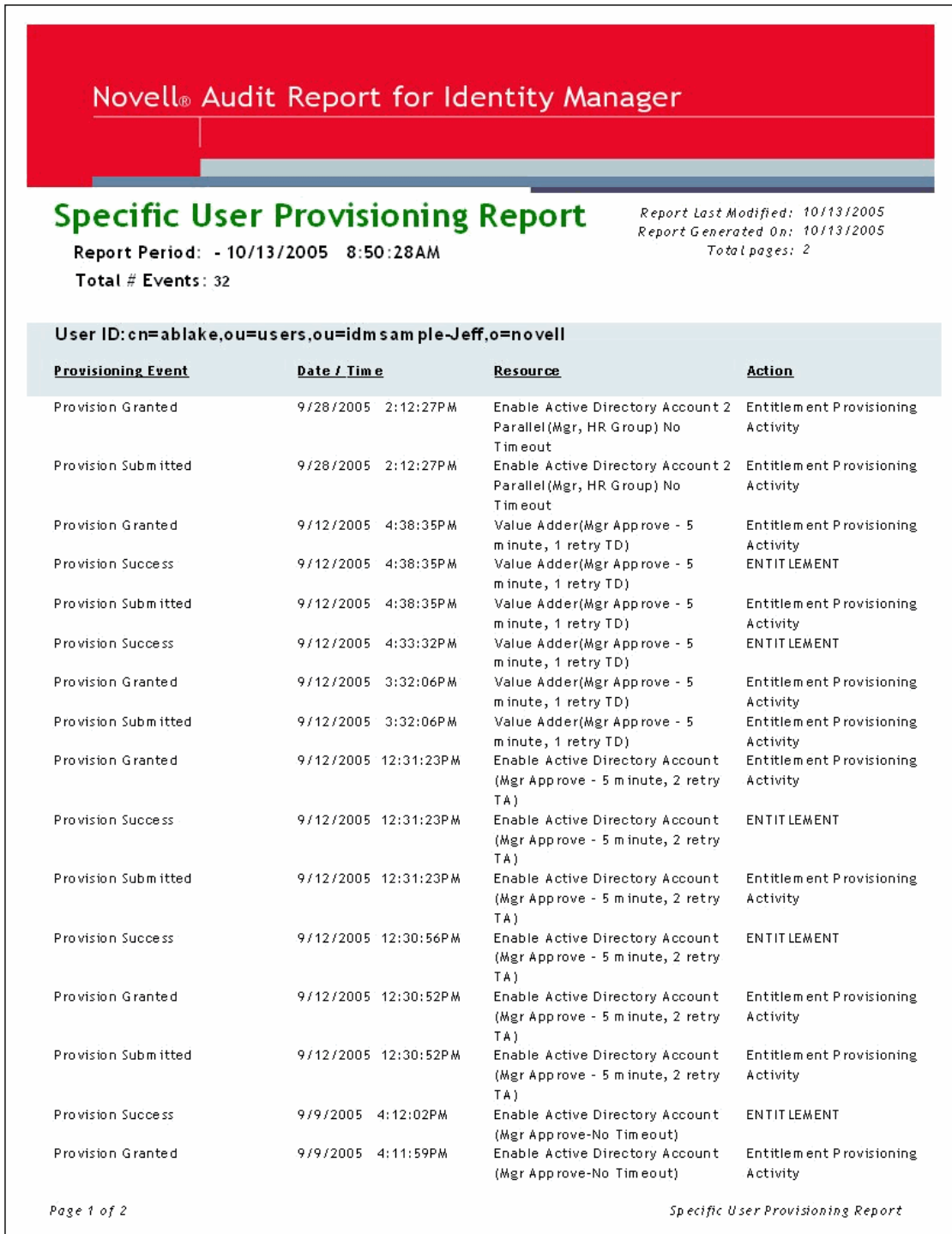
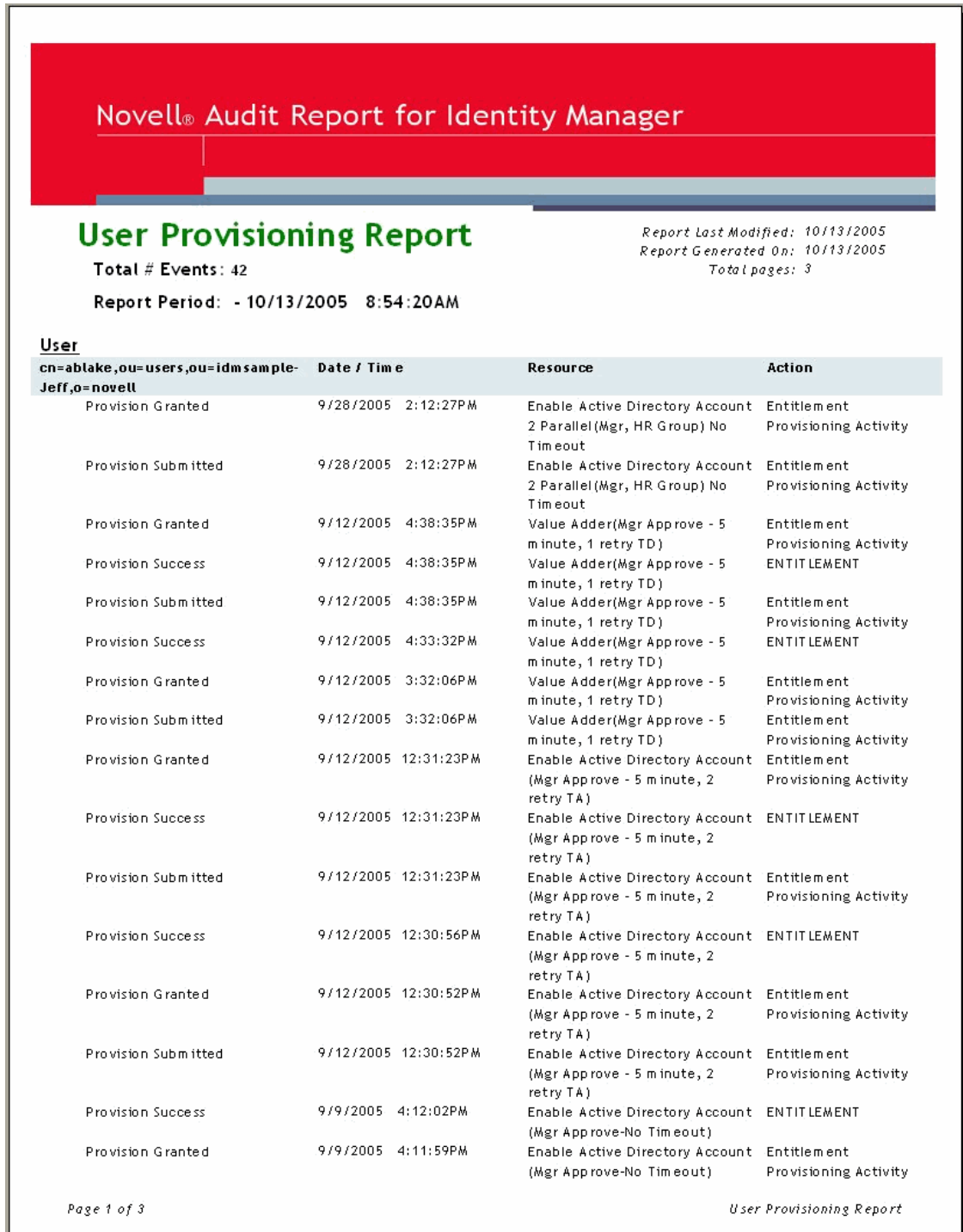


Figure C-8 User Provisioning Report



Manual Task Service Driver: Replacement Data

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 297).
- ◆ 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.xml.
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.xml.

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

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 291](#) 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

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="$responder-dn$"/>
</form:if-single-item>
```

```
    $responder-dn$  
</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

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 291 and [Appendix E, “Manual Task Service Driver: Automatic Replacement Data Items,”](#) on page 297.

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

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
1lbnNvZGVkUGFyYW1zdAACW0JbABB1bmNyeXB0ZWRDb250ZW50cQB+AAFMAAlw
YXJhbXNBbGd0ABJMamF2YS9sYW5nL1N0cmLuZztMAAdzZWFsQWxncQB+AAJ4cH
VyAAJbQqzzF/gGCFtgAgAAeHAAAAAPMA0ECEIBRohGPjxEAgEKdXEAFgAEAAAA
uMSFqzHXwtMx8DkRCzkK1O46sEz1u51o3MDvHn+3+fe6SphHr3HgjlI4Jp3rUk
H7y6dXvcu7iq21Vs+9o6iZVzljTIJX/jjRrVZlR5JOuRNhk8JHFZ8FhgsmiIAH
/Fs61k4WmyEcmYfWmfqfBVeThr3Avwcm6ranS5Mm2U5i9Z/DBR13pIAobMpWY
kMaz4+G9e6oovBsiPdp6jSPzbFxcgALI2AMBh4hf9jnx7zOU9Uvd9qXtaE2rR0
AANQQkV0ABBQQkVXaXR0TUQ1QW5kREVT</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=rO0ABXNyABlqYXZheC5jcnlwdG8uU2VhbGVkT2JqZWN0PjY9psO3VHACAA
RbAA1lbnNvZGVkUGFyYW1zdAACW0JbABB1bmNyeXB0ZWRDb250ZW50cQB%2BAAF
MAAlwYXJhbXNBbGd0ABJMamF2YS9sYW5nL1N0cmLuZztMAAdzZWFsQWxncQB%2B
AAJ4cHVyAAJbQqzzF%2FgGCFtgAgAAeHAAAAAPMA0ECEIBRohGPjxEAgEKdXEAF
gAEAAAAuMSFqzHXwtMx8DkRCzkK1O46sEz1u51o3MDvHn%2B3%2Bfe6SphHr3Hg
jlI4Jp3rUkH7y6dXvcu7iq21Vs%2B9o6iZVzljTIJX%2FjjRrVZlR5JouRNhk8J
HFZ8FhgsmiIAH%2FFs61k4WmyEcmYfWmfqfBVeThr3Avwcm6ranS5Mm2U5i9Z%
2FDBR13pIAobMpWYkMaz4%2BG9e6oovBsiPdp6jSPzbFxcgALI2AMBh4hf9jnx7
zOU9Uvd9qXtaE2rR0AANQQkV0ABBQQkVXaXR0TUQ1QW5kREVT</item>
  <item name="url">
https://192.168.0.1:8180/

```

```

process_template.xml?template=form_template.xml&responder-
dn=%5CPERIN-TAO%5Cnovell%5CProvo%5Cphb&responder-
dn=%5CPERIN-TAO%5Cnovell%5Cprovo%5Ccarol&subject-
name=Joe+the+Intern&association=1671b2%3Aee4246a561%3A-
7fff%3A192.168.0.1&protected-
data=r00ABXNyAB1qYXZheC5jcnlwdG8uU2VhbGVkt2JqZWN0PjY9psO3VHACAA
RbAA11bmNvZGVkUGFyYW1zdAACW0JbABB1bmNyeXB0ZWRDb250ZW50cQB%2BAAF
MAA1wYXJhbXBbGd0ABJMamF2YS9sYW5nL1N0cm1uZztMAAdzZWFsQWxncQB%2B
AAJ4cHVYAAJbQqzzF%2FgGCFtgAgAAeHAAAAAPMA0ECEIBRohGPjxEAgEKdXEAf
gAEAAAAuMSFqzHXwtMx8DkRCzckK1046sEz1u51o3MDvHn%2B3%2BfE6SphHr3Hg
jli4Jp3rUkH7y6dXvcu7iq21Vs%2B9o6iZVz1jTIJX%2FjjRrVZ1R5JouRNhk8J
HFZ8FhgsmlIAH%2FFs61k4WmyEcmYfWmfqfBVeThr3Avwcm6ranS5Mm2U5i9Z%
2FDDBR13pIAobMpWYkMaz4%2BG9e6oovBsiPdp6jSPzbFxcgALi2AMBh4hf9jnx7
zOU9Uvd9qXtaE2rR0AANQQkv0ABBQQkVXaXR0TUQ1QW5kREV
</item>
</replacement-data>

```

- g. The Subscriber processes `html_msg_template.xml` with `process_template.xml`. 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-TA0%5Cnovell%5CProvo%5Cphb&responder-dn=%5CPERIN-
TA0%5Cnovell%5CProvo%5Ccarol&subject-
name=Joe+the+Intern&association=45f0e3%3Aee45e07709%3A-
7fff%3A192.168.0.1&protected-
data=r00ABXNyABlqYXZheC5jcnlwdG8uU2VhbGVkT2JqZWN0PjY9psO3VHACAA
RbAA11bmNvZGVkUGFyYW1zdAACW0JbABB1bmNyeXB0ZWRDb250ZW50cQB%2BAAF
MAAlwYXJhbXNBbGd0ABJMamF2YS9sYW5nL1N0cm1uZztMAAdzZW50cQB%2B
AAJ4cHVyAAJbQqzzF%2FGcFTgAgAAeHAAAAAPMA0ECIr9Z1iG%2B03BAgEKdXE
AfgAEAAAAuMU%2FSoFRkebv2d5Sqa1F91ttjRY51yyW5%2B%2FFIfOuDdYikYi
Db0Jb6607S0dPHjQzeVgu6ptIvGqaEQOEjBjDkY%2Bi4VoVjUSXS3a8fiXB8moM
dPtLJ%2FGyE8Qiwbt4xbkQy48i02k99F2vGmlenRpSP6dD31kZ13dpJ0mGgq2yL
%2FeFaynKyqnjkHLMexcqD8W1VooaR11k2RPk5vDYvC8o2bn22OKKbOnSRM5Y1P
S0iwzxo0JVcnVYyt0AANQQkv0ABBQkvXaXR0TUQ1QW5kREVT">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">r00ABXNyABlqYXZheC5jcnlwdG8uU2VhbGVkT2JqZWN0PjY9psO3VHACAA
ARbAA11bmNvZGVkUGFyYW1zdAACW0JbABB1bmNyeXB0ZWRDb250ZW50cQB+AAFM
AA1wYXJhbXNBbGd0ABJMamF2YS9sYW5nL1N0cm1uZztMAAdzZW50cQB+AAJ

```

```

4cHVyAAJbQqzzF/
gGCFTgAgAAeHAAAAAPMA0ECIr9Z1iG+O3BAgEKdXEAFgAEAAAAuMU/
SoFRkebv2d5Sqa1F91ttjRY51yyW5+/
FifOuDdYikYiDbOJb6607S0dPHjQzeVgu6ptIvGqaEQOEjBjDkY+i4VoVjUSXS3
a8fiXB8moMdPtLJ/
GyE8Qiwbt4xbkQy48i02k99F2vGmlenRpSP6dD31kZ13dpJ0mGgq2yL/
eFaynKyqnjkHLMexcqD8WlVooaR11k2RPk5vDYvc8o2bn22OKKbOnSRM5Y1PS0i
Wzxo0JVcnVVyt0AANQQkv0ABBQQkVXaXR0TUQ1QW5kREVT</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="r00ABXNyABlqYXZheC5jcnlwdG8uU2VhbGVkt2JqZWN0PjY9psO3VHAC
AARbAA1lbmNvZGVkUGFyYW1zdAACW0JbABB1bmNyeXB0ZWRDb250ZW50cQB+AAF
MAAlwYXJhbXNBbGd0ABJMamF2YS9sYW5nL1N0cm1uZztMAAdzZWFsQWxncQB+AA
J4cHVyAAJbQqzzF/
gGCFtgAgAAeHAAAAAPMA0ECir9Z1iG+O3BAgEKdXEAfgAEAAAAuMU/
SoFRkebv2d5Sqa1F91ttjRY51yyW5+/
FIfoUdDyikYiDb0Jb6607S0dPHjQzeVgu6ptIvGqaEQOEjBjDkY+i4VoVjUSXS3
a8fiXB8moMdPtLJ/
GyE8QiwbT4xbkQy48i02k99F2vGmlenRpSP6dD31kZ13dpJ0mGgq2yL/
eFaynKyqjnkHLMexcqD8W1VooaR11k2Rpk5vDYvC8o2bn22OKKbOnSRM5Y1PS0i
Wzxo0JVcnVVyt0AANQQkV0ABBQQkVXaXR0TUQ1QW5kREVT">
        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">r0OABXNyABlqYXZheC5jcnlwdG8uU2VhbGVkT2JqZWN0PjY9psO3VHACA
ARbAA1lbnNvZGVkUGFyYW1zdAACW0JbABB1bnNyeXB0ZWRDb250ZW50cQB+AAFM
AAlwYXJhbXNBbGd0ABJMamF2YS9sYW5nL1N0cmluZztMAAdzZWFsQWxncQB+AAJ
4cHVyAAJbQqzzF/
gGCFtGAgAAeHAAAAAPMA0ECIr9Z1iG+O3BAgEKdXEAfgAEAAAAuMU/
SoFRkebvvh2d5SgalF91ttjRY51yyW5+/
FifOuDdYikYiDbOJb6607S0dPHjQzeVgu6ptIvGqaEQOEjBjDkY+i4VoVjUSXS3
a8fiXB8moMdPtLJ/
GyE8QiwbT4xbkQy48i02k99F2vGmlenRpSP6dD31kZl3dpJ0mGgq2yL/
eFaynKyqnjkiHLMexcqD8WlVooaRl1k2Rpk5vDYvC8o2bn22OKKbOnSRM5Y1PS0i
Wzxo0JVcnVVyt0AANQQkV0ABBQQkVXaXRoTUQ1QW5kREVT</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.xml` 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">rO0ABXNyABlqYXZheC5jcnlwdG8uU2VhbGVkt2JqZWN0PjY9psO3VHACA
ARbAA1lbmNvZGVkUGFyYW1zdAACW0JbABB1bmNyeXB0ZWRDb250ZW50cQB+AAFMAA
lWYXJhbXNBbGd0ABJMamF2YS9sYW5nL1N0cmcluZztMAAdzZWFsQWxncQB+AAJ
4cHVyAAJbQqzzF/
gGCFtgAgAAeHAAAAAPMA0ECir9Z1iG+O3BAgEKdXEAfgAEAAAAuMU/
SoFRkebv2d5Sqa1F91ttjRY51yyW5+/
FifOuDdYikYiDb0Jb6607S0dPHjQzeVgu6ptIvGqaEQOEjBjDkY+i4VoVjUSXS3
a8fiXB8moMdPtLJ/
GyE8QiwbT4xbkQy48i02k99F2vGmlenRpSP6dD31kZ13dpJ0mGgq2yL/
eFaynKyqjKHLMexcqD8W1VooaR11k2Rpk5vDYvC8o2bn22OKKbOnSRM5Y1PS0i
Wzxo0JVCnVVyt0AANQQkv0ABBQQkVXaXR0TUQ1QW5kREVT</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>

```

1. 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

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.xml` 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

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 `manultask_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&attribute=Telephone%20Number&value=555-1212
```

Replace *username.container* with the DN of a user in your tree.

Documentation Update



The documentation was updated on the following date:

- ♦ [Section K.1, “August 14, 2007,” on page 323](#)
- ♦ [Section K.2, “August 10, 2007,” on page 323](#)

K.1 August 14, 2007

The following section was updated:

K.1.1 Managing the Identity Manager Drivers

The following information was updated:

Location	Change
“Trace file size limit” on page 40	Added the note.
“Trace file size limit” on page 41	Added the note.

K.2 August 10, 2007

The following section was updated:

K.2.1 Options for Configuring a Remote Loader Appendix B

The following information was updated:

Location	Change
“com.novell.nds.dirxml.driver.jdbc.JDBCDSHIM” on page 262	Changed the entry from com.novell.nds.dirxml.jdbc.JDBCDSHIM to com.novell.nds.dirxml.driver.jdbc.JDBCDSHIM.