AUTHORIZED DOCUMENTATION

Driver for Active Directory* Implementation Guide

Novell®
Identity Manager

3.6

July 23, 2008

www.novell.com

Legal Notices

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

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

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

Novell, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed on the Novell Legal Patents Web page (http://www.novell.com/company/legal/patents/) and one or more additional patents or pending patent applications in the U.S. and in other countries.

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

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

Novell Trademarks

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

Third-Party Materials

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

Contents

About This Guide

This guide explains how to install, configure, and manage the Identity Manager Driver for Active Directory (Active Directory driver).

Audience

This guide is intended for Active Directory administrators, Novell® eDirectory™ administrators, and others who implement the Identity Manager driver for Active Directory.

Feedback

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

Documentation Updates

For the most recent version of this document, see the Novell Identity Manager Drivers Documentation Web site (http://www.novell.com/documentation/idm36drivers/index.html).

Additional Documentation

For documentation on using Identity Manager and the other Identity Manager drivers, see the Identity Manager Documentation Web site (http://www.novell.com/documentation/idm36).

Documentation Conventions

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

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

Overview

1

This section contains high-level information about how the Active Directory driver functions.

1.1 Key Terms

1.1.1 Identity Manager

Novell® Identity Manager is a service that synchronizes data among servers in a set of connected systems by using a robust set of configurable policies. Identity Manager uses the Identity Vault to store shared information, and uses the Metadirectory engine for policy-based management of the information as it changes in the vault or connected system. Identity Manager runs on the server where the Identity Vault and the Metadirectory engine are located.

1.1.2 Connected System

A connected system is any system that can share data with Identity Manager through a driver. Active Directory is a connected system.

1.1.3 Identity Vault

The Identity Vault is a persistent database powered by eDirectory™ and used by Identity Manager to hold data for synchronization with a connected system. The vault can be viewed narrowly as a private data store for Identity Manager or more broadly as a metadirectory that holds enterprise-wide data. Data in the vault is available to any protocol supported by eDirectory, including NCP™ (the traditional protocol used by such utilities as ConsoleOne® and iManager), LDAP, and DSML.

Because the vault is powered by eDirectory, Identity Manager can be easily integrated into your corporate directory infrastructure by using your existing directory tree as the vault.

1.1.4 Metadirectory Engine

The Metadirectory engine is the core server that implements the event management and policies of Identity Manager. The engine runs on the Java* Virtual Machine in eDirectory.

1.1.5 Active Directory Driver

A driver implements a data sharing policy for a connected system. You control the actions of the driver by using iManager to define the filters and the policy. For Active Directory, a driver implements the policy for a single domain.

1.1.6 Driver Shim

A driver shim is the component of a driver that converts the XML-based Identity Manager command and event language (XDS) to the protocols and API calls needed to interact with a connected system. The shim is called to execute commands on the connected system after the Output Transformation runs. Commands are usually generated on the Subscriber channel but can be generated by command write-back on the Publisher channel.

The shim also generates events from the connected system for the Input Transformation policy. A driver shim can be implemented either in Java class or as a native Windows* DLL file. The shim for Active Directory is `ADDriver.dll`.

`ADDriver.dll` is implemented as a native Windows DLL file. ADDriver uses several different Windows APIs to integrate with Active Directory. These APIs typically require some type of login and authentication to succeed. Also, the APIs might require that the login account have certain rights and privileges within Active Directory and on the machine where `ADDriver.dll` executes.

If you use the Remote Loader, `ADDriver.dll` executes on the server where the Remote Loader is running. Otherwise, it executes on the server where the Metadirectory engine is running.

1.1.7 Remote Loader

A Remote Loader enables a driver shim to execute outside of the Metadirectory engine (perhaps remotely on a different machine). The Remote Loader is typically used when a requirement of the driver shim is not met by the Identity Manager server. For example, if the Metadirectory engine is running on Linux*, the Remote Loader is used to execute the Active Directory driver shim on a Windows server.

The Remote Loader is a service that executes the driver shim and passes information between the shim and the Metadirectory engine. When you use a Remote Loader, you install the driver shim on the server where the Remote Loader is running, not on the server where the Metadirectory engine is running. You can choose to use SSL to encrypt the connection between the Metadirectory engine and the Remote Loader. For more information, see "Remote Loader" in the *Identity Manager 3.6 Installation Guide*.

When you use the Remote Loader with the Active Directory driver shim, two network connections exist:

- Between the domain controller and the Remote Loader
- Between Active Directory and the Active Directory driver shim

1.2 Data Transfers Between Systems

Data flows between Active Directory and the Identity Vault by using the Publisher and Subscriber channels.

The Active Directory driver supports Publisher and Subscriber channels.

The Publisher channel does the following:

- Reads events from Active Directory for the domain hosted on the server that the driver shim is connecting to.
- Submits that information to the Identity Vault.

The Subscriber channel does the following:

- Watches for additions and modifications to the Identity Vault objects.
- Makes changes to Active Directory that reflect those changes.

You can configure the driver so that both Active Directory and the Identity Vault are allowed to update a specific attribute. In this configuration, the most recent change determines the attribute value, except in the case of merge operations that are controlled by the filters and merge authority.

1.3 Key Driver Features

The sections below contains a list of the key driver features.

1.3.1 Local Platforms

A local installation is an installation of the driver on the Metadirectory server. The Active Directory driver can be installed on the Windows operating systems supported for the Metadirectory server. The supported operating system version is Windows Server* 2003 SP2 (32-bit).

For more information about local installations, see Section 2.2, “Where to Install the Active Directory Driver,” on page 19.

For additional information about system requirements, see “Metadirectory Server” in “System Requirements” in the *Identity Manager 3.6 Installation Guide*.

1.3.2 Remote Platforms

The Active Directory driver can use the Remote Loader service to run on a Windows server other than the Metadirectory server. The Remote Loader service for the Active Directory driver can be installed on Windows Server 2003 SP2 (32-bit and 64-bit) and Windows Server 2008 (64-bit).

For more information about remote installations, see Section 2.2, “Where to Install the Active Directory Driver,” on page 19.

For additional information about system requirements, see “Remote Loader” in “System Requirements” in the *Identity Manager 3.6 Installation Guide*.

1.3.3 Entitlements

The Active Directory driver implements entitlements.

Entitlements make it easier to integrate Identity Manager with the Identity Manager User Application and Role-Based Services in eDirectory. In the User Application, an action such as provisioning an account in Active Directory is delayed until the proper approvals have been made. In Role-Based Services, rights assignments are made based on attributes of a user object and not by regular group membership. Both of these services offer a challenge to Identity Manager because it is not obvious from the attributes of an object whether an approval has been granted or the user matches a role.

Entitlements standardize a method of recording this information on objects in the Identity Vault. From the driver perspective, an entitlement grants or revokes the right to something in Active Directory. You can use entitlements to grant the right to an account in Active Directory, to control group membership, and to provision Exchange mailboxes. The driver is unaware of the User Application or Role-Based Entitlements. It depends on the User Application server or the Entitlements driver to grant or revoke the entitlement for a user based upon its own rules.

You should enable entitlements for the driver only if you plan to use the User Application or Role-Based Entitlements with the driver. For more information about entitlements, see the *Identity Manager 3.6 Entitlements Guide*.

1.3.4 Password Synchronization Support

The Active Directory driver synchronizes passwords on both the Subscriber channel and the Publisher channel. For more information, see Chapter 6, “Synchronizing Passwords,” on page 45.

1.3.5 Data Synchronization Support

The Active Directory driver synchronizes User objects, Group objects, containers, and Exchange mailboxes.

1.4 Default Driver Configuration

The Active Directory driver is shipped with a default configuration file called `ActiveDirectory-IDM3_6_0-V4.xml`. When imported with Designer or iManager, this configuration file creates a driver with a set of policies and rules suitable for synchronizing with Active Directory. If your requirements for the driver are different from the default policies, you need to modify the default policies to do what you want. Pay close attention to the default Matching

policies. The data that you trust to match users usually is different from the default. The policies themselves are commented and you can gain a greater understanding of what they do by importing a test driver and reviewing the policies with Designer or iManager.

1.4.1 User Object Name Mapping

Management utilities for the Identity Vault, such as iManager and ConsoleOne, typically name user objects differently than the Users and Computers snap-in for the Microsoft Management Console (MMC). Make sure that you understand the differences so the Matching policy and any Transformation policies you have are implemented properly.

1.4.2 Data Flow

Data flow between Active Directory and the Identity Vault is controlled by the filters, mappings, and policies that are in place for the Active Directory driver.

Filters

The driver filter determines which classes and attributes are synchronized between Active Directory and the Identity Vault, and in which direction synchronization takes place.

Schema Mapping

Table 1-1 through Table 1-6 list Identity Vault user, group, and Organizational Unit attributes that are mapped to Active Directory user and group attributes.

The mappings listed in the tables are default mappings. You can remap same-type attributes.

Table 1-1 *Mapped User Attributes*

eDirectory - User	Active Directory - user
DirXML-ADAliasName	sAMAccountName

eDirectory - User	Active Directory - user
L	PhysicalDeliveryOfficeName
Physical Delivery Office Name	l
nspmDistributionPassword	nspmDistributionPassword

Table 1-2 *Mapped Group Attributes*

eDirectory - Group	Active Directory - group
DirXML-ADAliasName	sAMAccountName

eDirectory's L attribute is mapped to Active Directory's physicalDeliveryOfficeName attribute, and eDirectory's Physical Delivery Office Name attribute is mapped to Active Directory's L attribute. Because similarly named fields have the same value, mapping the attributes this way enables the attributes to work well with ConsoleOne and the Microsoft Management Console.

Table 1-3 *Mapped Organizational Unit Attributes*

eDirectory - Organizational Unit	Active Directory - organizationalUnit
L	physicalDeliveryOfficeName
Physical Delivery Office Name	l

Table 1-4 *Mapped Organization Attributes*

eDirectory - Organization	Active Directory - organization
L	physicalDeliveryOfficeName
Physical Delivery Office Name	l

The driver maps the Locality class, but there are no attributes for the class.

Table 1-5 *Mapped Locality Class*

eDirectory	Active Directory
Locality	locality

Table 1-6 *Mapped Non-Class Specific Attributes*

eDirectory	Active Directory
CN	cn
Description	description

eDirectory	Active Directory
DirXML-EntitlementRef	DirXML-EntitlementRef
DirXML-EntitlementResult	DirXML-EntitlementResult
Facsimile Telephone Number	facsimiletelephoneNumber
Full name	displayName
Given Name	givenName
Group Membership	memberOf
Initials	initials
Internet EMail Address	mail
Login Allowed Time Map	logonHours
Login Disabled	dirxml-uACAccountDisabled
Login Expiration Time	accountExpires
Login Intruder Reset Time	lockoutTime
Member	member
OU	ou
Owner	managedBy
Postal Code	PostalCode
Postal Office Box	postOfficeBox
S	st
SA	streetAddress
See Also	seeAlso
DirXML-SPEntitlements	DirXML-SPEntitlements
Surname	sn
Telephone Number	telephoneNumber
Title	title

Name Mapping Policies

The default configuration includes two name mapping policies that work together to help you reconcile different naming policies between the Identity Vault and Active Directory. When you create a user with the Active Directory Users and Computers tool (a snap-in for the Microsoft Management Console and abbreviated as MMC in this document) you see that the user full name is used as its object name. Attributes of the user object define Pre-Windows 2000 Logon Name (also known as the NT Logon Name or sAMAccountName) and the Windows 2000 Logon Name (also known as the userPrincipalName). When you create a user in the Identity Vault with iManager or ConsoleOne, the object name and the user logon name are the same.

If you create some users in Active Directory using MMC and other objects in the Identity Vault or another connected system that is synchronized with the Identity Vault, the object can look odd in the opposing console and might fail to be created in the opposing system at all.

The Full Name Mapping Policy is used to manage objects in Active Directory by using the MMC conventions. When it is enabled, The Full Name attribute in the Identity Vault is synchronized with the object name in Active Directory.

The NT Logon Name Mapping Policy is used to manage objects in Active Directory by using the Identity Vault conventions. When it is enabled, the Identity Vault object name is used to synchronize both the object name and NT Logon Name in Active Directory. Objects in Active Directory have the same names as the Identity Vault, and the NT Logon Name matches the Identity Vault logon name.

When both of the policies are enabled at the same time, the Active Directory object name is the Identity Vault Full Name, but the NT Logon Name matches the Identity Vault logon name.

When both policies are disabled, no special mapping is made. The object names are synchronized and there are no special rules for creating the NT Logon Name. Because the NT Logon Name is a mandatory attribute in Active Directory, you need some method of generating it during Add operations. The NT Logon Name (sAMAccountName) is mapped to the DirMXL-ADAliasName in the Identity Vault, so you could either use that attribute to control the NT Logon Name in Active Directory or you could build your own policy in the Subscriber Create policies to generate one. With this policy selection, users created with MMC use the object name generated by MMC as the object name in the Identity Vault. This name might be inconvenient for login to the Vault.

Use of the Name Mapping policies is controlled through Global Configuration Values. For information, see Section A.2, “Global Configuration Values,” on page 80.

Active Directory Logon Name Policies

The Windows 2000 Logon name (also known as the userPrincipalName or UPN) does not have a direct counterpart in the Identity Vault. UPN looks like an e-mail address (user@mycompany.com) and might in fact be the user’s e-mail name. The important thing to remember when working with UPN is that it must use a domain name (the part after the @ sign) that is configured for your domain. You can find out what domain names are allowed by using MMC to create a user and inspecting the domain name drop-down box when adding the UPN.

The default configuration offers several choices for managing userPrincipalName. If your domain is set up so that the user’s e-mail address can be used as a userPrincipalName, one of the options to track the user’s e-mail address is appropriate. You can make userPrincipalName follow either the Identity Vault or Active Directory e-mail address, depending on which side is authoritative for e-mail. If the user e-mail address is not appropriate, you can choose to have a userPrincipalName constructed from the user logon name plus a domain name. If more than one name can be used, update the policy after import to make the selection. If none of these options are appropriate, then you can disable the default policies and write your own.

Use of the Active Directory Logon Name policy is controlled through Global Configuration Values. For information, see Section A.2, “Global Configuration Values,” on page 80.

Preparing Active Directory

2

In this section:

2.1 Active Directory Prerequisites

- ❑ Novell® Identity Manager 3.6 and its prerequisites, as listed in “System Requirements” in the *Identity Manager 3.6 Installation Guide*.
- ❑ Windows Server 2003 SP2 (32-bit) or Windows Server 2008 (64-bit).
- ❑ Internet Explorer* 5.5 or later on the server running the Active Directory (AD) driver and on the target domain controller.
- ❑ Active Directory domain controller DNS name or IP address, depending on the authentication method.

Also, we recommend that the server hosting the Active Directory driver be a member of the Active Directory domain. This is required to provision Exchange mailboxes and synchronize passwords. If you don’t require these features, the server can be a member of any domain as long as the Simple (simple bind) authentication mode is used. To have bidirectional password synchronization, the Negotiate authentication option must be selected.

If you want to synchronize with an ADAM instance, see Appendix B, “Configuring the Driver for Use with an ADAM Instance,” on page 87 for more information.

If you want to synchronize Exchange accounts, see Appendix D, “Provisioning Exchange Accounts,” on page 99.

2.2 Where to Install the Active Directory Driver

The Active Directory driver shim must run on one of the supported Windows platforms. However, you don’t need to install the Metadirectory engine on this same machine. Using a Remote Loader, you can separate the engine and the driver shim, allowing you to balance the load on different machines or accommodate corporate directives.

The installation scenario you select determines how the driver shim is installed. If you choose to install the driver shim on the same machine as Identity Manager (where the Metadirectory engine and the Identity Vault are located), Identity Manager calls the driver shim directly. If you choose to install the driver shim on another machine, you must use the Remote Loader.

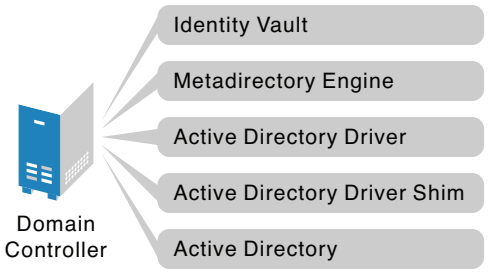
You can install the Active Directory driver on either the domain controller or a member server. Before you start the driver installation, determine where you want to install the driver.

- Section 2.2.2, “Remote Installation on Windows Server Only,” on page 20
- Section 2.2.3, “Remote Installation on Windows and Other Platforms,” on page 21
- Section 2.2.4, “Remote Installation on a Windows Member Server,” on page 21

2.2.1 Local Installation

A single Windows domain controller can host the Identity Vault, the Metadirectory engine, and the driver.

Figure 2-1 *All Components on the Domain Controller*



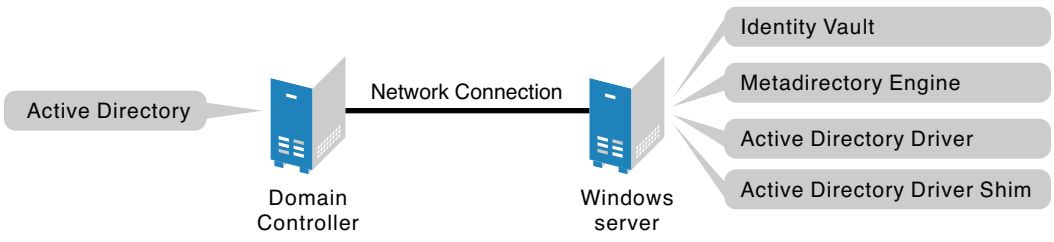
This configuration works well for organizations that want to save on hardware costs. It is also the highest-performance configuration because there is no network traffic between Identity Manager and Active Directory.

However, hosting Identity Vault and the Metadirectory engine on the domain controller increases the overall load on the controller and increases the risk that the controller might fail. Because domain controllers play a critical role in Microsoft networking, many organizations are more concerned about the speed of the domain authentication and the risks associated with a failure on the domain controller than about the cost of additional hardware.

2.2.2 Remote Installation on Windows Server Only

You can install the Identity Vault, the Metadirectory engine, and the driver on a separate computer from the Active Directory domain controller. This configuration leaves the domain controller free of any Identity Manager software.

Figure 2-2 *All Components on a Windows Server*

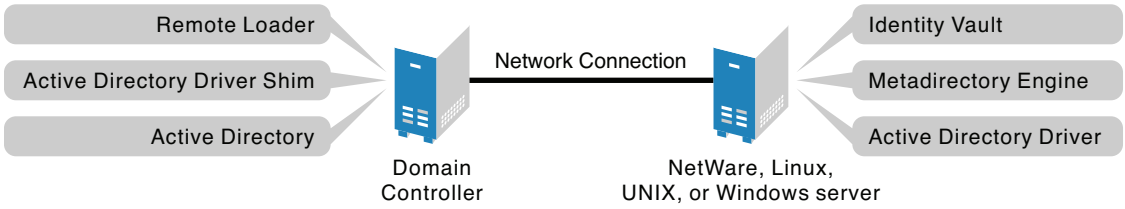


This configuration is attractive if corporate policy disallows running the driver on your domain controller.

2.2.3 Remote Installation on Windows and Other Platforms

You can install the Remote Loader and driver shim on the Active Directory domain controller, but install the Identity Vault and the Metadirectory engine on a separate server.

Figure 2-3 *Remote Loader and Driver on the Domain Controller*



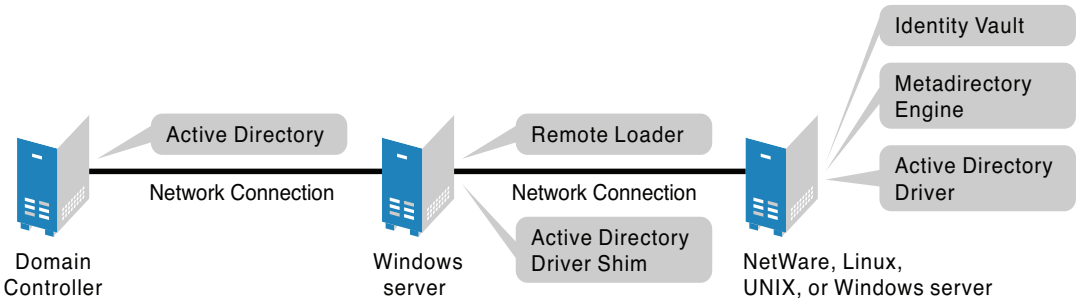
This configuration is attractive if your Identity Vault and Metadirectory engine (Identity Manager) installations are on a platform other than one of the supported versions of Windows.

Both Scenario 2 and Scenario 3 configurations eliminate the performance impact of hosting the Identity Vault and the Metadirectory engine on the domain controller.

2.2.4 Remote Installation on a Windows Member Server

If you have platform requirements and domain controller restrictions in place, you can use a three-server configuration.

Figure 2-4 *Remote Loader and Driver on a Windows Server*



This configuration is more complicated to set up, but it accommodates the constraints of some organizations. In this figure, the two Windows servers are member servers of the domain.

2.3 Addressing Security Issues

The major security issues to consider are authentication, encryption, and use of the Remote Loader. You might want to consider a security option called signing. See “Digitally sign communications” on page 64 in Chapter 9, “Security Best Practices,” on page 63.

A simple prescription for managing security is not possible because the security profile available from Windows varies with the service pack, DNS server infrastructure, domain policy, and local policy settings on the server. The following sections explain your security choices and provide suggested configurations. When implementing your driver and when upgrading components, pay close attention to security.

2.3.1 Authentication Methods

Authentication identifies the driver shim to Active Directory and, potentially, the local machine. To authenticate to Active Directory, you can use either the Negotiate method or the Simple (simple bind) method.

Table 2-1 *Authentication Methods*

Authentication Method	Description	Advantages	Disadvantages
Negotiate	The preferred method. Uses Kerberos*, NTLM, or a pluggable authentication scheme if one is installed.	The driver can be installed on any server in the domain.	The server hosting the driver must be a member of the domain.
Simple	Used when the server hosting the driver shim is not a member of the domain.	The driver can be installed on a server that is not a member of the domain.	Some provisioning services are unavailable, such as Exchange mailbox provisioning and password synchronization.

2.3.2 Encryption Using SSL

SSL encrypts data. Depending on your configuration, SSL can be used in two places:

- Between the Active Directory driver and the domain controller
- Between the Identity Vault and the Remote Loader running the Active Directory driver

Password synchronization occurs between Active Directory and the Identity Vault. You need to make sure that you use SSL with any communication that goes across the network.

If the Metadirectory engine, Identity Vault, the Active Directory driver, and Active Directory are on the same machine, you don’t need SSL. Communication isn’t going across the network.

However, if you are accessing Active Directory remotely by using an Active Directory driver shim on a member server, you need to set up SSL between the Active Directory driver shim and Active Directory. You do this by setting the SSL parameter to *Yes* on the driver configuration. See Step 5 on page 24, in Section , “SSL Connection between the Active Directory Driver and the Domain Controller,” on page 23.

If you are using the Remote Loader on the Domain Controller, you can set up SSL between the Metadirectory engine and the Remote Loader. For additional information on SSL and Remote Loaders, see "Creating a Secure Connection" in the *Identity Manager 3.6 Remote Loader Guide*.

The following table outlines where SSL connections can be used for each of the scenarios discussed in Section 2.2, "Where to Install the Active Directory Driver," on page 19:

Table 2-2 *SSL Connects*

Configuration	SSL Connections Available
Single-Server	No SSL connections are necessary.
Two-Server: Identity Manager and the Active Directory driver are on the same server	An SSL connection can be established between the Active Directory driver and the domain controller.
Dual-Server: Identity Manager is on one server but the Active Directory driver is on a separate server	An SSL connection can be established between Identity Manager and the Remote Loader running the Active Directory driver.
Three-Server	An SSL connection can be established between the Active Directory driver and the domain controller. An SSL connection can also be established between Identity Manager and the Remote Loader running the Active Directory driver.

- "SSL Connection between the Active Directory Driver and the Domain Controller" on page 23
- "SSL Connection Between the Remote Loader and Identity Manager" on page 25

SSL Connection between the Active Directory Driver and the Domain Controller

To make SSL connections to an Active Directory domain controller, you must be set up to use SSL. This involves setting up a certificate authority, then creating, exporting, and importing the necessary certificates.

- "Setting Up a Certificate Authority" on page 23
- "Creating, Exporting, and Importing Certificates" on page 24
- "Verifying the Certificate" on page 25

Setting Up a Certificate Authority

Most organizations already have a certificate authority. If this is the case for your organization, you need to export a valid certificate, then import it to the certificate store on your domain controller. The server hosting the driver shim must trust the root certificate authority that the issuing certificate authority of this certificate chains to.

If you do not have a certificate authority in your organization, you must establish one. Novell, Microsoft*, and several other third parties provide the tools necessary to do this. Establishing a certificate authority is beyond the scope of this guide. For more information about the Novell solution, see the *Novell Certificate Server 3.3 Administration Guide* (http://www.novell.com/documentation/lg/crt33/index.html).

Creating, Exporting, and Importing Certificates

After you have a certificate authority, the LDAP server must have the appropriate server authentication certificate installed, for LDAP SSL to operate successfully. Also, the server hosting the driver shim must trust the authority that issued those certificates. Both the server and the client must support 128-bit encryption.

1. Generate a certificate that meets the following Active Directory LDAP service requirements:
   - The LDAPS certificate is located in the Local Computer's Personal certificate store (programmatically known as the computer's MY certificate store).
   - A private key matching the certificate is present in the Local Computer's store and is correctly associated with the certificate.

     The private key must not have strong private-key protection enabled.
   - The Enhanced Key Usage extension includes the Server Authentication (1.3.6.1.5.5.7.3.1) object identifier (also known as OID).
   - The Active Directory fully qualified domain name (for example, DC01.DOMAIN.COM) of the domain controller appears in one of the following places:
     - The Common Name (CN) in the Subject field.
     - The DNS entry in the Subject Alternative Name extension.
   - The certificate was issued by a CA that the domain controller and the LDAPS clients trust.

     Trust is established by configuring the clients and the server to trust the root CA that the issuing CA chains to.

   This certificate permits the LDAP service on the domain controller to listen for and automatically accept SSL connections for both LDAP and global catalog traffic.

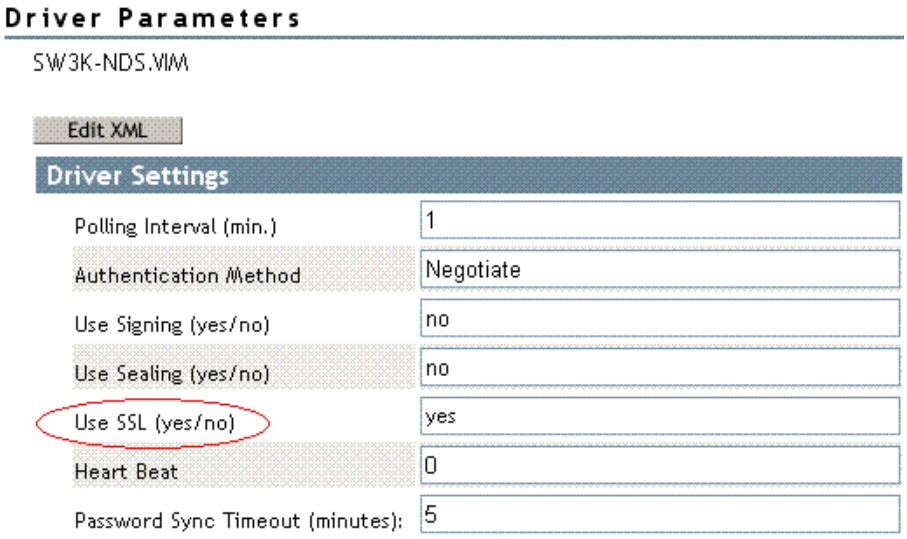
   NOTE: This information appears in the Microsoft Knowledge Base Article 321051, How to Enable LDAP over SSL with a Third-Party Certificate Authority (http://support.microsoft.com/default.aspx?scid=kb;en-us;321051). Consult this document for the latest requirements and additional information.

   ---

2. Export this certificate in one of the following standard certificate file formats:
   - Personal Information Exchange (PFX, also called PKCS #12)
   - Cryptographic Message Syntax Standard (PKCS #7)
   - Distinguished Encoding Rules (DER) Encoded Binary X.509
   - Base64 Encoded X.509
3. Install this certificate on the domain controller.
4. Ensure that a trust relationship is established between the server hosting the driver shim and the root certificate authority that issued the certificate.

   The server hosting the driver shim must trust the root certificate authority that the issuing certificate authority chains to.

   For more information on establishing a trust for certificates, see the *Policies to establish trust of root certification authorities* topic in *Windows Server 2003 Help*.
5. In iManager, edit the driver properties and change the *Use SSL (yes/no)* option to yes.



6 Restart the driver.

When the driver restarts, an SSL connection is negotiated between the domain controller and the server running the Active Directory driver shim.

Verifying the Certificate

To verify the certificate, authenticate to Active Directory via SSL. Use the ldifde command line utility found on Windows servers. To use the ldifde command:

1 Open a command line prompt

2 Enter `ldifde -f output/input file -t 636 -b administrator domain password -s computerFullName`

Here is an example of what you would enter if your server is configured for port 636.

```
ldifde -f out.txt -t 636 -b administrator dxad.novell.com novell -s
parent1.dxad3.lab.novell
```

The output is sent to the `out.txt` file. If you open the file and see the objects in Active Directory listed, you made a successful SSL connection to Active Directory and the certificate is valid.

SSL Connection Between the Remote Loader and Identity Manager

If you are using the Remote Loader, you need to set up SSL between the Metadirectory engine and the Remote Loader, and configure the settings between the driver and Active Directory.

For information on establishing an SSL connection between the Remote Loader and Identity Manager, see "Creating a Secure Connection" in the *Identity Manager 3.6 Remote Loader Guide*.

WARNING: When the Remote Loader is running on a Windows 2003 SP2 32-bit server, the certificate must be in Base64 format. If you use the DER format, the Remote Loader fails to connect to the Identity Manager engine.

2.4 Creating an Administrative Account

In a test environment, use the Administrator account until you get the Active Directory driver working. Then create an administrative account that has the proper rights (including restricted rights) for the Active Directory driver to use exclusively to authenticate to Active Directory.

Doing this keeps the Identity Manager administrative account insulated from changes to other administrative accounts. Advantages to this design are:

- You can use Active Directory auditing to track the activity of the Active Directory driver.
- You can implement a password change policy as with other accounts, then make necessary updates to the driver configuration.

This account name and password are stored in the driver configuration. Therefore, you must change this password whenever the account password changes. If you change the account password without updating the driver configuration, authentication fails the next time the driver is restarted.

At a minimum, this account must have Read and Replicating Directory Changes rights at the root of the domain for the Publisher channel to operate. You also need Write rights to any object modified by the Subscriber channel. Write rights can be restricted to the containers and attributes that are written by the Subscriber channel.

To provision Exchange mailboxes, your Identity Manager account must have "Act as part of the Operating System" permission for the logon account.

Windows Server 2003 requires that you have additional rights in order to see deleted objects. See Appendix C, "Changing Permissions on the CN=Deleted Objects Container," on page 97.

2.5 Becoming Familiar with Driver Features

This section discusses driver features you should become familiar with before deploying the Active Directory driver.

2.5.1 Multivalue Attributes

The way the Active Directory driver handles multivalue attributes has changed from version 2.

Version 2 treated multivalue attributes as single-valued on the Subscriber channel by ignoring all but the first change value in an Add or Modify operation. Version 3 of the Active Directory Driver fully supports multivalue attributes.

However, when the Active Directory driver synchronizes a multivalue attribute with a single-value attribute, the multivalue attribute is treated as single-valued. For example, the Telephone Number attribute is single-valued in Active Directory, and multivalue in the Identity Vault. When this attribute is synchronized from Active Directory, only a single value is stored in the Identity Vault.

This creates true synchronization and mapping between the two attributes, but can result in a potential loss of data if you have multiple values in an attribute that is mapped to an attribute with a single value. In most cases, a policy can be implemented to preserve the extra values in another location if this is required in your environment.

2.5.2 Using Custom Boolean Attributes to Manage Account Settings

The Active Directory attribute userAccountControl is an integer whose bits control logon account properties, such as whether logon is allowed, passwords are required, or the account is locked. Synchronizing the Boolean properties individually is difficult because each property is embedded in the integer value.

In version 2, the Active Directory driver took a shortcut that let you map userAccountControl to the eDirectory Login Disabled attribute, but didn't let you map the other property bits within the attribute.

In version 3, each bit within the userAccountControl attribute can be referenced individually as a Boolean value, or userAccountControl can be managed in-total as an integer. The driver recognizes a Boolean alias to each bit within userAccountControl. These alias values are included in the schema for any class that includes userAccountControl. The alias values are accepted on the Subscriber channel and are presented on the Publisher channel.

The advantage of this is that each bit can be used as a Boolean, so the bit can be enabled individually in the Publisher filter and accessed easily. You can also put userAccountControl into the Publisher filter to receive change notification as an integer.

The integer and alias versions of userAccountControl should not be mixed in a single configuration.

The following table lists available aliases and hexadecimal values. Read-only attributes cannot be set on the Subscriber channel.

Table 2-3 *Aliases and Hexadecimal Values*

Alias	Hexadecimal	Notes
dirxml-uACDontExpirePassword	0x10000	Read-write
dirxml-uACHomedirRequired	0x0008	Read-write
dirxml-uACInterdomainTrustAccount	0x0800	Read-only
dirxml-uACNormalAccount	0x0200	Read-only
dirxml-uACServerTrustAccount	0x2000	Read-only
dirxml-uACWorkstationTrustAccount	0x1000	Read-only
dirxml-uACAccountDisable	0x0002	Read-write
dirxml-uACPasswordNotRequired	0x0020	Read-write

For troubleshooting tips relating to the userAccountControl attribute, see Section 10.8, "The Active Directory Account Is Disabled after a User Add on the Subscriber Channel," on page 70.

2.5.3 Provisioning Exchange Mailboxes

The Active Directory driver can be configured to provision Exchange accounts as well as Active Directory accounts. The Active Directory driver can provision Exchange 2000, Exchange 2003, and Exchange 2007 accounts. For information on configuring the driver to provision the Exchange mailboxes, see Appendix D, “Provisioning Exchange Accounts,” on page 99.

2.5.4 Expiring Accounts in Active Directory

If you map the eDirectory attribute of Login Expiration Time to the Active Directory attribute of accountExpires, an account in Active Directory expires a day earlier than the time set in eDirectory.

This happens because Active Directory sets the value of the accountExpires attribute in full-day increments. The eDirectory attribute of Login Expiration Time uses a specific day and time to expire the account.

For example, if you set an account in eDirectory, to expire on July 15, 2007, at 5:00 p.m., the last full day this account is valid in Active Directory is July 14.

If you use the Microsoft Management Console to set the account to expire on July 15, 2007, the eDirectory attribute of Login Expiration Time is set to expire on July 16, 2007 at 12:00 a.m. Because the Microsoft Management Console doesn’t allow for a value of time to be set, the default is 12:00 a.m.

The driver uses the most restrictive settings. You can add an additional day to the expiration time in Microsoft depending upon what your requirements are.

2.5.5 Retaining eDirectory Objects When You Restore Active Directory Objects

Any Active Directory objects that are restored through the Active Directory tools delete the associated eDirectory™ object when the objects are synchronized. The Active Directory driver looks for a change in the isDeleted attribute on the Active Directory object. When the driver detects a change in this attribute, a Delete event is issued through the driver for the object associated with the Active Directory object.

If you don’t want eDirectory objects deleted, you must add an additional policy to the Active Directory driver. Identity Manager 3.6 comes with a predefined rule that changes all Delete events into Remove Association events. For more information, see “Command Transformation - Publisher Delete to Disable ” in the *Policies in Designer 3.0* guide.

Installing the Driver Files

3

There are several locations where you can install the driver files. You should review Section 2.2, “Where to Install the Active Directory Driver,” on page 19.

By default, the Active Directory driver files are installed on the Metadirectory server at the same time as the Metadirectory engine. The installation program extends the Identity Vault’s schema and installs the driver shim, the driver configuration file, and a utility to help with the configuration of the driver. It does not create the driver in the Identity Vault (see Chapter 4, “Creating a New Driver,” on page 33) or upgrade an existing driver’s configuration (see Chapter 5, “Upgrading an Existing Driver,” on page 43).

The following sections explain what to do if the Active Directory driver files are not on the server you want and how to install the Active Directory Discovery tool (used to gather configuration information) on the appropriate Active Directory server:

3.1 Installing the Driver Files

If you performed a custom installation and did not install the Active Directory driver on the Metadirectory server, you have two options:

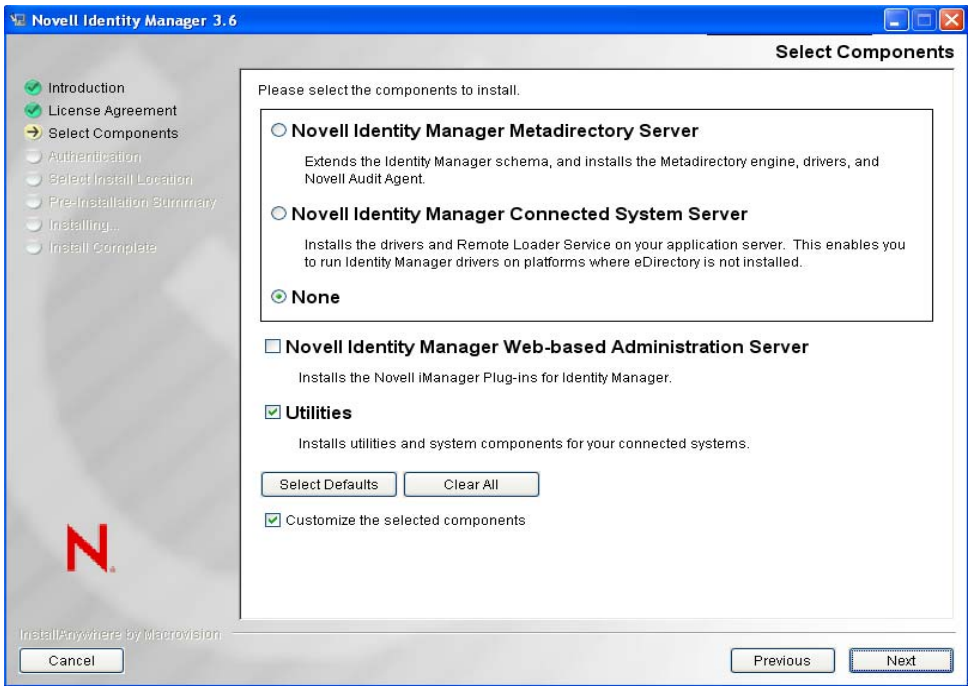
- Install the files on the Metadirectory server, using the instructions in “Installing the Metadirectory Server” in the *Identity Manager 3.6 Installation Guide*.
- Install the Remote Loader (required to run the driver on a non-Metadirectory server) and the driver files on a non-Metadirectory server where you want to run the driver. This is the method you should use if you do not want to install eDirectory™ and Identity Manager on the server that has Active Directory installed on it. See “Installing the Remote Loader” in the *Identity Manager 3.6 Installation Guide*.

If you decide to use the Remote Loader and install it on a member server, you must configure the driver to use an SSL connection between the Remote Loader and the Identity Manager server. For more information on how to set up an SSL connection, see “Creating a Secure Connection” in the *Identity Manager 3.6 Remote Loader Guide*.

3.2 Installing the Active Directory Discovery Tool

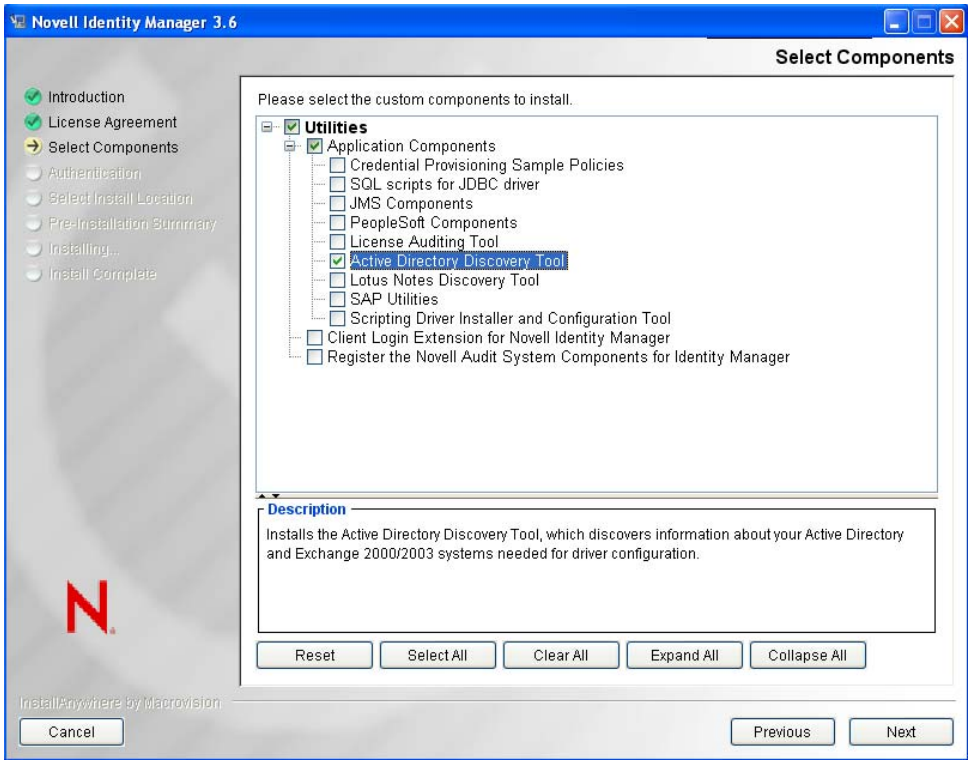
The Active Directory Discovery tool helps gather information required to configure the driver. To install the Active Directory Discovery tool:

1. On the workstation that you use to configure Active Directory, launch the Identity Manager installation.
2. In the Welcome dialog box, click *Next*, accept the license agreement, then click *Next* to display the Select Components page.



3 In the *Please select the components to install* section, select *None*.

4 Select both the *Utilities* option and the *Customize the selected components* options, then click *Next*.



5 Deselect all components except for *Active Directory Discovery Tool*, then click *Next*.

6 Specify the installation path (the default is sufficient), then click *Next*.

7 Review the selected options, then click *Install*.

Creating a New Driver

4

After the Active Directory driver files are installed on the server where you want to run the driver (see Chapter 3, "Installing the Driver Files," on page 29), you can create the driver in the Identity Vault. You do so by importing the basic driver configuration file and then modifying the driver configuration to suit your environment. The following sections provide instructions:

4.1 Gathering Configuration Information

The Active Directory Discovery Tool gathers the information needed to configure the Active Directory driver. The tool gathers a list of the domain controllers and Microsoft Exchange private message stores available in the domain and optionally creates an account in Active Directory suitable for the driver.

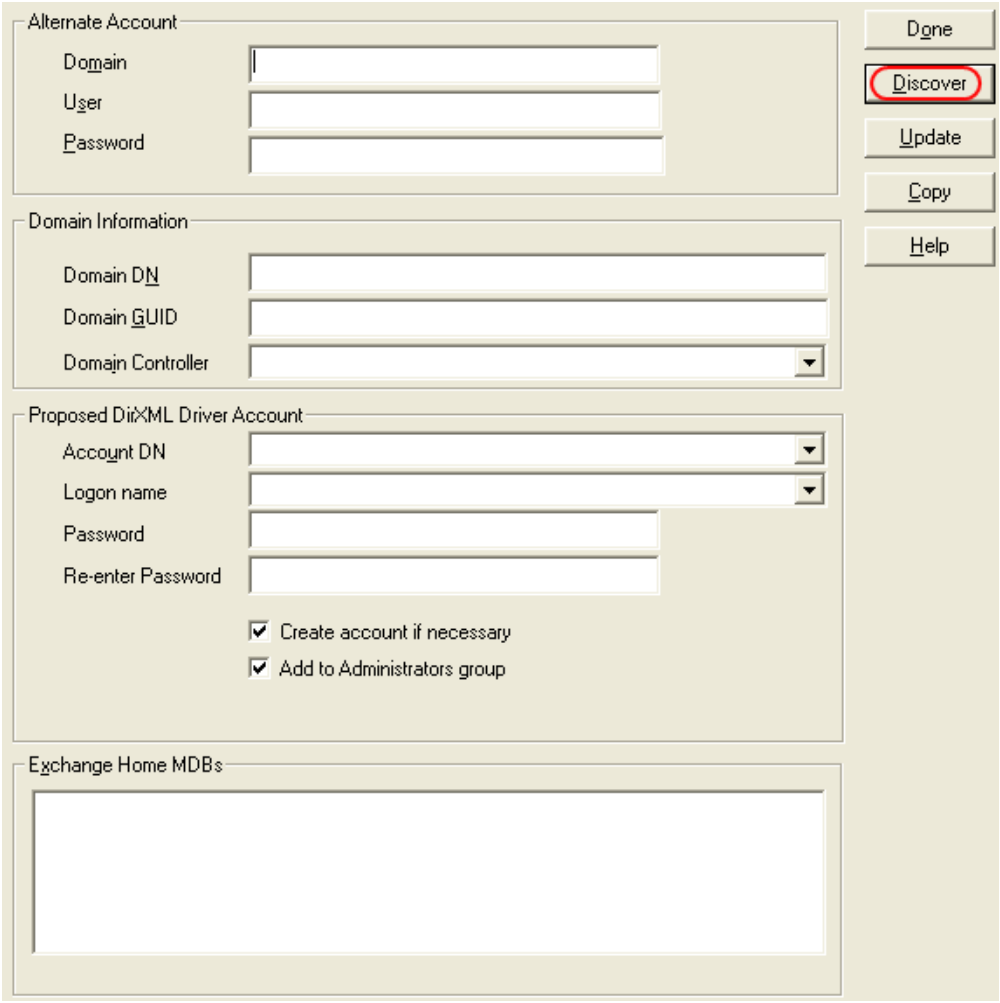
To run the tool:

1. On the workstation where you installed the tool, double-click the following file: `C:\Novell\NDS\DirXMLUtilities\ad_disc\ADManager.exe`.

   This is the default installation location for the file.
2. Click *Discover* to populate the tool with your domain information.

   The tool lists the following information:

   - Domain DN
   - Domain GUID
   - Domain Controller name
   - Proposed driver account name and password
   - Exchange Home MDB attribute



3 If you want to see information for another domain, specify the domain name, a user with sufficient rights to look up domain information, and that user's password, then click *Discover*.

4.2 Creating the Driver in Designer

You create the Active Directory driver by importing the driver's basic configuration file and then modifying the configuration to suit your environment. After you create and configure the driver, you need to deploy it to the Identity Vault and start it.

- Section 4.2.1, "Importing the Driver Configuration File," on page 34
- Section 4.2.2, "Configuring the Driver," on page 36
- Section 4.2.3, "Deploying the Driver," on page 36
- Section 4.2.4, "Starting the Driver," on page 37

4.2.1 Importing the Driver Configuration File

1 In Designer, open your project.

2 In the Modeler, right-click the driver set where you want to create the driver, then select *New* > *Driver* to display the Driver Configuration Wizard.

3 In the Driver Configuration list, select *Active Directory*, then click *Run*.

4 On the Import Information Requested page, fill in the following fields:

Driver Name: Specify a name that is unique within the driver set.

Connected System or Driver Name: Specify the name of the connected system or Identity Manager driver.

Domain DNS Name: Specify the DNS name of the Active Directory domain managed by this driver.

Active Directory User Container: Specify the container where user objects reside in Active Directory.

User Container: Select the Identity Vault container where Active Directory users will be added if they don't already exist in the vault. This value becomes the default for all drivers in the driver set. If you don't want to change this value for all drivers, leave this field unchanged and change the value on the driver's Global Configuration Values page after you've finished importing the driver.

Driver is Local/Remote: Select *Local* if this driver will run on the Metadirectory server without using the Remote Loader service. Select *Remote* if you want the driver to use the Remote Loader service, either locally on the Metadirectory server or remotely on another server.

Authentication ID: Specify an Active Directory account with administrative privileges to be used by Identity Manager. The form of the name used depends on the selected authentication mechanism.

For *Negotiate*, provide the name form required by your Active Directory authentication mechanism. For example:

- Administrator: AD Logon Name
- Domain/Administrator: Domain qualified AD Logon Name

For *Simple*, provide an LDAP ID. For example:

- cn=DirXML,cn=Users,DC=domain,dc=com

Authentication Password: Provide the password for the specified Active Directory account.

Authentication Context: Specify the name of the Active Directory domain controller to use for synchronization.

For example, for the *Negotiate* authentication method, use the DNS name (for example, `mycontroller.domain.com`). For the *Simple* authentication method, you can use the IP address of your server (for example, `10.10.128.23` or the DNS name).

If no value is specified, `localhost` is used.

5 (Conditional) If you chose to run the driver remotely, click *Next*, then fill in the fields listed below. Otherwise, skip to Step 6.

Remote Host Name and Port: Specify the host name or IP address of the server where the driver's Remote Loader service is running.

Driver Password: Specify the driver object password that is defined in the Remote Loader service. The Remote Loader requires this password to authenticate to the Metadirectory server.

Remote Password: Specify the Remote Loader's password (as defined on the Remote Loader service). The Metadirectory engine (or Remote Loader shim) requires this password to authenticate to the Remote Loader

6 Click *Next* to import the driver configuration.

At this point, the driver is created from the basic configuration file. To ensure that the driver works the way you want it to for your environment, you must review and modify (if necessary) the driver's default configuration settings.

7 To review or modify the default configuration settings, click *Configure*, then continue with the next section, Configuring the Driver.

or

To skip the configuration settings at this time, click *Close*. When you are ready to configure the settings, continue with Configuring the Driver.

4.2.2 Configuring the Driver

There are many settings that can help you customize and optimize the driver. The settings are divided into categories such as Driver Configuration, Engine Control Values, and Global Configuration Values (GCVs). Although it is important for you to understand all of the settings, your first priority should be to review the Driver Parameters located on the Driver Configuration page and the Global Configuration Values. These settings must be configured properly for the driver to start and function correctly.

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

1 Open your project.

2 In the Modeler, right-click the driver icon or the driver line, then select *Properties*.

In addition to the driver settings, you should review the set of default policies and rules provided by the basic driver configuration. Although these policies and rules are suitable for synchronizing with Active Directory, your synchronization requirements for the driver might differ from the default policies. If this is the case, you need to change them to carry out the policies you want. The default policies and rules are discussed in Section 1.4, "Default Driver Configuration," on page 14.

4.2.3 Deploying the Driver

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

1 In Designer, open your project.

2 In the Modeler, right-click the driver icon or the driver line, then select *Live* > *Deploy*.

3 If you are authenticated to the Identity Vault, skip to Step 5; otherwise, specify the following information:

- **Host:** Specify the IP address or DNS name of the server hosting the Identity Vault.
- **Username:** Specify the DN of the user object used to authenticate to the Identity Vault.
- **Password:** Specify the user's password.

4 Click *OK*.

5 Read through the deployment summary, then click *Deploy*.

6 Read the successful message, then click *OK*.

7 Click *Define Security Equivalence* to assign rights to the driver.

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

7a Click *Add*, then browse to and select the object with the correct rights.

7b Click *OK* twice.

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

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

8a Click *Add*, then browse to and select the user object you want to exclude.

8b Click *OK*.

8c Repeat Step 8a and Step 8b for each object you want to exclude.

8d Click *OK*.

9 Click *OK*.

4.2.4 Starting the Driver

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

To start the driver:

1 In Designer, open your project.

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

For information about management tasks for the driver, see Chapter 7, "Managing Active Directory Groups and Exchange Mailboxes," on page 57.

4.3 Creating the Driver in iManager

You create the Active Directory driver by importing the driver's basic configuration file and then modifying the configuration to suit your environment. After you create and configure the driver, you need to start it.

4.3.1 Importing the Driver Configuration File

1 In iManager, click  to display the Identity Manager Administration page.

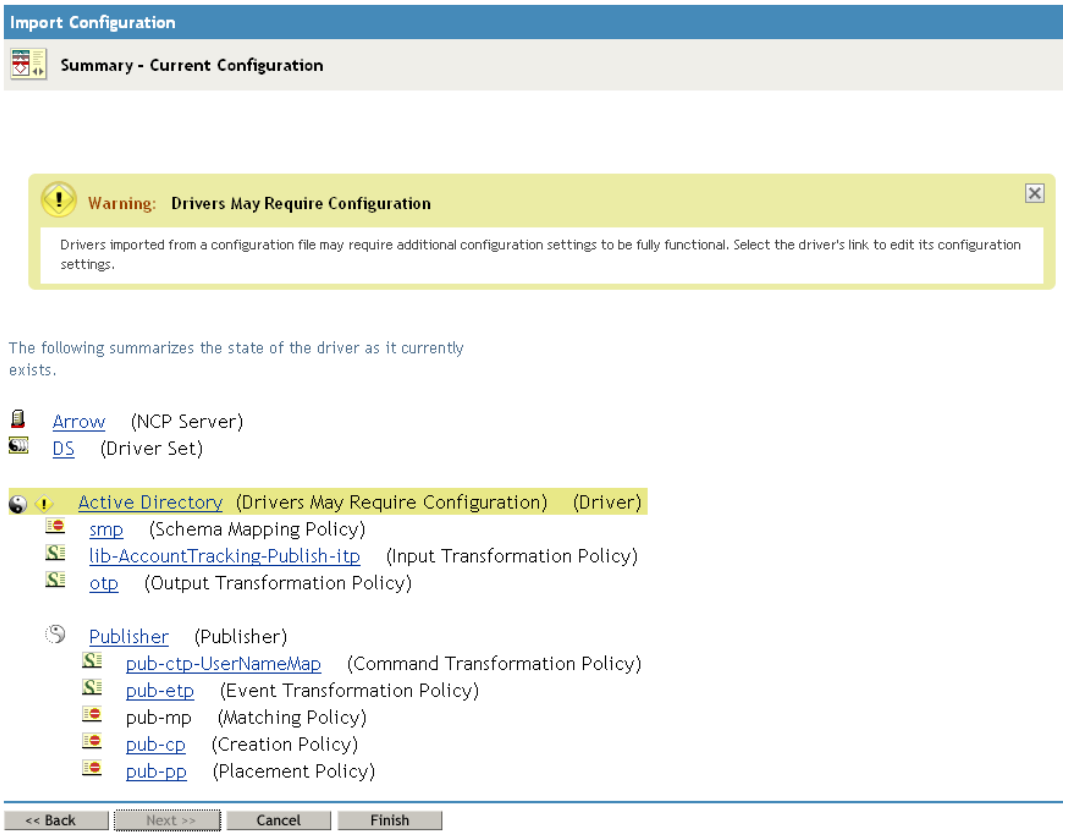
2 In the Administration list, click *Import Configuration* to launch the Import Configuration Wizard.

3 Follow the wizard prompts, filling in the requested information (described below) until you reach the Summary page.

Prompt	Description
Where do you want to place the new driver?	You can add the driver to an existing driver set, or you can create a new driver set and add the driver to the new set. If you choose to create a new driver set, you are prompted to specify the name, context, and server for the driver set.
Import a configuration into this driver set	Use the default option, *Import a configuration from the server (.XML file)*. In the *Show* field, select *Identity Manager 3.6 configurations*. In the *Configurations* field, select the ActiveDirectory file.
Driver name	Type a name for the driver. The name must be unique within the driver set.
Connected System or Driver Name	Specify the name of the connected system, application or Identity Manager driver.
Domain DNS Name	Specify the DNS name of the Active Directory domain managed by this driver.
Active Directory User Container	Specify the container where user objects reside in Active Directory.
User Container	Select the Identity Vault container where Active Directory users will be placed. This value becomes the default for all drivers in the driver set. If you don't want to change this value for all drivers, leave this field unchanged and change the value on the driver's Global Configuration Values page after you've finished importing the driver.
Driver is Local/Remote	Select *Local* if this driver will run on the Metadirectory server without using the Remote Loader service. Select *Remote* if you want the driver to use the Remote Loader service, either locally on the Metadirectory server or remotely on another server.
Remote Host Name and Port	This applies only if the driver is running remotely. Specify the host name or IP address of the server where the driver's Remote Loader service is running.
Driver Password	This applies only if the driver is running remotely. Specify the driver object password that is defined in the Remote Loader service. The Remote Loader requires this password to authenticate to the Metadirectory server.
Remote Password	This applies only if the driver is running remotely. Specify the Remote Loader's password (as defined on the Remote Loader service). The Metadirectory engine (or Remote Loader shim) requires this password to authenticate to the Remote Loader

Prompt	Description
Authentication ID	Specify an Active Directory account with administrative privileges to be used by Identity Manager. The form of the name used depends on the selected authentication mechanism. For *Negotiate*, provide the name form required by your Active Directory authentication mechanism. For example: ◆ Administrator: AD Logon Name ◆ Domain/Administrator: Domain qualified AD Logon Name For *Simple*, provide an LDAP ID. For example: ◆ cn=DirXML,cn=Users,DC=domain,dc=com
Authentication Password	Provide the password for the Active Directory account specified.
Authentication Context	Provide the name of the Active Directory domain controller to use for synchronization. For example, for the *Negotiate* authentication method, use the DNS name (for example, `mycontroller.domain.com`). For the *Simple* authentication method, you can use the IP address of your server (for example, `10.10.128.23` or the DNS name). If no value is specified, `localhost` is used.
Define Security Equivalences	The driver requires rights to objects within the Identity Vault. The Admin user object is most often used to supply these rights. However, you might want to create a DriversUser (for example) and assign security equivalence to that user. Whatever rights that the driver needs to have on the server, the DriversUser object must have the same security rights.
Exclude Administrative Roles	You should exclude any administrative User objects (for example, Admin and DriversUser) from synchronization.

When you finish providing the information required by the wizard, a Summary page similar to the following is displayed.



At this point, the driver is created from the basic configuration file. To ensure that the driver works the way you want it to for your environment, you must review and modify (if necessary) the driver's default configuration settings.

4 To modify the default configuration settings, click the linked driver name, then continue with the next section, Configuring the Driver.

or

To skip the configuration settings at this time, click *Finish*. When you are ready to configure the settings, continue with Configuring the Driver.

4.3.2 Configuring the Driver

There are many settings that can help you customize and optimize the driver. The settings are divided into categories such as Driver Configuration, Engine Control Values, and Global Configuration Values (GCVs). Although it is important for you to understand all of the settings, your first priority should be to review the Driver Parameters located on the Driver Configuration page and the Global Configuration Values. These settings must be configured properly for the driver to start and function correctly.

To configure the settings:

1 Make sure the Modify Object page for the Active Directory driver is displayed in iManager. If it is not:

1a In iManager, click to display the Identity Manager Administration page.

1b Click *Identity Manager Overview*.

1c Browse to and select the driver set object that contains the new driver.

1d Click the driver set name to access the Driver Set Overview page.

1e Click the upper right corner of the driver, then click *Edit properties*.

2 Review the settings on the various pages and modify them as needed for your environment. The configuration settings are explained in Appendix A, "Driver Properties," on page 75.

3 After modifying the settings, click *OK* to save the settings and close the Modify Object page.

4 (Conditional) If the Active Directory driver's Summary page for the Import Configuration wizard is still displayed, click *Finish*.

WARNING: Do not click *Cancel* on the Summary page. This removes the driver from the Identity Vault and results in the loss of your work.

In addition to the driver settings, you should review the set of default policies and rules provided by the basic driver configuration. Although these policies and rules are suitable for synchronizing with Active Directory, your synchronization requirements for the driver might differ from the default policies. If this is the case, you need to change them to carry out the policies you want. The default policies and rules are discussed in Section 1.4, "Default Driver Configuration," on page 14.

4.3.3 Starting the Driver

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

To start the driver:

1 In iManager, click  to display the Identity Manager Administration page.

2 Click *Identity Manager Overview*.

3 Browse to and select the driver set object that contains the driver you want to start.

4 Click the driver set name to access the Driver Set Overview page.

5 Click the upper right corner of the driver, then click *Start driver*.

For information about management tasks with the driver, see Chapter 8, "Managing the Driver," on page 61.

4.4 Activating the Driver

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

For information on activation, refer to "Activating Novell Identity Manager Products" in the *Identity Manager 3.6 Installation Guide*.

Upgrading an Existing Driver

5

If you are running the driver on the Metadirectory server, the driver shim files are updated when you update the server unless they were not selected during a custom installation. If you are running the driver on another server, the driver shim files are updated when you update the Remote Loader on the server.

The 3.6 version of the driver shim supports drivers created by using any 3.*x* version of the driver configuration file. You can continue to use these driver configurations until you want to upgrade them.

The following sections provide information to help you upgrade an existing driver to version 3.6:

5.1 Supported Upgrade Paths

You can upgrade from any 3.*x* version of the Active Directory driver. Upgrading a pre-3.*x* version of the driver directly to version 3.6 is not supported.

5.2 What’s New in Version 3.6

Version 3.6 of the driver does not include any new features that are not already in the 3.5.1 version. However, the process for creating a new driver has changed. For detailed information, see Chapter 4, “Creating a New Driver,” on page 33.

5.3 Upgrade Procedure

The process for upgrading the Active Directory driver is the same as for other Identity Manager drivers. For detailed instructions, see “Upgrading” in the *Identity Manager 3.6 Installation Guide*.

Synchronizing Passwords

6

To set up password synchronization among Active Directory, the Identity Vault, and connected systems, you need to complete the tasks in the "Password Management Checklist" in the *Identity Manager 3.6 Password Management Guide*. The information in the following sections supplements the information in that guide.

For information on troubleshooting password synchronization, see "Tips on Password Synchronization" on page 69.

6.1 Setting Up SSL

For the driver to set a password in Active Directory (Subscriber channel), it must have a secure connection provided by one of the following conditions:

- The machine running the driver is the same machine as the domain controller.
- The machine running the driver is in the same domain as the domain controller.
- The machine not in the domain requires the Simple method and SSL set up between it and the domain controller. Bidirectional password synchronization is available only when using the Negotiate authentication mechanism.

  Refer to Microsoft documentation for instructions, such as Enabling Secure Sockets Layer for SharePoint Portal Server 2003 (http://office.microsoft.com/en-us/assistance/HA011648191033.aspx).

In addition, the driver must have SSL enabled or have signing and sealing enabled. Enabling SSL or signing and sealing is done in the driver parameters. For more information, see Section A.1.5, "Driver Parameters," on page 78.

6.2 Setting Up Password Synchronization Filters

The Active Directory driver must be configured to run on only one Windows machine. However, for password synchronization to occur, you must install a password filter (`pwFilter.dll`) on each domain controller and configure the registry to capture passwords to send to the Identity Vault.

The password filter is automatically started when the domain controller is started. The filter captures password changes that users make by using Windows clients, encrypts the changes, and sends them to the driver to update the Identity Vault.

To simplify installation and administration of password filters, an Identity Manager PassSync utility is added to the Control Panel when the driver is installed. This utility gives you two choices for setting up the password filters, depending on whether you want to allow remote access to the registry on your domain controllers:

- Section 6.2.1, “Allowing Remote Access to the Registry,” on page 46
- Section 6.2.2, “Not Allowing Remote Access to the Registry,” on page 50

6.2.1 Allowing Remote Access to the Registry

If you allow remote access to the registry of each domain controller from the machine where you are running the driver, use the procedure in this section to configure the password filter. It allows the Identity Manager PassSync utility to configure each domain controller from one machine.

If you configure all the domain controllers from one machine, the Identity Manager PassSync utility provides the following features to help you during setup:

- Lets you specify which domain you want to participate in password synchronization.
- Automatically discovers all the domain controllers for the domain.
- Lets you remotely install the `pwFilter.dll` on each domain controller.
- Automatically updates the registry on the machine where the driver is running and on each domain controller.
- Lets you view the status of the filter on each domain controller.
- Lets you reboot a domain controller remotely.

  Rebooting the domain controller is necessary when you first add a domain for password synchronization, because the filter that captures password changes is a DLL file that starts when the domain controller is started.

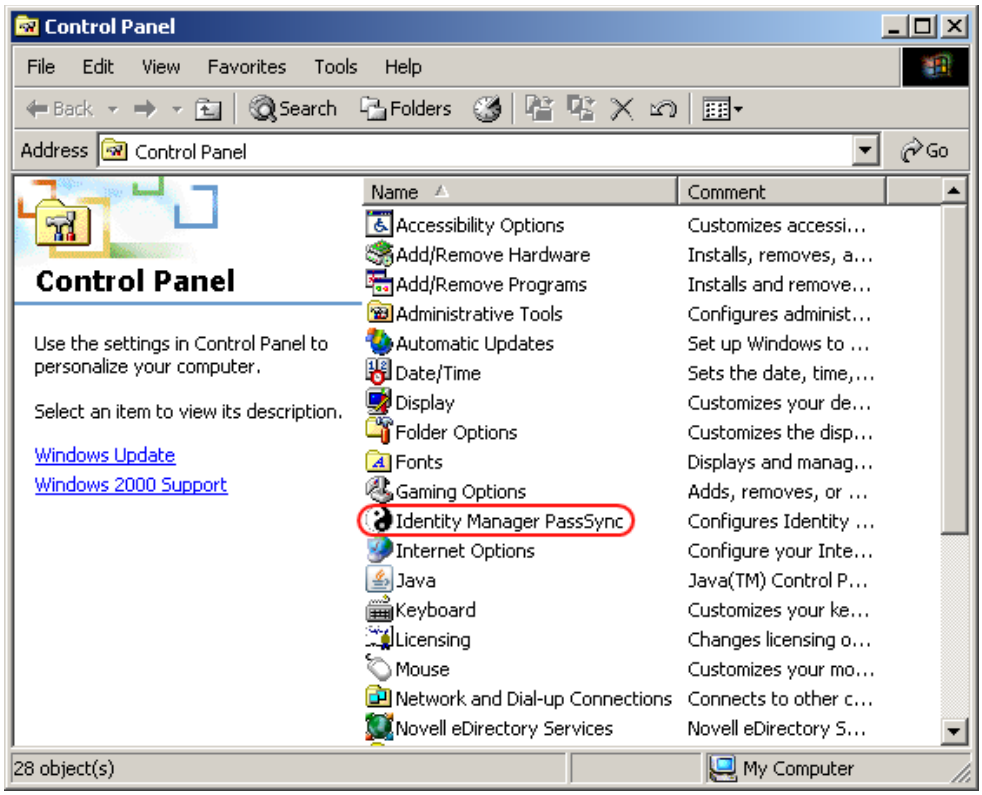
Because setting up the filter requires rebooting the domain controller, you might want to perform this procedure after hours, or reboot only one domain controller at a time. If the domain has more than one domain controller, keep in mind that each domain controller where you want Password Synchronization to function must have the filter installed and must be rebooted.

1. Confirm that port 135 (the RPC endpoint mapper) is accessible on the domain controllers and on the machine where the Active Directory driver is configured to run.

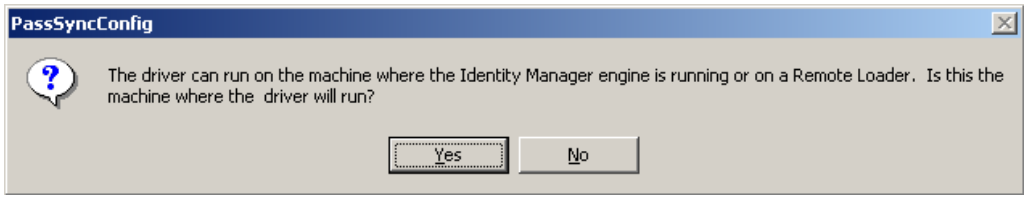
   If you are using NetBIOS over TCP, you also need these ports:

   - 137: NetBIOS name service
   - 138: NetBIOS datagram service
   - 139: NetBIOS session service

   A firewall could prevent the ports from being accessible remotely.
2. Log in with an administrator account on the computer where the driver is installed.
3. At the computer where the driver is installed, click *Start* > *Control Panel* > *Identity Manager PassSync*.



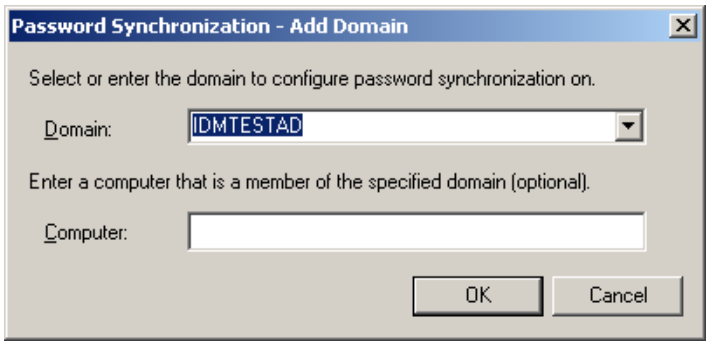
4 In the dialog box that is displayed, click *Yes* to specify that this is the machine where the driver is installed.



You only receive this prompt the first time you run the utility. After you complete the configuration, you are not shown this prompt again unless you remove this domain from the list.

5 Click *Add*, then browse to and select the domain that you want to participate in password synchronization.

The drop-down list displays known domains.



The Identity Manager PassSync utility discovers all the domain controllers for that domain, and installs `pwFilter.dll` on each domain controller. It also updates the registry on the computer where you are running the drivers, and on each domain controller. This might take a few minutes.

The `pwFilter.dll` doesn't capture password changes until the domain controller has been rebooted. The Identity Manager PassSync utility lets you see a list of all the domain controllers and the status of the filter on them. It also lets you reboot the domain controller from inside the utility.

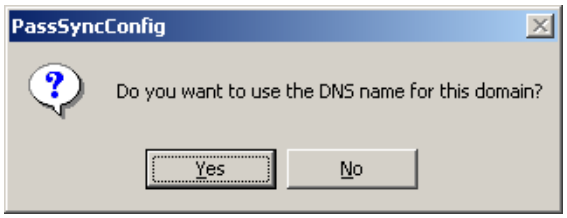
6 (Optional) Specify a computer in the domain, then click *OK*.

If you leave the *Computer* field blank, PassSync queries the local machine. Therefore, if you are running PassSync on a domain controller, you don't need to specify a name. PassSync queries the local machine (in this case, a domain controller) and gets (from the database) the list of all domain controllers in the domain.

If you aren't installing on a domain controller, specify the name of a computer that is in the domain and that can get to a domain controller.

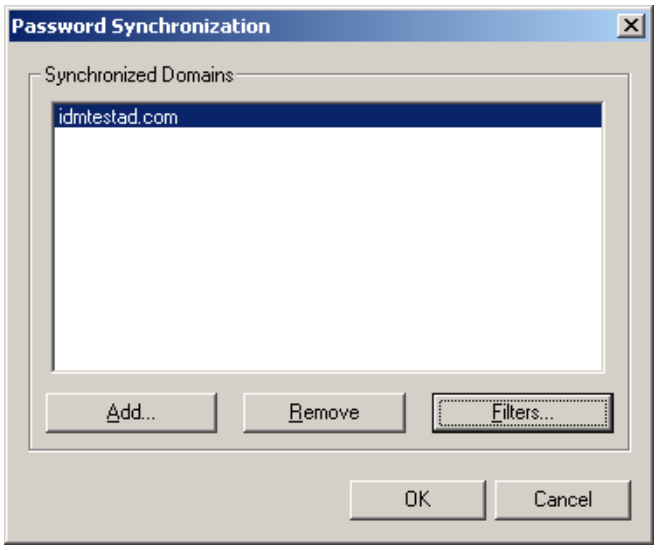
If you receive an error message indicating that PassSync can't locate a domain, specify a name.

7 Click *Yes* to use the domain's DNS name.

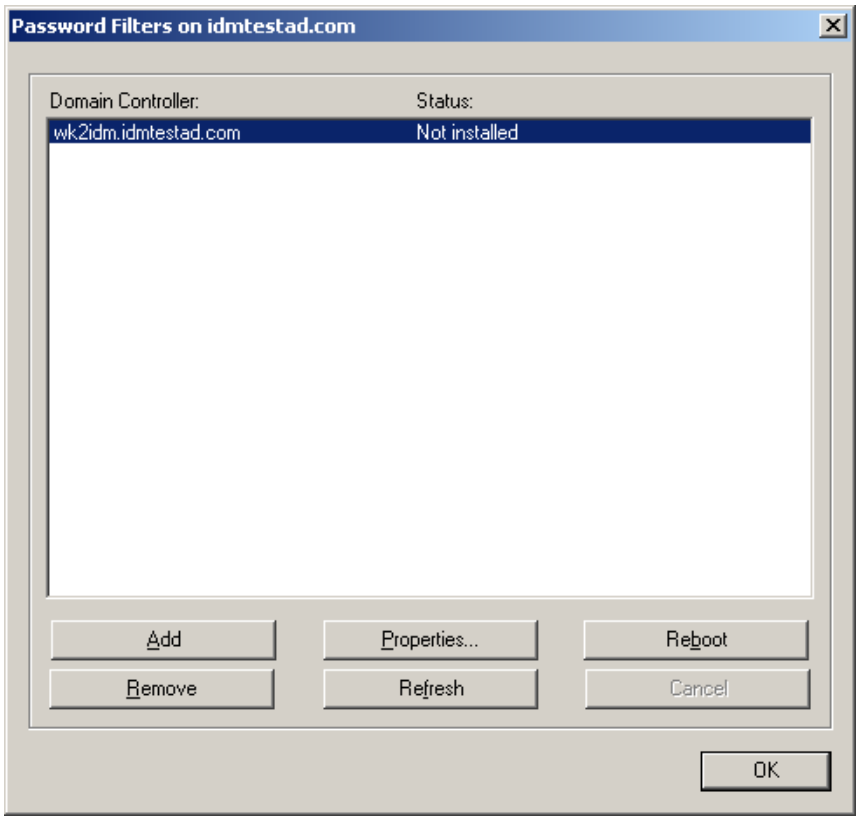


You can select *No*, but the DNS name provides more advanced authentication and the ability to more reliably discover domains in bigger installations. However, the choice depends on your environment.

8 Select the name of the domain you want to participate in password synchronization from the list, then click *Filters*.



The utility displays the names of all the domain controllers in the selected domain and the status of the filter.



The status for each domain controller should display the filter state as *Not installed*. However, it might take a few minutes for the utility to complete its automated task, and in the meantime the status might say *Unknown*.

9 To install the filter, click *Reboot*.

You can choose to reboot the domain controllers at a time that makes sense for your environment. Just keep in mind that password synchronization won't be fully functional until every domain controller has been rebooted.

10 When the status for all domain controllers is *Running*, test password synchronization to confirm that it is working.

11 To add more domains, click *OK* to return to the list of domains, and repeat Step 5 through Step 10.

6.2.2 Not Allowing Remote Access to the Registry

If you do not want to allow remote access to the registry of each domain controller, you must set up the password filters on each domain controller separately. To do this, go to each domain controller, install the driver files so you have the Identity Manager PassSync utility, and use the utility on each machine to install the password filter and update the registry.

In this procedure, you install the driver so that you have the Identity Manager PassSync utility. Then you use the utility to install the `pwFilter.dll` file, specify the port to use, and specify which host machine is running the Identity Manager Driver for Active Directory.

Because setting up the filter requires rebooting the domain controller, you might want to perform this procedure after hours, or reboot only one domain controller at a time. If a domain has more than one domain controller, keep in mind that each domain controller where you want Password Synchronization to function must have the filter installed and must be rebooted.

This procedure is for any domain controller that does not have the Active Directory driver installed on it.

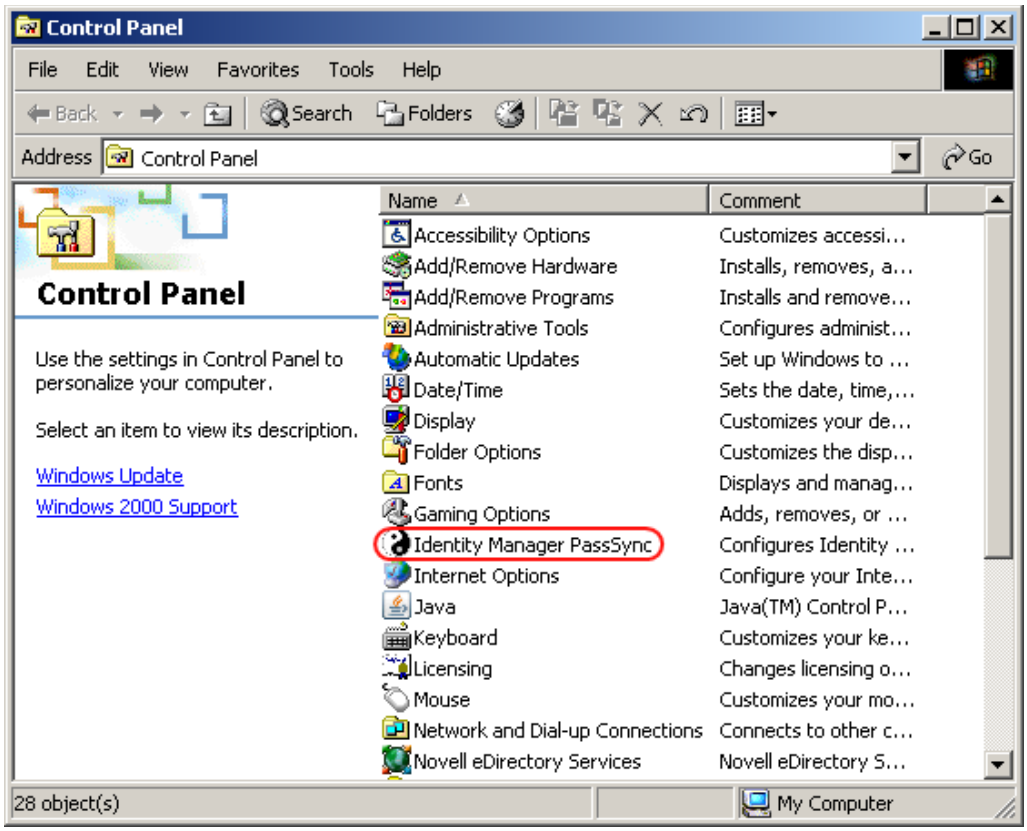
1 Confirm that the following ports are available on both the domain controller and the machine where the Identity Manager Driver for Active Directory is configured to run:

- 135: The RPC endpoint mapper
- 137: NetBIOS name service
- 138: NetBIOS datagram service
- 139: NetBIOS session service

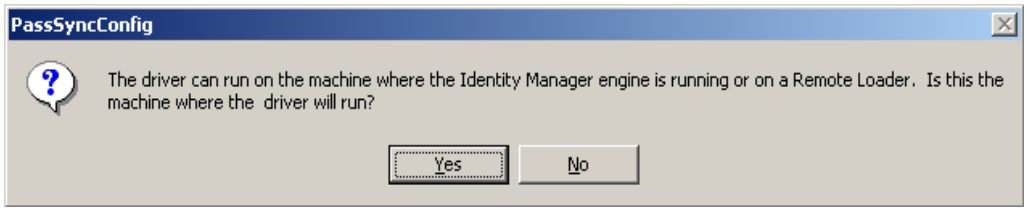
2 On the domain controller, use the Identity Manager Installation to install only the Identity Manager Driver for Active Directory.

Installing the driver installs the Identity Manager PassSync utility.

3 Click *Start* > *Settings* > *Control Panel* > *Identity Manager PassSync*.

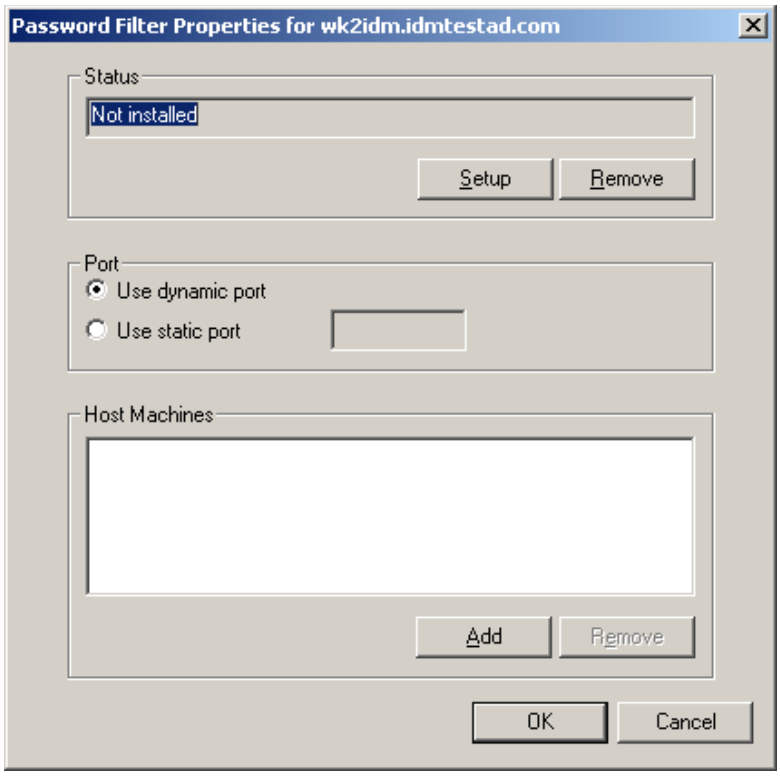


4 In the dialog box that displays, click *No* to specify that this machine is not running the Active Directory driver.



After you complete the configuration, you are not shown this prompt again unless you remove the password filter by using the *Remove* button in the Password Filter Properties dialog box.

5 After you click *No*, the Password Filter Properties dialog box appears, with a status message indicating that the password filter is not installed on this domain controller.

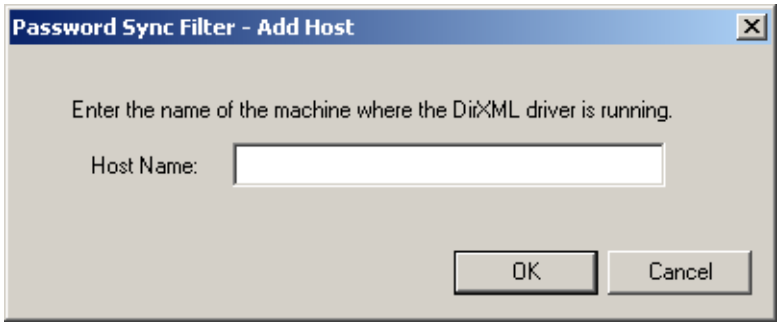


6 Click the *Setup* button to install the password filter, `pwFilter.dll`.

7 For the *Port* setting, specify whether to use dynamic port or static port.

Use the static port option only if you have decided to configure your remote procedure call (RPC) for the domain controller differently than the default.

8 Click *Add* to specify the hostname of the machine running the Identity Manager driver, then click *OK*.



This step is necessary so that the password filter knows where to send the password changes. The password filter captures password changes, and must send them to the Identity Manager driver to update the Identity Manager data store.

9 Verify that the information specified in Step 6 through Step 8 is correct, then click *OK*.

10 Reboot the domain controller to complete the installation of the password filter.

You can choose to reboot at a time that makes sense for your environment. Just keep in mind that password synchronization won't be fully functional until every domain controller has the password filter installed and has been rebooted.

After the installation is complete and the domain controller is rebooted, the password filter is loaded automatically whenever the domain controller starts up.

11 Check the status for the password filter again by clicking *Start* > *Settings* > *Control Panel*, and double-clicking the Identity Manager PassSync utility.

Confirm that the status says Running.

12 Repeat Step 2 through Step 11 for each domain controller that you want to participate in Password Synchronization.

13 When the status says Running for all the domain controllers, test password synchronization to confirm that it is working by having a user change his or her password using the Windows Client. This should initiate the synchronization process.

6.3 Retrying Synchronization after a Failure

The following sections explain the retry methods used after a synchronization failure:

- Section 6.3.1, "Retrying after an Add or Modify Event," on page 53
- Section 6.3.2, "Password Expiration Time," on page 53

6.3.1 Retrying after an Add or Modify Event

If a password change sent from Active Directory is not successfully completed in the Identity Vault, the driver caches the password. It is not retried again until an Add or Modify event occurs for the user that the password belongs to. (Previously, these saved passwords were retried at every polling interval.)

When the driver polls for changes in Active Directory, the driver receives Add or Modify events for users. For each user Add or Modify event, the driver checks to see if it has a password saved for this new user. If it does, the driver sends the password to the Identity Vault as a Modify user event.

If you have set up Password Synchronization to send e-mail messages to users when password synchronization fails, this enhancement minimizes the number of e-mails that a user might receive.

6.3.2 Password Expiration Time

The Password Expiration Time parameter lets you determine how long to save a particular user's password if synchronization is not successful on the first try. The driver saves a password until it is successfully changed in the Identity Vault, or until the Password Expiration Time elapses.

You are prompted to specify an expiration time when you import the sample driver configuration. If you don't specify a time, or if the interval field contains invalid characters, the default setting is 60 minutes. If the time specified is less than three times the polling interval specified, the driver changes the time to be three times the polling interval.

Set the value large enough to handle whatever temporary backlog of passwords exists. If you are doing bulk changes, set the timeout large enough to handle all the changes. The rule of thumb is to allow one second per password. For example, to synchronize 18,000 passwords, allow 300 minutes (18,000 passwords divided by 60 seconds).

A setting of -1 is indefinite. Although this setting can handle bulk changes, it can cause problems. For example, a password might never be synchronized because the account wasn't associated. Such a password would therefore remain in the system forever. A number of similar situations could result in a large inventory of unsynchronized passwords held by the system.

Scenarios Relating to Password Expiration Time

On the Publisher channel, password synchronization might occur before the Add event. The driver retries immediately following the Add event.

- "Scenario: No Effect" on page 54
- "Scenario: Increasing the Expiration Time" on page 54
- "Scenario: Never Meeting Requirements" on page 55
- "Scenario: E-Mail Notifications" on page 55

Scenario: No Effect

A new user with a password is created in Active Directory. The filter immediately sends the new password to the driver. However, the driver hasn't yet received that user Add event because the event occurred between polling intervals. Because the driver has not yet created the user in the Identity Vault, the password synchronization is not successful on this first attempt. The driver caches the password.

At the next polling interval, the driver receives the Add user event for the new user. The driver also checks to see if it has a password cached for this new user. The driver sends the Add user event to the Identity Vault, and also sends a Modify user event to synchronize the password.

In this case, the password synchronization is delayed by only one polling interval.

The Password Expiration Time parameter doesn't have an effect in this situation.

Scenario: Increasing the Expiration Time

A new user with a password is created in Active Directory. However, the user information doesn't meet the requirements of the Create policy for the Active Directory driver.

For example, perhaps the Create rule requires a full name, and the required information is missing. Like the No Effect example, the filter immediately sends the password change to the driver. However, on the first try the password change is not successful in the Identity Vault because the user doesn't exist yet. The driver caches the password.

In this case, however, even when the driver polls for changes in Active Directory and discovers the new user, the driver can't create the new user because the user information doesn't meet the Create policy's requirements.

Creating the new user and synchronizing the password are delayed until all the user information is added in Active Directory to satisfy the Create policy. Then the driver adds the new user in the Identity Vault, checks to see if it has a password cached for this new user, and sends a Modify user event to synchronize the password.

The Password Expiration Time parameter affects this scenario only if the time interval elapses before the user information in Active Directory meets the requirements of the Create policy. If the Add event comes in after the password has expired and the driver doesn't have the password cached for that user, synchronization can't occur. Because the driver doesn't have a cached password, the driver uses the default password in the password policy.

After the user changes the password in either Active Directory or the Identity Vault, that password is synchronized.

If Password Synchronization is set up for bidirectional flow of passwords, a password can also be synchronized from the Identity Vault to Active Directory when a password change is made in the Identity Vault.

If your Create policy is restrictive, and it generally takes longer than a day for a new user's information to be completed in Active Directory, you might want to increase the Password Expiration Time parameter interval accordingly. The driver can then cache the passwords until the user is finally created in the Identity Vault.

Scenario: Never Meeting Requirements

A user with a password is created in Active Directory. However, this user never meets the criteria of the Create policy for the Active Directory driver.

For example, perhaps the new user in Active Directory has a Description that indicates the user is a contractor, and the Create policy blocks creation of User objects for contractors because the business policy is that contract employees are not intended to have a corresponding user account in the Identity Vault. Like the previous example, the filter immediately sends the password change, but the password synchronization isn't successful on the first attempt. The driver caches the password.

In this case, a corresponding user account is never created in the Identity Vault. Therefore, the driver never synchronizes the cached password. After the Password Expiration Time has passed, the driver removes the user password from its cache.

Scenario: E-Mail Notifications

Markus has an Active Directory account and a corresponding Identity Vault account. He changes his Active Directory password, which contains six characters. However, the password doesn't meet the eight-character minimum required by the Password Policy that the administrator created in eDirectory. Password Synchronization is configured to reject passwords that do not meet the policy and to send a notification e-mail to Markus saying that password synchronization failed. The driver caches the password and retries it only if a change is made to the User object in Active Directory.

In this case, shortly after changing a password, Markus receives an e-mail stating that the password synchronization wasn't successful. Markus receives the same e-mail message each time the driver retries the password.

If Markus changes the password in Active Directory to one that complies with the Password Policy, the driver synchronizes the new password to the Identity Vault successfully.

If Markus doesn't change to a compliant password, the password synchronization is never successful. When the Password Expiration Time elapses, the driver deletes the cached password and no longer retries it.

6.4 Disabling Password Synchronization on a Driver

You can disable password synchronization on a driver by setting the *Password Sync Timeout* parameter to 0. Sometimes there is a need to have two Active Directory drivers enabled for one domain, but you only want one driver handling the password synchronization. Make sure that the *Password Sync Timeout* parameter is set to 0 on the driver that does not synchronize passwords.

A use case for this is if one driver is synchronizing User objects and another driver is synchronizing Contacts. Contacts are displayed in the Exchange Global Address List (GAL), but they do not require an Active Directory license because they do not authenticate.

See "Password Sync Timeout (minutes)" on page 80 in the Appendix A, "Driver Properties," on page 75 for more information about this parameter.

Managing Active Directory Groups and Exchange Mailboxes

7

The following sections provide information to help you use the Active Directory driver to manage groups and Exchange mailboxes that reside in Active Directory:

7.1 Managing Groups

The Active Directory group class defines two types of groups and three scopes for membership in the group. Type and scope are controlled by the groupType attribute, which can be set via an Identity Manager policy when a group is created in Active Directory and changed by modifying the attribute.

A group holds a collection of object references. The Distribution Group type gives no special rights or privileges to its members and is commonly used as a distribution list for Exchange. The Security Group type is a security principal. Its members receive the rights and privileges of the group. Security Groups have a pre-Windows 2000 logon name (samAccountName) and a Security Identifier (SID) that can be used in Security Descriptor (SD) Access Control Lists (ACL) on other objects to grant or deny rights and privileges to its members.

Group scope controls whether an object from a foreign domain can be a member of the group and also whether the group itself can be a member of another group. The three scopes are Domain Local, Global, and Universal. How these scopes behave, or whether the scope is valid at all, depends on whether Active Directory is operating in Windows 2000 Mixed, Windows 2000 Native, or Windows 2003 mode.

In general, Domain Local groups can hold references to objects anywhere in the forest but can be assigned permissions only within the domain. Global groups are the opposite. They can only hold references to objects within the domain but can be assigned permissions throughout the forest. Universal groups can hold references and can be assigned permissions throughout the forest. However, Universal groups come with their own restrictions and performance issues. Groups should be created and used in conformance with Microsoft recommendations.

The groupType attribute is a 32-bit integer whose bits define type and scope. Groups can have only a single scope at any given time.

Table 7-1 *GroupType Attribute*

GroupType Attribute	Scope	Bits That Define Type and Scope
GROUP_TYPE_GLOBAL_GROUP	Distribution	0x00000002
GROUP_TYPE_DOMAIN_LOCAL_GROUP	Distribution	0x00000004
GROUP_TYPE_UNIVERSAL_GROUP	Distribution	0x00000008

GroupType Attribute	Scope	Bits That Define Type and Scope
GROUP_TYPE_SECURITY_ENABLED	Security	0x80000000

7.2 Managing Microsoft Exchange Mailboxes

The Active Directory driver can be configured to create, move, and delete Microsoft Exchange mailboxes for users in Active Directory. Mailboxes are managed by setting and removing the value for the homeMDB attribute on the user object. This attribute holds the Distinguished Name of the Exchange Private Message Database (MDB) where the mailbox resides. The driver manages mailboxes on Exchange servers that are in the same domain as the driver only.

There are several different ways to manage Exchange mailboxes. The default configuration manages mailboxes through policy decisions made in the Subscriber Command Transformation policy. When a user meets the given conditions, a mailbox is created, moved, or removed. The import file gives you three choices for mailbox management:

- Entitlements
- Policies
- Do not Manage Exchange Mailboxes

When you use the entitlement method for provisioning, a user is granted or denied a mailbox based on the entitlement set on the user in the Identity Vault. The entitlement holds the Distinguished Name of the MDB and a state value that tells the driver whether the entitlement is granted or revoked. The entitlement itself is managed by the User Application or the Role-Based Entitlements driver. In either case, the external tool grants (or revokes) the right to the mailbox, the Subscriber Command Transformation policy translates that right into an add-value or remove-value on the homeMDB attribute and the driver shim translates the change to homeMDB into the proper calls to the Exchange management system.

If you are using entitlements and have multiple MDBs in your organization, you use the User Application to decide which MDB is to be assigned to a given user. The role of the Identity Manager driver is to respond to the entitlements placed on the user object, not to put them there. If you are using the User Application, you are given a list of Exchange MDBs to choose from as the workflow item flows through the approval process. If you are using Role-Based entitlements, the MDB is assigned to the group that holds the role for the user.

When you use the policy-based method for provisioning, the Subscriber Command Transformation policy uses information about the state of the user object in the Identity Vault to assign the MDB. The driver shim translates the change into the proper calls to the Exchange management system. The default policy uses a simple rule for assigning the mailbox. It assumes that there is only one MDB and that all users that have come this far through the policy chain should be assigned to that MDB. Because the rules for assigning different MDBs vary widely from company to company, the default configuration does not attempt to establish a “right way” of doing it. You implement your own policies simply by changing the default assignment rules. You use DirXML® Script if statements to define the conditions for mailbox assignments and the do-set-dest-attribute command for the homeMDB attribute to effect the change. You can get a list of Exchange MDBs by using the `ADManager.exe` tool or by your own means.

When it is not managing Exchange mailboxes, the driver will synchronize the user’s e-mail address and mail nickname.

There are other ways to manage the Exchange mailbox. For instance, you could extend the schema of the Identity Vault to hold the homeMDB information and use basic data sychronization to assign the mailbox to the user in Active Directory. In this case, you would use your own tool to make assignments in the Identity Vault.

The default policy works well for simple mailbox assignment to a single MDB. If you want the policy to reflect more complex rules demanded in your environment, the policy must be changed.

Managing the Driver

8

As you work with the Active Directory driver, there are a variety of management tasks you might need to perform, including the following:

- Starting, stopping, and restarting the driver
- Viewing driver version information
- Using Named Passwords to securely store passwords associated with the driver
- Monitoring the driver's health status
- Backing up the driver
- Inspecting the driver's cache files
- Viewing the driver's statistics
- Using the DirXML® Command Line utility to perform management tasks through scripts
- Securing the driver and its information
- Synchronizing objects
- Migrating and resynchronizing data
- Activating the driver
- Upgrading an existing driver

Because these tasks, as well as several others, are common to all Identity Manager drivers, they are included in one reference, the *Identity Manager 3.6 Common Driver Administration Guide*.

Security Best Practices

9

This following sections contains a description of the security parameters unique to the Active Directory driver.

- Section 9.1, “Default Configuration of the Security Parameters,” on page 63
- Section 9.2, “Recommended Security Configurations when Using the Remote Loader,” on page 65
- Section 9.3, “Recommended Security Configurations when Using the Simple Authentication Method,” on page 66

For additional information about securing your Identity Manager system, see the *Identity Manager 3.6 Security Guide*.

9.1 Default Configuration of the Security Parameters

The security parameters must be configured correctly for the driver to function properly. In most instances, the driver does not start if the parameters are not configured correctly.

To change these parameters in iManager:

1. Click *Identity Manager* > *Identity Manager Overview*, then click *Search* to search for the driver set that is associated with the driver.
2. Browse to the driver, then click the upper right corner of the driver icon.
3. Click *Edit Properties* > *Driver Configuration* > *Driver Parameters*.
4. Review the driver parameters in Table 9-1, and decide if you need to make any changes.

To change these parameters in Designer:

1. Open a project in the modeler, then right-click the driver line and select *Properties* > *Driver Configuration*.
2. Click *Driver Parameters*.
3. Review the driver parameters in Table 9-1, and decide if you need to make any changes.

Table 9-1 *Security Parameters*

Security Parameter	Description
Authentication ID	The account the driver uses to access the domain data. The *Authentication ID* can be specified by using different formats: ◆ If the *Authentication method* is set to *negotiate*, the user name is specified with the domain name or the full qualified domain name. For example, `user` or `domain\user`. ◆ If the *Authentication method* is set to *simple*, the user name must be specified using an LDAP fully distinguished name. For example, `cn=IDMadmin,cn=Users,dc=domain,dc=com`.

Security Parameter	Description
Authentication context	The context used to access domain data. The *Authentication context* can be specified by using different formats: ◆ If the *Authentication method* is set to *negotiate*, use the DNS name of the Active Directory domain controller. For example, `mycontroller.mydomain.com`. ◆ If the *Authentication method* is set to *simple*, use the DNS name of the Active Directory domain controller or the IP address of the LDAP server. For example, `mycontroller.mydomain.com` or `10.0.0.1`.
Application password	The password for the *Authentication ID* account.
Authentication Method	The method of authentication to Active Directory. *Negotiate* uses Microsoft's security package to negotiate the logon type. Typically Kerberos or NTLM is selected. *Simple* uses LDAP style simple bind for logon. If you want to use Password Synchronization, select *Negotiate*.
Digitally sign communications	This setting requires Windows 2003 or Windows 2000 with the most recent support pack, and Internet Explorer 5.5 SP2 or later on both servers. This enables signing on a Kerberos or NTLM v2 authenticated connection. Select *Yes* to digitally sign the communication between the driver shim and Active Directory. This does not hide the data from view on the network, but it reduces the chance of security attacks. Signing only works when you use the *Negotiate* authentication method and the underlying security provider selects NTLM v2 or Kerberos for its protocol. Do not use this option with SSL. Select *No* to have communications not signed.
Digitally sign and seal communications	This setting requires Windows 2003 or Windows 2000 with the most recent support pack, and Internet Explorer 5.5 SP2 or later on both servers. This setting enables encryption on a Kerberos or NTLM v2 authenticated connection. Select *Yes* to digitally encrypt communication between the driver shim and the Active Directory database. Sealing only works when you use the *Negotiate* authentication method and the underlying security provider selects NTLM v2 or Kerberos for its protocols. Do not use this option with SSL. Select *No* to not have communication between the driver shim and the Active Directory database signed and sealed.

Security Parameter	Description
Use SSL for encryption	Select *Yes* to digitally encrypt communication between the driver shim and the Active Directory database. This option can be used with *Negotiate* or *Simple* authentication methods. SSL requires that the Microsoft server running the driver shim imports the domain controller's server certificate. For more information, see Securing Windows 2000 Server (http://www.microsoft.com/technet/security/prodtech/windows2000/secwin2k/swin2kad.mspx). By default, the parameter is set to *No*. If you set this value to *Yes*, the SSL pipe is encrypted for the entire conversation. An encrypted pipe is preferred because the driver typically synchronizes sensitive information. However, encryption slows the general performance of your servers.
Logon and impersonate	Select *Yes* to log on and impersonate the driver authentication account for CDOEXM (Collaboration Data Object for Exchange Management) and Password Set support. The driver performs a local logon. The authentication account must have the proper rights assignment. For more information, see Section 2.4, "Creating an Administrative Account," on page 26. If *No* is selected, the driver performs a network logon only.

9.2 Recommended Security Configurations when Using the Remote Loader

If you are using the Remote Loader, the following table lists the recommended security configurations for the driver.

Table 9-2 *Recommended Security Configuration for the Remote Loader*

Parameter	Description
Authentication ID	The account the driver uses to access the domain data. Use the domain logon name, for example Administrator.
Authentication Context	The DNS name of the domain controller. If you don't want to run the driver on your Active Directory domain controller, use *hostname* for the Negotiate method but use *hostname* or the IP address for the *simple* method.
Application Password	The password used for the *Authentication ID*.
Remote Loader Password	The password for the Remote Loader service.
Authentication Method	Select *negotiate*.
Digitally sign communications	Select *No*. Requires Windows 2003 or Windows 2000 with the most recent support pack, and Internet Explorer 5.5 SP2 or later on both servers.
Digitally sign and seal communications	Select *No*. Requires Windows 2003 or Windows 2000 with the most recent support pack, and Internet Explorer 5.5 SP2 or later on both servers.

Parameter	Description
Use SSL for encryption	Select *Yes*. SSL is required to perform a Subscriber password check, a Subscriber password set, and a Subscriber password modify when the driver shim isn't running on the domain controller.

9.3 Recommended Security Configurations when Using the Simple Authentication Method

SSL is recommended if you have selected the Simple authentication mechanism because Simple authentication passes passwords in clear text.

Table 9-3 *Recommended Security Configuration when Using the Simple Authentication Method*

Parameter	Description
Authentication ID	The account the driver uses to access the domain data. Use LDAP format for the *Authentication ID*. For example, cn=IDMadmin,cn=Users,dc=domain,dc=com
Authentication Context	IP address of domain controller.
Password	The password for the specified *Authentication ID*.
Digitally sign communications	Select *No*.
Digitally sign and seal communications	Select *No*.
Use SSL for encryption	Select *Yes*. SSL requires that the Microsoft server running the driver shim imports the domain controller's server certificate imported. For more information, see Securing Windows 2000 Server (http://www.microsoft.com/technet/security/prodtech/windows2000/secwin2k/swin2kad.mspx).

Troubleshooting

10

Refer to the following sections if you are experiencing a problem with the Active Directory driver.

10.1 Changes Are Not Synchronizing from the Publisher or Subscriber

To synchronize changes in Active Directory, the account used by the Identity Manager driver must have the proper rights set up. For information on the necessary rights, see Section 2.4, “Creating an Administrative Account,” on page 26.

If you use the default policies, you must also meet the requirements for the Create, Match, and Placement policies.

The attribute dirxml-uACLockout is not synchronized on the Publisher channel.

10.2 Using Characters Outside the Valid NT Logon Names

The default Subscriber creation policy generates an NT Logon Name (also known as the sAMAccountName and the Pre-Windows 2000 Logon Name) based on the relative distinguished name (RDN) of the account in the Identity Vault. The NT Logon name uses a subset of the ASCII character set. The default policy strips any character outside of the valid range before creating an object in Active Directory.

If the policy doesn't satisfy the business rules of your company, you can change the policy after import. Businesses that use Identity Vault account names outside of the traditional ASCII character set should pay particular attention to this policy.

10.3 Synchronizing c, co, and countryCode Attributes

When you use the Active Directory management console to select a country for a user, three attributes are set:

Table 10-1 *Attributes for Country*

Attribute	Description
c	Contains a two-character country code as defined by the ISO.
co	Contains a longer name for the country.
countryCode	Contains a numeric value (also defined by the ISO) that represents the country.

Because the ISO-defined numeric country codes are intended for use by applications that can't handle alphabetic characters, the default schema in the Identity Vault includes c and co but not countryCode.

Identity Manager is capable of mapping c and co. It can also map countryCode if you add a similar attribute to the eDirectory schema.

Active Directory's management console tries to keep all three of these attributes synchronized, so that when you set the country in the console, all three attributes have appropriate values. Some administrators might want a similar behavior when the attribute is set through Identity Manager. For example, you might want to configure the driver so that even though only c is in the Filter, co and countryCode are also set when a change for c is sent on the Subscriber channel.

10.4 Synchronizing Operational Attributes

Operation attributes are attributes that are maintained by an LDAP server that contains special operational information. Operation attributes are read-only. They can't be synchronized or changed.

10.5 Password Complexity on Windows 2003

Passwords must meet criteria that the password policies specify.

Complexities and requirements in Windows 2000/2003 password policies are different from complexities and requirements in eDirectory™.

If you plan to use Password Synchronization, create and use passwords that match the rules of complexity in both Active Directory and eDirectory. Otherwise, the passwords fail.

TIP: Make the password policies for both systems as similar to each other as you can. In a lab environment, disable strong-password functionality on Windows 2003 servers before installing the Active Directory driver. After the Active Directory driver is working properly, make sure that passwords used in eDirectory and Active Directory satisfy the rules of complexity for both systems. Then re-enable strong-password functionality on the Windows 2003 server.

For troubleshooting tips, see TID 10083320 (http://support.novell.com/cgi-bin/search/searchtid.cgi?/10083320.htm).

10.6 Tips on Password Synchronization

We recommend that you use a secure connection when you are synchronizing passwords. Vulnerable connections are between the following:

- The Metadirectory engine and the Remote Loader
- The Remote Loader and Active Directory

  This is true only when you run the Remote Loader remotely from the domain controller that you're connecting to.
- The Metadirectory engine and Active Directory when you aren't using the Remote Loader

  This is true only if the domain controller isn't local to this machine.

You can create a secure connection by doing one or more of the following:

- Configure SSL between the Metadirectory engine and the Remote Loader
- Run the Remote Loader on the domain controller
- Configure SSL between the driver shim and Active Directory

  This doesn't apply if you are running the driver on the domain controller that you're connecting to.

For password synchronization to work when the driver shim isn't running on the domain controller, you must have SSL configured.

10.6.1 Providing Initial Passwords

If you see an error about a password not complying when a user is initially created, you need to check your password policies.

For example, perhaps you want the Active Directory driver to provide the initial password for a user when the Active Directory driver creates a User object in the Identity Vault. When a user is created, the driver shim creates the user and then sets the password.

Because adding the user and setting the password are done separately, the new user in this example receives the default password, even if only momentarily. The password is soon updated because the Active Directory driver sends it immediately after adding the user.

If the default password doesn't comply with the eDirectory Password Policy for the user, an error is displayed. For example, if a default password that was created by using the user's surname is too short to comply with the Password Policy, you might see a -216 error saying that 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:

- Change the policy on the Publisher channel that creates default passwords, so that default passwords conform to the Password policies (created by using the *Manage Password Policies* option in Password Management) that have been defined for your organization in the Identity Vault. When the initial password comes from the authoritative application, it replaces the default password.

  This option is preferable. We recommend that a default password policy exist in order to maintain a high level of security within the system.
- Remove the policy on the Publisher channel that creates 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 eDirectory. The assumption for this option is that the password for the newly created User object eventually comes through the Publisher channel, so the user object exists without a password only for a short time.

These measures are especially important if the initial password doesn't come with the Add event, but comes in a subsequent event.

10.7 Where to Set the SSL Parameter

The SSL parameter in the driver configuration is for SSL between the Active Directory driver and Active Directory. It isn't for SSL between the Metadirectory engine and the Remote Loader. See "Encryption Using SSL" on page 22.

10.8 The Active Directory Account Is Disabled after a User Add on the Subscriber Channel

The default configuration maps the Identity Vault Logon Disabled attribute to the dirxml-uACAccountDisable bit of the userAccountControl attribute in Active Directory. A Subscriber Add operation might set Logon Disabled to False (account enabled), but the Publisher loopback of the Add operation reports that Logon Disabled is True (account disabled).

Additionally, inspecting the object in Active Directory might show that the account is disabled. This happens in part because of the way that the driver creates objects in Active Directory and in part because of a mismatch of policies between the driver and Active Directory itself.

If the account remains disabled in Active Directory after the provisioning cycle completes, you might have a mismatch between policies configured for the driver and policies enforced by Active Directory.

For example, consider a Password Required policy. If a user Add operation contains an invalid password (or no password at all), the account created in Active Directory should be disabled. But Active Directory might set the dirxml-uACPasswordNotRequired bit in userAccountControl without the driver's knowledge.

This causes the logon enable action of the Add operation to fail if the Add operation does not include a policy for dirxml-uACPasswordNotRequired. Therefore, the account stays disabled.

Later (perhaps almost immediately because of a Merge operation), the driver might attempt to enable the account again by setting Logon Disabled to False. If you want to override the Active Directory policy and ensure that accounts always require a password, you should set dirxml-uACPasswordNotRequired to False whenever Logon Disabled changes on the Subscriber channel.

10.9 Moving a Parent Mailbox to a Child Domain

If you move a parent mailbox to a mailbox store in a child domain by changing a user's homeMDB attribute, the driver fails the move. The error code returned is 0x80072030.

This error occurs on inter-domain moves. Moving an Exchange parent mailbox to a child domain isn't supported.

10.10 Restoring Active Directory

When you need to restore some or all of Active Directory, the driver might pick up interim events and perform unwanted actions on the Identity Vault. To restore safely, temporarily disable the driver during the restore operation and then bring the Identity Vault back into synchronization with Active Directory.

1. Disable the driver.
2. Delete the Dirxml-DriverStorage attribute on the driver object in the Identity Vault.
3. Restore Active Directory.
4. Set the Active Directory driver to Manual or Automatic startup.
5. Start the driver.
6. Re-migrate to find unassociated objects.

10.11 Moving the Driver to a Different Domain Controller

You can configure the driver to synchronize against a different domain controller by changing the driver Authentication Context parameter. When you restart the driver, the state information that the driver uses to track changes in Active Directory is invalid, and Active Directory might replay a large number of old events to bring the state back to the current time.

You can avoid this replay by removing the driver state information while updating the Authentication Context:

1. Stop the driver.
2. Delete the Dirxml-DriverStorage attribute on the Driver object in the Identity Vault.
3. Update the Authentication Context parameter.
4. Start the driver.

   This causes a resynchronization of associated objects in the Identity Vault.
5. Re-migrate to find unassociated objects in Active Directory.

10.12 Migrating from Active Directory

When migrating from Active Directory to the Identity Vault, you need to be concerned about object containment, DN references, and search limits on the Active Directory server. The general strategy for dealing with containment is to migrate containers first, objects that might be members of groups (including user objects) second, and groups last. If you have a moderately large number of objects to migrate, you need to adjust your strategy to handle the LDAP search constraints configured on the Active Directory server. You can change the constraints on the LDAP server or adjust your migration to get only a subset of objects each time (for instance, migrating container by container or migrating objects starting with A, B, etc.).

10.13 Setting LDAP Server Search Constraints

This section contains an example terminal session showing you how to use `ntdsutil.exe` to change the LDAP search parameters on your domain controller. You should only change these settings on the domain controller being used for Identity Manager synchronization for the duration of the migration. Write down the current configuration values and run `ntdsutil.exe` after migration completes to restore the original values. `ntdsutil.exe` can be run on any member server.

1. At a command prompt, type `ntdsutil`.
2. Type `LDAP Policies`, then press Enter.
3. Type `Connections`, then press Enter.
4. Type `Connect to domain` *`domain_name`*, then press Enter.
5. Type `Connect to server` *`server_name`*, then press Enter.
6. Type `Quit`, then press Enter.
7. Type `Show Values`, then press Enter.

```
C:\>ntdsutil
ntdsutil: LDAP Policies
ldap policy: Connections
server connections: Connect to domain raptor
Binding to \\raptor1.raptor.lab ...
Connected to \\raptor1.raptor.lab using credentials of locally logged on user.
server connections: Connect to server raptor1
Disconnecting from \\raptor1.raptor.lab...
Binding to raptor1 ...
Connected to raptor1 using credentials of locally logged on user.
server connections: Quit
ldap policy: Show Values

Policy                          Current(New)
MaxPoolThreads                  4
MaxDatagramRecv                 4096
MaxReceiveBuffer                10485760
InitRecvTimeout                 120
MaxConnections                  5000
MaxConnIdleTime                 900
MaxPageSize                     1000
MaxQueryDuration                120
MaxTempTableSize                10000
MaxResultSetSize                262144
```

```
MaxNotificationPerConn          5
MaxValRange                     1500
ldap policy: set MaxQueryDuration to 1200
ldap policy: set MaxResultSetSize to 6000000
ldap policy: Commit Changes
ldap policy: Quit
ntdsutil: Quit
Disconnecting from raptor1...
C:\>
```

10.14 Error Messages

The following sections contains a list of common error messages.

LDAP_SERVER_DOWN

Source: The status log or DSTrace screen.

Explanation: The driver can't open the LDAP port on the Active Directory domain controller configured for synchronization.

Possible Cause: The server named in the driver authentication context is incorrect.

Possible Cause: You are using an IP address for authentication context, and you have disabled non-Kerberos authentication to Active Directory. Kerberos requires a DNS name for authentication context.

Possible Cause: You have incorrectly configured the driver to use an SSL connection to Active Directory.

Action: The authentication context should hold the DNS name or the IP address of the domain controller you use for synchronization. If you leave the parameter empty, the driver attempts to connect to the machine that is running the driver shim (either the same server that is running Identity Manager, or the server hosting the Remote Loader).

Action: The driver shim can authenticate only using the pre-Windows 2000 Logon method or simple bind. If you have disabled NTLM, NTLM2, and simple bind on your network, you might receive the LDAP_SERVER_DOWN message.

Action: Something is wrong with the certificate that was imported to the driver shim server, or no certificate was imported.

LDAP_AUTH_UNKNOWN

Source: The status log or DSTrace screen.

Explanation: The driver is unable to authenticate to the Active Directory database.

Action: Try to authenticate to the Active Directory database again.

Solution: Unhide the driver parameter of retry-ldap-auth-unknown to allow the driver to retry the authentication when it fails.

1. Open the driver configuration file in the an XML editor.
2. Search for retry-ldap-auth-unknown.
3. Change the hide="true" to hide="false".
4. Access the driver parameters, see Section A.1.5, "Driver Parameters," on page 78 for more information.
5. Select *Driver Settings* > *Access Options* > *Retry LDAP Auth unknown* error, then select *Yes*.
6. Click *OK*, then restart the driver.

Error initializing connection to DirXML: SSL library initialization error: error:00000000:lib(0) :func(0) :reason(0)

Source: The status log or DSTrace screen.

Explanation: The Remote Loader cannot make an SSL connection to the Identity Manager engine.

Possible Cause: Incorrect format for the certificate file.

Action: If you are running a Windows 2003 R2 SP1 32-bit server, and are using a self-signed certificate in DER format, the connection fails. The certificate must have a Base64 format for the SSL connection to work.

An error was encountered while reading domain on the network 1208

Source: Password Sync Control Panel Applet on Windows 2008

Action: The Computer Browser service must be started to get the list of computers on the network. By default, it is disabled. Go to *Administrative tools* > *Services* and start the service.

10.15 Troubleshooting Driver Processes

Viewing driver processes is necessary to analyze unexpected behavior. To view the driver processing events, use DSTrace. You should only use it during testing and troubleshooting the driver. Running DSTrace while the drivers are in production increases the utilization on the Identity Manager server and can cause events to process very slowly. For more information, see "Viewing Identity Manager Processes" in the *Identity Manager 3.6 Common Driver Administration Guide*.

Driver Properties

A

This section provides information about the Driver Configuration and Global Configuration Values properties for the Active Directory driver. These are the only unique properties for drivers. All other driver properties (Named Password, Engine Control Values, Log Level, and so forth) are common to all drivers. Refer to "Driver Properties" in the *Identity Manager 3.6 Common Driver Administration Guide* for information about the common properties.

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

A.1 Driver Configuration

In iManager:

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

   By default, the Driver Configuration page is displayed.

In Designer:

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

The Driver Configuration options are divided into the following sections:

A.1.1 Driver Module

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

Table A-1 *Driver Modules*

Option	Description
Java	This option is not used with the Active Directory driver.
Native	Used to specify the name of the `.dll` file that is instantiated for the application shim component of the driver. If this option is selected, the driver is running locally. The driver `.dll` is: `addriver.dll`
Connect to Remote Loader	Used when the driver is connecting remotely to the connected system. Designer includes two suboptions: ◆ *Driver Object Password*: Specifies a password for the Driver object. If you are using the Remote Loader, you must enter a password on this page. Otherwise, the remote driver does not run. The Remote Loader uses this password to authenticate itself to the remote driver shim. ◆ *Remote Loader Client Configuration for Documentation*: Includes information on the Remote Loader client configuration when Designer generates documentation for the driver.

A.1.2 Driver Object Password (iManager Only)

Table A-2 *Driver Object Password*

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

A.1.3 Authentication

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

Table A-3 *Authentication Parameters*

Option	Description
Authentication ID or *User ID*	Specify a user application ID. This ID is used to pass Identity Vault subscription information to the application. Example: `Administrator`
Authentication Context or *Connection Information*	Specify the IP address or name of the server the application shim should communicate with.
Remote Loader Connection Parameters or *Host name* Port *KMO* *Other parameters*	Used only if the driver is connecting to the application through the Remote Loader. The parameter to enter is `hostname=xxx.xxx.xxx.xxx port=xxxx kmo=certificatename`, when the host name is the IP address of the application server running the Remote Loader server and the port is the port the Remote Loader is listening on. The default port for the Remote Loader is 8090. The `kmo` entry is optional. It is only used when there is an SSL connection between the Remote Loader and the Metadirectory engine. Example: `hostname=10.0.0.1 port=8090 kmo=IDMCertificate`
Driver Cache Limit (kilobytes) or *Cache limit (KB)*	Specify the maximum event cache file size (in KB). If it is set to zero, the file size is unlimited. Click *Unlimited* to set the file size to unlimited in Designer.
Application Password or *Set Password*	Specify the password for the user object listed in the *Authentication ID* field.
Remote Loader Password or *Set Password*	Used only if the driver is connecting to the application through the Remote Loader. The password is used to control access to the Remote Loader instance. It must be the same password specified during the configuration of the Remote Loader on the connected system.

A.1.4 Startup Option

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

Table A-4 *Startup Options*

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

Option	Description
Manual	The driver does not start when the Identity Manager server is started. The driver must be started through Designer or iManager.
Disabled	The driver has a cache file that stores all of the events. When the driver is set to Disabled, this file is deleted and no new events are stored in the file until the driver state is changed to Manual or Auto Start.
Do not automatically synchronize the driver	This option only applies if the driver is deployed and was previously disabled. If this is not selected, the driver re-synchronizes the next time it is started.

A.1.5 Driver Parameters

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

Table A-5 *Driver Parameters*

Option	Description
Driver Options > Authentication Options	
Show authentication options	The options are *show* or *hide*. It enables you to see and change the authentication options for the driver.
Authentication Method	The method of authentication to Active Directory. *Negotiate* uses Microsoft's security package to negotiate the logon type. Typically Kerberos or NTLM is selected. *Simple* uses LDAP style simple bind for logon. If you want to use Password Synchronization, select *Negotiate*.
Digitally sign communications	Select *Yes* to digitally sign communication between the driver shim and Active Directory. This does not hide the data from view on the network, but it reduces the change of security attacks. Signing only works when you use the *Negotiate* authentication method and the underlying security provider selects NTLM2 or Kerberos for its protocol. Do not use this option with SSL. Select *No* to have communications not signed.
Digitally sign and seal communications	Select *Yes* to digitally encrypt communication between the driver shim and the Active Directory database. Sealing only works when you the *Negotiate* authentication method and the underlying security provider selects NTLM2 or Kerberos for its protocols. Do not use this option with SSL. Select *No* to not have communication between the driver shim and the Active Directory database signed and sealed.

Option	Description
Use SSL for encryption	Select *Yes* to digitally encrypt communication between the driver shim and the Active Directory database. This option can be used with the *Negotiate* or *Simple* authentication methods. SSL requires that the Microsoft server running the driver shim imports the domain controller's server certificate imported. For more information, see Securing Windows 2000 Server (http://www.microsoft.com/technet/security/prodtech/windows2000/secwin2k/swin2kad.mspx).
Logon and impersonate	Select *Yes* to log on and impersonate the driver authentication account for CDOEXM (Collaboration Data Object for Exchange Management) and Password Set support. The driver performs a local logon. The authentication account must have the proper rights assignment. For more information, see Section 2.4, "Creating an Administrative Account," on page 26. If *No* is selected, the driver performs a network logon only.
Driver Options > Exchange Options	
Show Microsoft Exchange options	Select *show* to display the Microsoft Exchange options. These parameters control whether the driver shim uses the Microsoft CDOEXM Exchange management APIs and whether to interpret changes in the homeMDB attribute as a Move or a Delete of the mailbox. Select *hide* if you are not synchronizing Exchange accounts.
Exchange Management interface type (use-cdoexm/use-post-cdexm)	Exchange mailboxes can be controlled by calls to the Microsoft Exchange management system instead of regular attribute synchronization. When enabled, the driver intercepts changes to the Active Directory homeMDB attribute and calls into the desired interface for Exchange Management. The *use-cdoexm* option enables the use of the CDOEXM (Collaboration Data Objects for Exchange Management) subsystem. The *use-post-cdoexm*option requires use of Exchange 2007 or newer and requires installation of the Identity Manager Exchange service.
Allow Exchange mailbox move (yes/no)	Select *Yes* to enable the driver to intercept modifications to the Active Directory homeMDB attribute and call into the CDOEXM subsystem to move the mailboxes to the new message data store. Select *No* if you do not want mailboxes moved when the Active Directory account is moved.
Allow Exchange mailbox delete (yes/no)	Select *Yes* to enable the driver to intercept removals of the Active Directory homeMDB attribute and calls into the CDOEXM subsystem to delete the mailbox. Select *No* if you don't want to delete the mailbox account when the Active Directory account is deleted.
Driver Settings > Access Options	
Show access options	Select *show* to display the domain controller access options. These parameters control the scope of the Active Directory queries along with several Publisher polling and timeout parameters. Select *hide* to hide the domain controller access options.

Option	Description
Driver Polling Interval	Specify the number of minutes to delay before querying the Active Directory data base for changes. A larger number reduces the load on the Active Directory database, but it also reduces the responsiveness of the driver. The default value is 1 minute.
Publisher heartbeat interval	Allows the driver to send a periodic status message on the Publisher channel when there has been no Publisher channel traffic for the given number of seconds. The default value is 1 second.
Password Sync Timeout (minutes)	Specify the number of minutes for the driver to attempt to synchronize a given password. The driver does not try to synchronize the password after this interval has been exceeded. The recommended value is at least three times the value of the polling interval. For example, if the *Driver Polling Interval* is set to 10 minutes, set the *Password Sync Timeout* to 30 minutes. If this value is set to 0, password synchronization is disabled for this driver. The default value is 5 minutes.
Search domain scope	The driver reads information from other domains when objects in those domains are referenced. If the account you use for authentication has no rights in the other domain, the reads might fail. Select *Yes* to enable this option if you get access errors during regular operations.

A.2 Global Configuration Values

Global configuration values (GCVs) are values that can be used by the driver to control functionality. GCVs are defined on the driver or on the driver set. Driver set GCVs can be used by all drivers in the driver set. Driver GCVs can be used only by the driver on which they are defined.

The Active Directory driver includes many GCVs. You can also add your own if you need additional ones as you implement policies in the driver.

To access the driver's GCVs in iManager:

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

   or

   To add a GCV to the driver set, click *Driver Set*, then click *Edit Driver Set properties*.

To access the driver's GCVs in Designer:

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

   or

   To add a GCV to the driver set, right-click the driver set icon  , then click *Properties* > *GCVs*.

The global configuration values are organized as follows:

- Table A-6, "Driver Parameters," on page 81
- Table A-7, "Entitlements," on page 81
- Table A-8, "Password Management," on page 82
- Table A-9, "Name Mapping Policy," on page 83
- Table A-10, "Credential Provisioning," on page 84
- Table A-11, "Account Tracking," on page 85

Table A-6 *Driver Parameters*

Option	Description
Connected System or Driver Name	Contains the name of the connected system, application, or Identity Manager driver. This value is used by e-mail notification templates to identify the source of the notification messages.
Domain DNS Name	Specify the DNS name of the Active Directory domain managed by this driver.
Active Directory User Container	Specify the container where user objects reside in Active Directory.

Table A-7 *Entitlements*

Option	Description
Show entitlements configuration	Select *show* to display the global configuration values for entitlements. Select *hide* to not have the global configuration values displayed. The driver can use entitlements to manage user accounts and group memberships in Active Directory and to provision Exchange mailboxes. When using entitlements, the driver works in conjunction with entitlement agents such as the Identity Manager User Application or Role-Based Entitlements to control the conditions under provisioning occurs. See "Entitlements" on page 14 for more information.
Use User Account Entitlement	Select *True* to enable the driver to manage user accounts based on the driver's User Account entitlement. Select *False* to disable management of user accounts based on the entitlement.

Option	Description
When account entitlement revoked	Select the desired action in the Active Directory database when a User Account entitlement is revoked from an Identity Vault user. The options are *Disable Account* or *Delete Account*.
Use Group Entitlement	Select *True* to enable the driver to manage Active Directory group membership based on the driver's Group entitlement. Select *False* to disable management of group membership based on entitlement.
Use Exchange Mailbox Entitlement	Select *True* to enable the driver to manage Exchange mailboxes in Active Directory based on the driver's Exchange Mailbox entitlement. Select *False* to disable management of mailboxes based on the entitlement.

Table A-8 *Password Management*

Option	Description
Show password management policy	Select *show* to display the global configuration values for password management. Select *hide* to not have the password management global configuration values displayed. In Designer, you must click the  icon next to an option to edit it. This displays the Password Synchronization Options dialog box for a better view of the relationship between the different GCVs. In iManager, you should edit the Password Management Options on the *Server Variables* tab rather than under the GCVs. The Server Variables page has a better view of the relationship between the different GCVs. For more information about how to use the Password Management GCVs, see "Configuring Password Flow " in the *Identity Manager 3.6 Password Management Guide*.
Application accepts passwords from Identity Manager	If *True*, allows passwords to flow from the Identity Manager data store to the connected system.
Identity Manager accepts passwords from application	If *True*, allows passwords to flow from the connected system to Identity Manager.
Publish passwords to NDS password	Use the password from the connected system to set the non-reversible NDS® password in eDirectory™.
Publish passwords to Distribution Password	Use the password from the connected system to set the NMAS™ Distribution Password for Identity Manager password synchronization.
Require password policy validation before publishing passwords	If *True*, applies NMAS password policies during publish password operations. The password is not written to the data store if it does not comply.
Reset user's external system password to the Identity Manager password on failure	If *True*, on a publish Distribution Password failure, attempt to reset the password in the connected system by using the Distribution Password from the Identity Manager data store.

Option	Description
Notify the user of password synchronization failure via e-mail	If *True*, notify the user by e-mail of any password synchronization failures.

Table A-9 *Name Mapping Policy*

Option	Description
Show name mapping policy	Select *show* to display the global configuration values for the name mapping policy. Select *hide* to not have the global configuration values displayed. The following GCVs are used in the name mapping policy. If the policy does not meet your needs, you can modify it by editing the UserNameMap policies in the Subscriber and Publisher Command Transformation policies.
Full Name Mapping	Select *True* to synchronize the Identity Vault user's Full Name with the Active Directory object name and display name. This policy is useful when creating user accounts in Active Directory by using the Microsoft Management Console Users and Computers snap-in
Logon Name Mapping	Select *True* to synchronize the Identity Vault user's object name with the Active Directory Pre-Windows 2000 Logon Name (also known as the NT Logon Name and the sAMAccountName).
Use Principal Name Mapping	Allows you to choose a method for managing the Active Directory Logon Name (also known as the userPrincipalName). userPrincipalName takes the form of an e-mail address, such as usere@domain.com. Although the driver can place any value into userPrincipalName, it is not useful as a logon name unless the domain is configured to accept the domain name used with the name. ◆ *Follow Active Directory e-mail address* sets userPrincipalName to the value of the Active Directory mail attribute. This option is useful when you want the user's e-mail address to be used for authentication and Active Directory is authoritative for e-mail addresses. ◆ *Follow Identity Vault e-mail address* sets userPrincipalName to the value of the Identity Vault e-mail address attribute. This option is useful when you want the user's e-mail address to be used for authentication and the Identity Vault is authoritative for e-mail addresses. ◆ *Follow Identity Vault name* is useful when you want to generate userPrincipalName from the user logon name plus a hard-coded string defined in the policy. ◆ *None* is useful when you do not want to control userPrincipalName or when you want to implement your own policy.

Table A-10 *Credential Provisioning*

Option	Description
Show credential provisioning configuration	Select *show* to display the global configuration values for the Credential Provisioning policy. Select *hide* to not have the global configuration values displayed. The following GCVs are used in the name mapping policy. If the policy does not meet your needs, you can modify it by editing the UserNameMap policies in the Subscriber and Publisher Command Transformation policies.
Enable Credential Provisioning Policies	Select *True* to enable the driver's policies for provisioning credentials.
On user creation	Select *True* to provision new users with credentials
On user enable/disable	Select *True* to provision credentials to user accounts that have just been enabled and to deprovision credentials from user accounts that have been disabled.
On password changes	Select *True* to reprovision credentials when Identity Vault passwords change.
Application Credential ID	Specify the ID that SecureLogin uses to identify the login. This login is linked with the application in the SecureLogin client.
Application User ID Attribute	Specify the name of the attribute from which to retrieve application userid from.
Provision to Novell SecretStore	Select *True* if Novell SecretStore is to be used by the credential provisioning policies.
SecretStore Shared Secret Type	Select the shared secret type that Novell SecretStore is using.
Use Enhanced Protection Password	Select *True* if the Novell SecretStore Enhanced Protection Password is to be used. If true is selected then the named password 'secretstore-enhanced-proctection-password' must be appropriately set.
Provision to Novell SecureLogin Repository	Select *True* if the Novell SecureLogin repository is to be used by the credential provisioning policies.
Set Novell SecureLogin Passphrase	Select *True* to enable the SecureLogin passphrase to be set.
SecureLogin Passphrase Question	If you enabled the passphrase to be set, specify the passphrase question. The question needs to be one that can be verified against an Identity Vault attribute.
SecureLogin Passphrase Answer Value Attribute	If you enabled the passphrase to be set, specify the Identity Vault attribute used to verify the user's response to the passphrase question.

Table A-11 *Account Tracking*

Option	Description
Show Account Tracking Configuration	Select *show* to display the global configuration values for account tracking through Novell Sentinel. Select *hide* to not have the global configuration values displayed. The account tracking GCVs enable Sentinel to track Active Directory accounts based on unique identifiers that you define. You must have both Sentinel 6.1 and the Identity Manager Driver for Sentinel 6.1 installed in order to track account information. For information about Sentinel, see the Sentinel 6.1 Documentation Web site (http://www.novell.com/documentation/sentinel61). The Identity Manager Driver for Sentinel 6.1 is included with the Novell Compliance Management Platform. For information, see the Identity and Security Management product Web site (http://www.novell.com/identityandsecurity/).

Configuring the Driver for Use with an ADAM Instance

B

The Active Directory driver can be configured for use with a Microsoft Active Directory Application Mode (ADAM) instance. You import a configuration file to create a driver to connect to the ADAM instance.

There are multiple ways to configure your environment to synchronize the information. For example, Novell® recommends setting up your own certification authority (CA) in order to issue certificates that can be used for SSL connections to ADAM. If you already have server certificates, or if you have access to another CA that can issue valid certificates, you can ignore the steps that describe how to set up your own CA. Likewise, if you don't want to configure SSL (required if you want to set passwords on the Subscriber channel) then you can skip the section about configuring Certificate Services.

Any discussion of setting passwords is referring to the Subscriber channel (from Identity Manager to ADAM). Password synchronization on the Publisher channel (from ADAM to Identity Manager) is not currently possible, unless a regular user attribute (not the userPassword attribute) is used in ADAM to store the password.

B.1 Prerequisites

To achieve synchronization with an ADAM instance, you need the following items installed on one or more computers running Windows Server 2003. Windows Server 2003 is the only supported platform for this configuration.

- An Identity Manager server or Remote Loader where the Active Directory driver is configured.
- Internet Information Services (IIS) (must be installed before Certificate Services)
- Certificate Services
- A certification authority (can be your own standalone CA configured when you install Certificate Services)
- An ADAM instance (this example uses a standalone instance)

B.2 Installation Tasks

The following installation tasks must be completed in the order that they are listed. If a step is not necessary for your setup, you can skip it.

B.2.1 Installing Internet Information Services

If you want to set up your own CA in order to configure SSL on ADAM, you need to install Internet Information Services (IIS).

1. On your Windows Server 2003 machine, access the Control Panel, then click *Add or Remove Programs*.
2. In the left pane, select *Add/Remove Windows Components*.
3. Select *Application Server*, then click *Details*.
4. Select *Internet Information Services (IIS)*, then click *Details*.
5. Verify that at least the following are selected:
   - *Common Files*
   - *Internet Information Services Manager*
   - *World Wide Web Service*
6. Click *OK* twice, then click *Next* to complete the installation.

   You might be prompted to insert your original installation media for Windows Server 2003.

B.2.2 Installing Certificate Services

1. On your Windows Server 2003 machine, access the Control Panel, then click *Add or Remove Programs*.
2. In the left pane, select *Add/Remove Windows Components*.
3. Select to *Certificate Services*, then click *Next* to complete the installation.

B.2.3 Installing ADAM

1. On your Windows Server 2003 machine, access the Control Panel, then click *Add or Remove Programs*.
2. In the left pane, select *Add/Remove Windows Components*.
3. Select *Active Directory Services*, then click *Details*.
4. Select *Active Directory Application Mode (ADAM)*, then click *OK*.
5. Click *Next* to complete the installation.

   The Active Directory driver doesn't currently have a way to change the port when making a connection, so you need to use the defaults. If the values default to something else, you probably already have a service using those ports, and you might need to disable or uninstall the other service.
6. Click *Next*.
7. Select *Yes* to create an application directory partition, unless you plan on doing it later.
8. Specify the DN of the location where you'd like to synchronize users. For example, `CN=People,DC=adamtest1,DC=COM`.

9 Click *Next*.

10 Leave the default locations for data files and data recovery files, then click *Next*.

11 Select an account for the ADAM service, then click *Next*.

If you are installing ADAM on a server that is not already part of a domain, you might get a warning at this point. This is usually not a problem with ADAM, and you should continue with the installation.

12 Click *Next* to assign the current user (the one you are logged in as) rights to administrate ADAM.

13 Select *Import the selected LDIF files for this instance of ADAM*.

14 Select *MS-User.LDF*, then click *Add*.

15 Click *Next*.

16 Review the installation summary, then click *Next*.

B.2.4 Requesting and Installing the Server Certificate

1 On the server where you installed IIS and Certificate Services, specify the following address in a Web browser: `http://localhost/certsrv`.

2 You should see a welcome message from Certificate Services. If you do not, go back and make sure you have IIS and Certificate Services both installed.

3 The steps for requesting and installing a certificate are found at [.NET] Using SSL with ADAM (http://erlend.oftedal.no/blog/?blogid=7).

4 On your ADAM server, make sure you have the certificate installed in the following location in MMC: Certificates - Service (adaminstance) on Local Computer\ADAM_adaminstance\Personal.

5 On the Identity Manager server (or the Remote Loader computer) where the driver is running, make sure that you have only the CA certificate and make sure it is in Certificates - Current User\Trusted Root Certificates.

See Active Directory Application Mode: Frequently Asked Questions (http://www.microsoft.com/windowsserver2003/adam/ADAMfaq.mspx) for additional resources.

B.3 Configuration Tasks

B.3.1 Setting the Default Naming Context for Your ADAM Instance

1 Start the ADSI Edit application by selecting *Start* > *All Programs* > *ADAM* > *ADAM ADSI Edit*.

2 In the tree view, select the root item called *ADAM ADSI Edit*.

3 Under the *Action* menu, select *Connect to*.

4 In the *Connection name* field, type `Configuration`.

5 Select *Well-known naming context*. Make sure the value in the drop-down list is set to *Configuration*.

6 Set the other authentication credentials as appropriate, then click *OK*.

7 In the tree view, expand the *Configuration* item and those items underneath it until you can select the following entry:

```
CN=NTDS Settings,CN=ServerName$InstanceName,CN=Servers,
CN=Default-First-Site-Name, CN=Sites,CN=Configuration,CN={GUID}
```

Keep in mind that in the above DN, you should replace ServerName, InstanceName, and GUID with those values actually used in your ADAM instance.

8 Under the *Action* menu, select *Properties*.

9 Select the *msDS-DefaultNamingContext* attribute, then click *Edit*.

10 Specify the same value you used in Step 8 in Section B.2.3, "Installing ADAM," on page 88.

11 Click *OK* twice.

12 Restart your ADAM instance so the new default naming context takes effect.

B.3.2 Creating a User in ADAM with Sufficient Rights

For the driver to work properly, it is best to create a user object specifically for the driver to use. This user should only have the rights to do the work that is required. For more information, see Section 2.4, "Creating an Administrative Account," on page 26.

B.3.3 Creating the ADAM Driver

You can create the ADAM driver through Designer or iManager.

- "Creating the ADAM Driver in Designer" on page 90
- "Creating the ADAM Driver in iManager" on page 90

Creating the ADAM Driver in Designer

1 Open a project in Designer. In the Modeler, right-click the driver set and select *New* > *Driver*.

2 From the drop-down list, select *ADAM,* then click *Run*.

3 Configure the driver by filling in the fields. Specify information for your environment. For information on the settings, see Table B-1 on page 91.

4 After specifying parameters, click *Finish* to import the driver.

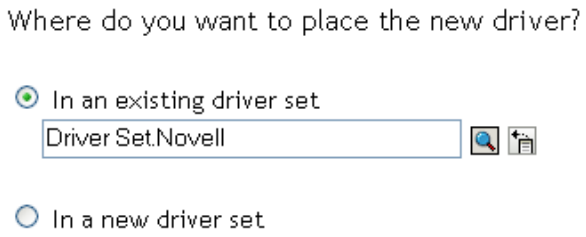
5 After the driver is imported, customize and test the driver.

6 After the driver is fully tested, deploy the driver into the Identity Vault. See "Deploying a Driver to an Identity Vault" in the *Designer 3.0 for Identity Manager 3.6 Administration Guide*.

Creating the ADAM Driver in iManager

1 In iManager, select *Identity Manager Utilities* > *Import Configuration*.

2 Select a driver set, then click *Next*.

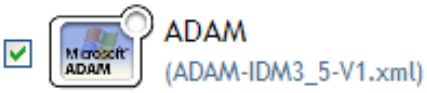


If you place this driver in a new driver set, you must specify a driver set name, context, and associated server.

3 Select how you want the driver configurations sorted:

- All configurations
- Identity Manager 3.5 configurations
- Identity Manager 3.0 configurations
- Configurations not associated with an Identity Manager version

4 Select the *ADAM* driver, then click *Next*.



5 Configure the driver by filling in the configuration parameters, then click *Next*. For information on the settings, see Table B-1 on page 91.

6 Define security equivalences, using a user object that has the rights that the driver needs to have on the server, then click *OK*.

Use the user created in Section B.3.2, “Creating a User in ADAM with Sufficient Rights,” on page 90.

7 Identify all objects that represent administrative roles and exclude them from replication, then click *OK*.

Exclude the security-equivalence object (for example, DriversUser) that you specified in Step 6. If you delete the security-equivalence object, you have removed the rights from the driver, and the driver can’t make changes to Identity Manager.

8 Click *Finish*.

NOTE: The parameters are presented on multiple screens. Some parameters are only displayed if the answer to a previous prompt requires more information to properly configure the policy.

Table B-1 *Configuration Parameters for the ADAM Driver*

Parameter	Description
Driver name	Specify the name of the driver object.

Parameter	Description
Authentication ID	Specify the name of the user object created in Section B.3.2, "Creating a User in ADAM with Sufficient Rights," on page 90. The name needs to be specified as a full LDAP DN. Example, `CN=IDM,CN=Users,DC=domain,DC=com`
Authentication Password	Specify the password of the user object with sufficient rights.
Authentication Context	Specify the DNS name or IP address of the ADAM instance server.
Driver Polling Interval	Specify the number of minutes to delay before querying Active Directory for changes. The default value is 1 minute.
Password Sync Timeout	Specify the number of minutes for the driver to attempt to synchronize a given password. The default value is 5 minutes.
Driver is Local/Remote	Configure the driver for use with the Remote Loader service by selecting *Remote*, or select *Local* to configure the driver for local use.
Name mapping policy selection	Select whether to accept the full policy or parts of the policy manually. The policy maps the Identity Vault Full Name attribute to the Active Directory object name and the policy maps the Active Directory Pre-Windows 2000 Logon Name to the Identity Vault user name.
Remote Host Name and Port	Remote option only. 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. This setting displays only if you set *Driver is Local/Remote* to *Remote*.
Driver Password	Remote option only. The Remote Loader uses the Driver Object Password to authenticate itself to the Identity Manager server. The password must be the same password that is specified as the Driver object password on the Remote Loader. This setting displays only if you set *Driver is Local/Remote* to *Remote*.
Remote Password	Remote option only. The Remote Loader password is used to control access to the Remote Loader instance. The password must be the same password that is specified as the Remote Loader password on the Remote Loader. This setting displays only if you set *Driver is Local/Remote* to *Remote*.

Parameter	Description
Full Name Mapping	Name mapping policy selection only. Select *Yes* if you want the Identity Vault Full Name attribute to be synchronized with the Active Directory object name and display name. This policy is useful when creating user accounts in Active Directory by using the Microsoft Management Console Users and Computer snap-in.
Logon Name Mapping	Select *Yes* if you want the Identity Vault object name synchronized with the Active Directory Pre-Windows 2000 Logon Name (also known as the NT Logon Name and the sAMAccountName). This policy is useful when creating user accounts in Active Directory by using the Microsoft Management Console Users and Computer snap-in.
Import will proceed to Active Directory logon name policy selections	Select *OK*.
Base container in eDirectory	Specify the base container in the Identity Vault for synchronization. This container is used in the Subscriber Matching policies to limit the Identity Vault objects being synchronized and in the Publisher Placement policies when adding objects to the Identity Vault. New users are placed in this container by default. Use the dot format. For example, `users.myorg`.
Publisher Placement	*Mirrored* places objects hierarchically within the base container. *Flat* places objects strictly within the base container. This selection builds the default Publisher Placement policies. **NOTE:** If you select *Mirrored*, the driver assumes that the structure of the eDirectory™ database is the same in Active Directory as the eDirectory base container. If the structure is not the same, the objects are not placed properly. Create the same structure in Active Directory that exists in eDirectory, or migrate the eDirectory containers before migrating User objects.

Base container in Active Directory	Specify the base container in Active Directory, in LDAP format. New users are placed in this container by default. For example,
	
	`CN=Users,DC=MyDomain,DC=com`
	
	If the target container doesn't exist, you must create it and make sure it is associated with the eDirectory base container before trying to add users to this container.
	
	If you are creating or using a container other than Users in Active Directory, the container is an OU, not a CN. For example,
	
	`OU=Sales,OU=South,DC=MyDomain,DC=com`

Parameter	Description
Active Directory Placement	*Mirrored* places the objects hierarchically within the base container. *Flat* places objects strictly within the base container. This selection builds the default Subscriber Placement policies. **NOTE:** If you select *Mirrored*, the driver assumes that the structure of the Active Directory database is the same in eDirectory as the Active Directory base container. If the structure is not the same, the objects are not placed properly. Create the same structure in eDirectory that exists in Active Directory, or migrate the Active Directory containers before migrating User objects.
Configure Data Flow	Establishes the initial driver filter that controls the classes and attributes that will be synchronized. The purpose of this option is to configure the driver to best express your general data flow policy. It can be changed after import to reflect specific requirements. *Bidirectional* sets classes and attributes to synchronize on both the Publisher and Subscriber channels. A change in either the Identity Vault or Active Directory is reflected on the other side. Use this option if you want both sides to be authoritative sources of data. *AD to Vault* sets classes and attributes to synchronize on the Publisher channel only. A change in Active Directory is reflected in the Identity Vault, but Identity Vault changes are ignored. Use this option if you want Active Directory to be the authoritative source of data. *Vault to AD* sets classes and attributes to synchronize on the Subscriber channel only. A change in the Identity Vault is reflected in Active Directory, but Active Directory changes are ignored. Use this option if you want the Identity vault to be the authoritative source of data. **IMPORTANT:** Delete, Move, and Rename events are independent of the filter. It does not matter which option you select; these events are processed by the driver. If you do not want these events to synchronize, you must change the default configuration of the driver. You can use one of the predefined policies that comes with Identity Manager 3.6 to change Delete events into Remove Association events. For more information, see “Creating and Managing Policies” in the *Policies in Designer 3.0* guide. To block Move and Rename events, you must customize the driver.

Parameter	Description
Password Failure Notification User	Password synchronization policies are configured to send e-mail notifications to the associated user when password updates fail. You have the option of sending a copy of the notification e-mail to another user, such as a security administrator. If you want to send a copy, enter or browse for the DN of that user. Otherwise, leave this field blank.
Group membership policy	Configure Elements option only. Group membership in Active Directory can be controlled by synchronizing the membership list or by using Entitlements. *Entitlements* uses the Workflow service or the Role-Based Entitlements to assign group membership. *Synchronize* uses policies to synchronize the group membership list. *None* does not synchronize group membership information.
User Principal Name Mapping	Allows you to choose a method for managing the Active Directory Windows 2000 Logon Name (also known as the userPrincipalName). userPrincipalName takes the form of an e-mail address, such as in usere@domain.com. Although the shim can place any value into userPrincipalName, it is not useful as a logon name unless the domain is configured to accept the domain name used with the name. *Follow Active Directory e-mail address* sets userPrincipalName to the value of the Active Directory mail attribute. This option is useful when you want the user's e-mail address to be used for authentication and Active Directory is authoritative for e-mail addresses. *Follow Identity Vault e-mail address* sets userPrincipalName to the value of the Identity Vault e-mail address attribute. This option is useful when you want the user's e-mail address to be used for authentication and the Identity Vault is authoritative for e-mail addresses. *Follow Identity Vault name* is useful when you want to generate userPrincipalName from the user logon name plus a hard-coded string defined in the policy. *None* is useful when you do not want to control userPrincipalName or when you want to implement your own policy.

Changing Permissions on the CN=Deleted Objects Container

C

When an Active Directory object is deleted, a small portion of the object remains for a specified time so that other domain controllers that are replicating changes become aware of the deletion. By default, only the System account and members of the Administrators group can view the contents of this container. This section describes how to modify the permissions on the CN=Deleted Objects container.

Changing permissions on the Deleted Objects container might be necessary if you have enterprise applications or services that bind to Active Directory with a non-System or non-Admin account and poll for directory changes.

This process requires `dscals.exe` from the Active Directory Application Mode (ADAM) package. This version is an upgrade from the one in the Windows Server 2003 Support Tools and now supports the required capabilities. The ADAM Administration Tools are supported on Windows XP Professional, Windows Server 2003 Standard Edition, Windows Server 2003 Enterprise Edition, and Windows Server 2003 Datacenter Edition.

To get and install the ADAM Administration Tools:

1. Download the ADAM retail package from the ADAM Web page (http://www.microsoft.com/downloads/details.aspx?FamilyId=9688F8B9-1034-4EF6-A3E5-2A2A57B5C8E4&displaylang=en).
2. Double-click the downloaded file and provide a directory to extract the archive into.
3. Launch the Active Directory Application Mode Setup Wizard by double-clicking `adamsetup.exe`, then click *Next*.
4. Review and accept the license terms, then click *Next*.
5. Select *ADAM administration tools only*, then click *Next*.
6. Review the selections, then click *Next*.
7. When setup has concluded, click *Finish*.

After the ADAM Administration Tools are installed, modify the permissions on the CN=Deleted Objects container:

1. Log in with a user account that is a member of the Domain Admins group.
2. *Click Start > All Programs > ADAM > ADAM Tools Command Prompt*.
3. At the command prompt, enter the following command:

   ```
   dsacls "CN=Deleted Objects,DC=Contoso,DC=com" /takeownership
   ```

   Substitute the distinguished name of the Deleted Objects container for your own domain.

   Each domain in the forest will have its own Deleted Objects container.

   The following output should be displayed:

```
Owner: Contoso\Domain Admins
  Group: NT AUTHORITY\SYSTEM
  Access list:
  {This object is protected from inheriting permissions from the parent}
  Allow BUILTIN\Administrators  SPECIAL ACCESS
                                LIST CONTENTS
                                READ PROPERTY
  Allow NT AUTHORITY\SYSTEM     SPECIAL ACCESS
                                DELETE
                                READ PERMISSONS
                                WRITE PERMISSIONS
                                CHANGE OWNERSHIP
                                CREATE CHILD
                                DELETE CHILD
                                LIST CONTENTS
                                WRITE SELF
                                WRITE PROPERTY
                                READ PROPERTY
The command completed successfully
```

4 To grant a security principal permission to view the objects in the CN=Deleted Objects container, enter the following command:

```
dsacls "CN=Deleted Objects,DC=Contoso,DC=com" /g CONTOSO\JaneDoe:LCRP
```

In this example, the user CONTOSO\JaneDoe has been granted List Contents and Read Property permissions on the container. These permissions are sufficient to allow the user to view the contents of the Deleted Objects container. However, these permissions don't allow the user to make any changes to objects in that container. These permissions are equivalent to the default permissions granted to the Administrators group. By default, only the System account has permission to modify objects in the Deleted Objects container.

The following output should be displayed:

```
  Owner: CONTOSO\Domain Admins
Group: NT AUTHORITY\SYSTEM
  Access list:
  {This object is protected from inheriting permissions from the parent}
  Allow BUILTIN\Administrators  SPECIAL ACCESS
                                LIST CONTENTS
                                READ PROPERTY
Allow NT AUTHORITY\SYSTEM     SPECIAL ACCESS
                                DELETE
                                READ PERMISSONS
                                WRITE PERMISSIONS
                                CHANGE OWNERSHIP
                                CREATE CHILD
                                DELETE CHILD
                                LIST CONTENTS
                                WRITE SELF
                                WRITE PROPERTY
                                READ PROPERTY
Allow CONTOSO\JaneDoe         SPECIAL ACCESS
                                LIST CONTENTS
                                READ PROPERTY
The command completed successfully.
```

The user CONTOSO\JaneDoe now has permissions to view deleted objects in the CONTOSO domain.

Provisioning Exchange Accounts

D

The Active Directory driver can be configured to provision Active Directory accounts as well as Exchange 2000, Exchange 2003, and Exchange 2007 accounts.

The driver can synchronize Exchange 2000 and Exchange 2003 accounts or Exchange 2007 accounts. It cannot synchronize all types of Exchange accounts at the same time. If you have multiple types of Exchange accounts, you must set up two separate drivers to synchronize the Exchange 2007 accounts and Exchange 2000 and 2003 accounts.

D.1 Provisioning Exchange 2000 and 2003 Accounts

There are two different ways to provision the Exchange 2000 and 2003 mailbox accounts with the Active Directory driver. You can set attributes on User objects so a Microsoft program (the Recipient Update Service) can use this information to provision to users to the Exchange database. Or you enable Collaboration Data Objects for Exchange Management (CDOEXM), which is the method documented in this section.

CDOEXM is an API that is provided by Microsoft. The Active Directory driver uses this API to provision the Exchange accounts.

With CDOEXM enabled, an Exchange 2000 or 2003 mailbox is provisioned by setting the homeMDB attribute. When the homeMDB attribute is set, the driver automatically sets all required attributes. The driver can create, delete, and move mailboxes. The mailbox moves that are supported are only interdomain moves.

The homeMDB attribute is set during initial configuration, but you can change the setting by modifying the driver policy. To find out what the homeMDB attribute is for your Exchange system, see Section 4.1, “Gathering Configuration Information,” on page 33.

To configure the driver to synchronize Exchange 2000 and 2003 accounts:

1. If the server that is running the driver is a non-Exchange server, make sure the Exchange Management tools is installed on this server.
2. Verify that the authentication account for the driver has enough rights to create, delete, or move Exchange accounts.
3. If the driver is running on a member server, you must use SSL and you must run the Remote Loader service as a specific domain user with enough rights to delete, create, or move Exchange accounts.
4. Run the Active Directory Discovery tool to find out what the homeMDB attribute is for the Exchange 2000 or 2003 system. For more information, see Section 4.1, “Gathering Configuration Information,” on page 33.

5 Specify the configuration parameters to provision the Exchange mailboxes, when you are creating a driver object. See Table D-1 for a list of Exchange parameters. See Chapter 4, "Creating a New Driver," on page 33 for information on how to create the driver object.

6 Verify that you have selected *use-cdoexm* to provision the Exchange 2000 and 2003 mailboxes. See Exchange Management interface type for more information.

Table D-1 *Exchange Provisioning Configuration Parameters*

Parameter	Description
Exchange Policy	Exchange provisioning can be handled by a driver policy, Entitlements, or skipped entirely. A user can be assigned a mailbox in Exchange (the user is mailbox enabled) or have information about a foreign mailbox stored in the Identity Vault record (the user is mail enabled). When using Entitlements, an external service such as the Workflow service or Role-Based Entitlements makes these decisions and the driver policy simply applies them. *Implement in policy* uses the policies in the driver instead of Entitlements to assign Exchange mailboxes. When using the driver policy, the decision to mailbox-enable or mail-enable a user, plus the Exchange message database where the account will reside, is controlled completely in the policy. When *None* is selected, the default configuration does not create Exchange mailboxes but does synchronize the Identity Vault Internet E-Mail Address with the Active Directory mail attribute.
Exchange Management interface type	The driver cannot provision both Exchange 2007 mailboxes and Exchange 2000 and 2003 mailboxes. This option allows you to select which type of mailboxes the driver can provision. *use-cdoexm* synchronizes Exchange 2000 and Exchange 2003 accounts. *use-post-cdoexm* synchronizes Exchange 2007 accounts.
Allow Exchange mailbox move (yes/no)	When this option is enabled, the driver shim intercepts modifications to the Active Directory homeMDB attribute to move the mailbox to the new message data store. *Yes* moves the Exchange mailbox. *No* does not move the Exchange mailbox.
Allow Exchange mailbox delete (yes/no)	When this option is enabled, the driver shim intercepts removal for the Active Directory homeMDB attribute to delete the mailbox. *Yes* allows the Exchange mailbox to be deleted. *No* does not allow the Exchange mailbox to be deleted.

Parameter	Description
Default Exchange MDB	Specify the default Exchange Message Database (MDB). To obtain the correct name for the Exchange MDB, see Section 4.1, "Gathering Configuration Information," on page 33. For example, `[CN=Mailbox Store (CONTROLLER),CN=First Storage Group,CN=InformationStore,CN=CONTROLLER,CN=Servers,CN=First Administrative Group,CN=Administrative Groups,CN=Domain,CN=Microsoft Exchange,CN=Services,CN=Configuration,DC=Domain,DC=com]` The driver can be updated to manage additional MDBs after the import is complete.

D.2 Provisioning Exchange 2007 Accounts

Exchange 20007 no longer supports CDOEXM for mailbox management. In order to provision the Exchange 2007 mailboxes, the Active Directory driver uses Windows PowerShell* in the form of a service.

This service is installed on the server that is running the Active Directory driver. If you decided to run the driver locally, the driver is installed on the Identity Manager server. If you decided to run the driver remotely, the driver is installed on the same server as the Remote Loader service.

The service listens on a default port of 8097. This is set when the service is installed. It is stored in the registry key `HKEY_LOCAL_MACHINE\SOFTWARE\Novell\IDM_AD_EX_SERVICE`. The value can be edited if necessary. If you edit the registry key, both the service and the driver must be restarted.

The Active Directory driver creates, moves, and disables Exchange 2007 mailboxes.

To provisions the Exchange 2007 mailboxes, the following steps must be completed:

1. Meet the prerequisites
2. Install the service
3. Configure the driver

Prerequisites

On the server where the driver will run, whether that is as a Remote Loader service or if the driver is installed locally, the following items must be installed:

- ❑ Microsoft .NET Framework version 2.0 or above.
- ❑ Exchange 2007 Management Tools for the correct platform: 32-bit or 64-bit.

Installing the Service

To install the service, you must use the .NET Framework `InstallUtil.exe` utility. The version folder is the current version of the .NET Framework that is installed.

The default location for a 32-bit server is `C:\WINDOWS\Microsoft.Net\Framework\version\InstallUtil.exe`.

The default location for a 64-bit server is `C:\WINDOWS\Microsoft.Net\Framework64\version\InstallUtil.exe`.

To use `InstallUtil.exe`:

1. Open a .NET command prompt.
2. Issue the command `InstallUtil IDMExService.exe` to register the service and create the correct registry entries.
3. To start the service, select *Start* > *Control Panel* > *Administrative Tools* > *Services*.
4. Right-click the service *IDM_AD_Ex2007_Service*, then select *Start*.

To uninstall the service, issue the command `InstallUtil /u IDMExService.exe`.

Configuring the Driver

You need to create a new driver object and select the correct fields to enable provisioning with Exchange 2007 or modify the existing driver.

To create a new driver:

1. When you are creating a driver object, specify the configuration parameters to provision the Exchange 2007 mailboxes.

   See Table D-1 for a list of Exchange parameters. See Chapter 4, “Creating a New Driver,” on page 33 for information on how to create the driver object.
2. Verify that you have selected *use-post-cdoexm* to provision Exchange 2007 mailboxes. See “Exchange Management interface type” on page 100 for more information.
3. Start the driver to provision the Exchange 2007 mailboxes.

To modify an existing driver in Designer:

1. Right-click the Active Directory driver in the Modeler, then select *Properties*.
2. Select *Driver Configuration* > *Driver Parameters* > *Edit XML*.
3. Search for the heading `<header display-name="Exchange Options"/>`.
4. Change the following lines:

Old XML	New XML
<definition display-name="Use CDOEXM for Exchange (yes/no)" name="use-CDOEXM" type="enum">	<definition display-name="Exchange Management interface type (use-cdoexm/use-post-cdoexm)" name="exch-api-type" type="enum">
<enum-choice display-name="Yes">yes</enum-choice>	<enum-choice display-name="use-cdoexm">use-cdoexm</enum-choice>
<enum-choice display-name="No">no</enum-choice>	<enum-choice display-name="use-post-cdoexm">use-post-cdoexm</enum-choice>
<definition display-name="Allow CDOEXM Exchange mailbox move (yes/no)" name="cdoexm-move" type="enum">	<definition display-name="Allow Exchange mailbox move (yes/no)" name="exch-move" type="enum">
<definition display-name="Allow CDOEXM Exchange mailbox delete (yes/no)" name="cdoexm-delete" type="enum">	<definition display-name="Allow Exchange mailbox delete (yes/no)" name="exch-delete" type="enum">

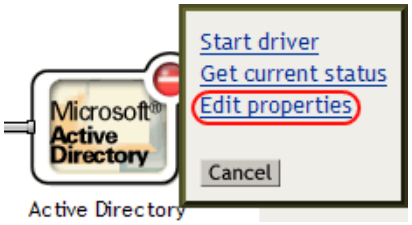
5 Click *OK* twice to save the changes.

To modify an existing driver in iManager:

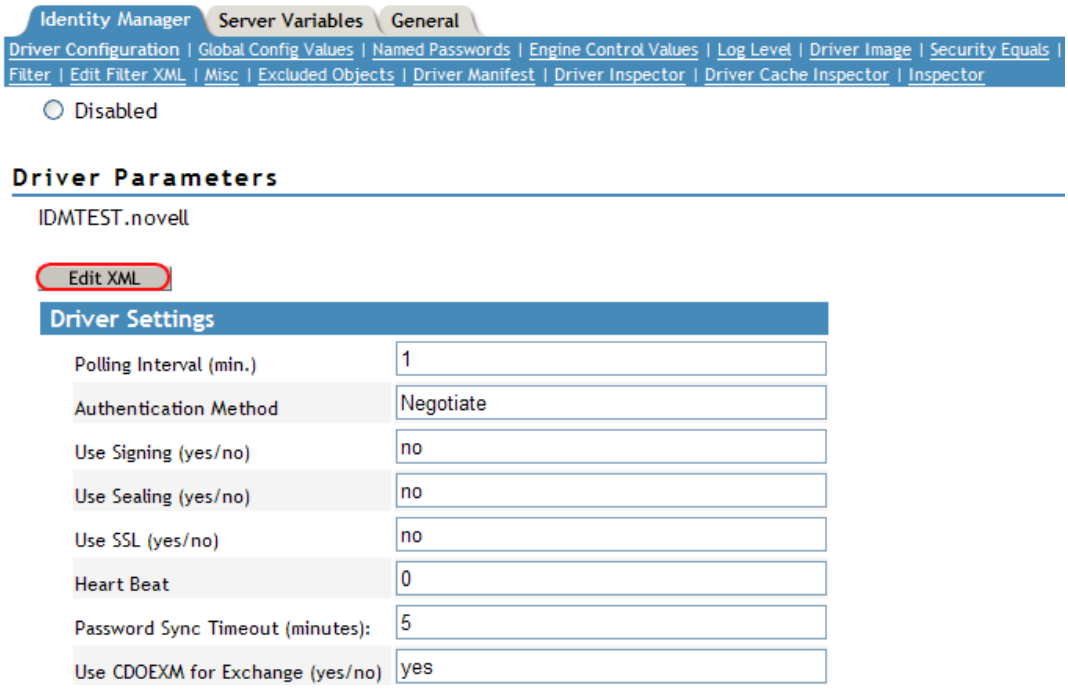
1 Select *Identity Manager* > *Identity Manager Overview*.

2 Select the driver set where the Active Directory driver is stored, then click *Search*.

3 Click the upper right corner of the Active Directory driver, then click *Edit Properties*.



4 In the *Driver Configuration* tab, click *Edit XML* under *Driver Parameters*.



5 Click the *Enable XML editing* check box.

6 Search for the heading `<header display-name="Exchange Options"/>`.

7 Change the following lines:

Old XML	New XML
`<definition display-name="Use CDOEXM for Exchange (yes/no)" name="use-CDOEXM" type="enum">`	`<definition display-name="Exchange Management interface type (use-cdoexm/use-post-cdoexm)" name="exch-api-type" type="enum">`
`<enum-choice display-name="Yes">yes</enum-choice>`	`<enum-choice display-name="use-cdoexm">use-cdoexm</enum-choice>`
`<enum-choice display-name="No">no</enum-choice>`	`<enum-choice display-name="use-post-cdoexm">use-post-cdoexm</enum-choice>`
`<definition display-name="Allow CDOEXM Exchange mailbox move (yes/no)" name="cdoexm-move" type="enum">`	`<definition display-name="Allow Exchange mailbox move (yes/no)" name="exch-move" type="enum">`
`<definition display-name="Allow CDOEXM Exchange mailbox delete (yes/no)" name="cdoexm-delete" type="enum">`	`<definition display-name="Allow Exchange mailbox delete (yes/no)" name="exch-delete" type="enum">`

8 Click *OK* twice to save the changes.

Trace Levels

E

The driver supports the following trace levels:

Table E-1 *Supported Trace Levels*

Level	Description
0	No trace messages are displayed or logged
1	Basic trace messages are displayed and logged
2	Level 1 messages and the contents of XML documents that are used during event processing are displayed and logged
3	Trace Level 2 messages and extensive rule processing messages are displayed and logged the above plus template instantiations