NetIQ® Identity Manager

Driver for Multi-Domain Active Directory Implementation Guide

November 2022

Legal Notice

For information about NetIQ trademarks, see https://www.netiq.com/company/legal/.

Copyright (C) 2019 NetIQ Corporation. All rights reserved.

Contents

About this Book and the Library **7**
About NetIQ Corporation **9**

1 Understanding the Multi-Domain Active Directory Driver **11**

Key Terms . . . 11
Identity Manager . . . 12
Connected System . . . 12
Identity Vault . . . 12
Identity Manager Engine . . . 12
Multi-Domain Active Directory Driver . . . 12
Driver Shim . . . 13
.NET Remote Loader . . . 13
Data Transfers Between Systems . . . 13
Support for Standard Driver Features . . . 14
Supported Operations . . . 14
Remote Platforms . . . 14
Multi-Domain Support . . . 15
PowerShell Command Support . . . 15
Entitlements and Permission Collection and Reconciliation Service . . . 15
Automatic Domain Controller Discovery and Failover . . . 17
Domain Controller Failover . . . 18
Password Synchronization Support . . . 18
Data Synchronization Support . . . 18
Nested Group Synchronization Support . . . 18
Scalability . . . 18
Multiple Active Directory User Account Support . . . 18
Default Driver Configuration . . . 18
User Object Name Mapping . . . 19
Data Flow . . . 19
Checklist for Enabling User Synchronization . . . 22

2 Preparing Multi-Domain Active Directory **23**

Driver Prerequisites . . . 23
Deploying the Multi-Domain Active Directory Driver . . . 24
Remote Installation on Windows and Other Platforms . . . 24
Remote Installation on a Windows Member Server . . . 25
Securing Driver Communication . . . 26
Authentication Methods . . . 26
Encryption Using SSL . . . 26
Creating an Administrative Account . . . 29
Configuring System Permissions . . . 30
Windows Message Queuing Permissions . . . 31
Becoming Familiar with Driver Features . . . 31
Schema Changes . . . 31
Structuring eDirectory Container Hierarchy . . . 31

Moving Cross Domain Objects . . . 32
Automatic Failover . . . 32
Multivalue Attributes . . . 33
Using Custom Boolean Attributes to Manage Account Settings . . . 33
Provisioning Exchange Mailboxes . . . 34
Expiring Accounts in Active Directory . . . 34
Driver Response Behavior . . . 35

3 Installing the Driver Files 37

Preparing for Driver Installation . . . 37
Installing the Driver Files . . . 37
Configuring the Multi-Domain Active Directory Driver . . . 37

4 Creating a New Driver 39

Creating the Driver in Designer . . . 39
Importing the Current Driver Packages . . . 40
Installing the Driver Packages . . . 40
Configuring Domain Connections for Multi-Domain Active Directory Driver . . . 45
Configuring the Driver . . . 47
Deploying the Driver . . . 48
Starting the Driver . . . 49
Activating the Driver . . . 49
Adding Packages to an Existing Driver . . . 50

5 Upgrading an Existing Driver 53

Supported Upgrade Paths . . . 53
What's New in Version 5.1.0 . . . 53
Upgrading the Driver . . . 53
Upgrading the Installed Packages . . . 53
Applying the Driver Patch . . . 54

6 Synchronizing Passwords 57

Securing Driver Connections . . . 57
Setting Up Password Synchronization Filters . . . 58
Allowing Remote Access to the Registry . . . 58
Not Allowing Remote Access to the Registry . . . 62
Retrying Synchronization after a Failure . . . 66
Retrying after an Add or Modify Event . . . 66
Password Expiration Time . . . 66
Disabling Password Synchronization on a Driver . . . 69
Diagnosing Password Synchronization Issues . . . 69

7 Managing Active Directory Groups and Exchange Mailboxes 71

Managing Groups . . . 71
Managing Microsoft Exchange Mailboxes . . . 72

8 Provisioning Exchange Mailboxes **75**

Setting Up Domains for Exchange Provisioning 75
Setting Up Exchange Server Permissions 76
Supported Operations on Exchange Mailboxes 77
Configuring the Driver 77
Configuring the Driver to Support Database Load Balancing 78
Support for Multiple Exchange Server in the Environment 78

9 Configuring PowerShell Support **79**

Overview of PowerShell Functionality 79
System Requirements 79
Implementing PowerShell Cmdlets in the Multi-Domain Active Directory Driver 80
Sample Active Directory Policy Rule with Cmdlets 80
Available Active Directory and Exchange Cmdlets 81
Creating Active Directory Policies with Cmdlets 81
Verifying Active Directory Cmdlet Execution 82

10 Security Best Practices **83**

Security Considerations 83
Default Configuration of the Security Parameters 83
Recommended Security Configurations for the Simple Authentication Method 85

11 Troubleshooting **87**

Data Protection from Unauthorized Access 87
Error Displays When Stopping the Remote Loader Service 87
Issues with User Account Reconciliation in Resource Catalog 88
Changes Are Not Synchronizing from the Publisher or Subscriber 88
Using Characters Outside the Valid NT Logon Names 88
Synchronizing c, co, and countryCode Attributes 88
Synchronizing Operational Attributes 89
Password Complexity on Windows Server 89
Tips on Password Synchronization 90
Providing Initial Passwords 90
Where to Set the SSL Parameter 91
Password Filter Synchronization State Definitions 91
The Active Directory Account Is Disabled after a User Add on the Subscriber Channel 92
Restoring Active Directory 93
Setting LDAP Server Search Constraints 93
Error Messages 94
Driver Binaries Fail to Load on Windows 2012 Devices 95
Setting a Password in Multi-Domain Active Directory Driver Resets the eDirectory Password Expiration Date to the Current Date 96
Troubleshooting Driver Processes 96

A Known Issues **97**

Loss of Existing Resource Assignments During Upgrade 97
Issue with Enabling Out of Band Sync for Attributes with Distinguished Name Syntax 97

Applying the Latest Driver Package Does Not Change the Default Setting of Enable Service Channel ECV 97
32-Bit pwfilter.dll File Is Not Available 98

B Driver Properties 99

Driver Configuration 99
Driver Module 100
Driver Object Password 100
Authentication 100
Startup Option 101
Driver Parameters 102
Subscriber Settings 104
Publisher Settings 105
ECMAScript (Designer Only) 105
Global Configurations (Designer Only) 105
Global Configuration Values 105
Managed System Information 106
Password Synchronization 107
Configuration 108
Account Tracking 109
Entitlements 110

C Migrating Users Per Domain 113

D Trace Levels 115

E Microsoft Windows Events 117

About this Book and the Library

The *Identity Manager Driver for Multi-Domain Active Directory Implementation Guide* explains how to install, configure, and manage the Identity Manager Driver for Multi-Domain Active Directory.

Intended Audience

This book provides information for individuals responsible for understanding administration concepts and implementing a secure, distributed administration model.

Other Information in the Library

For more information about the library for Identity Manager, see the following resources:

- Identity Manager documentation website (https://www.netiq.com/documentation/identity-manager-47/)
- Identity Manager drivers documentation website (https://www.netiq.com/documentation/identity-manager-47-drivers/)

About NetIQ Corporation

We are a global, enterprise software company, with a focus on the three persistent challenges in your environment: Change, complexity and risk—and how we can help you control them.

Our Viewpoint

Adapting to change and managing complexity and risk are nothing new

In fact, of all the challenges you face, these are perhaps the most prominent variables that deny you the control you need to securely measure, monitor, and manage your physical, virtual, and cloud computing environments.

Enabling critical business services, better and faster

We believe that providing as much control as possible to IT organizations is the only way to enable timelier and cost effective delivery of services. Persistent pressures like change and complexity will only continue to increase as organizations continue to change and the technologies needed to manage them become inherently more complex.

Our Philosophy

Selling intelligent solutions, not just software

In order to provide reliable control, we first make sure we understand the real-world scenarios in which IT organizations like yours operate—day in and day out. That's the only way we can develop practical, intelligent IT solutions that successfully yield proven, measurable results. And that's so much more rewarding than simply selling software.

Driving your success is our passion

We place your success at the heart of how we do business. From product inception to deployment, we understand that you need IT solutions that work well and integrate seamlessly with your existing investments; you need ongoing support and training post-deployment; and you need someone that is truly easy to work with—for a change. Ultimately, when you succeed, we all succeed.

Our Solutions

- Identity & Access Governance
- Access Management
- Security Management
- Systems & Application Management
- Workload Management
- Service Management

Contacting Sales Support

For questions about products, pricing, and capabilities, contact your local partner. If you cannot contact your partner, contact our Sales Support team.

Worldwide:	www.netiq.com/about_netiq/officelocations.asp
United States and Canada:	1-888-323-6768
Email:	info@netiq.com
Web Site:	www.netiq.com

Contacting Technical Support

For specific product issues, contact our Technical Support team.

Worldwide:	www.netiq.com/support/contactinfo.asp
North and South America:	1-713-418-5555
Europe, Middle East, and Africa:	+353 (0) 91-782 677
Email:	support@netiq.com
Web Site:	www.netiq.com/support

Contacting Documentation Support

Our goal is to provide documentation that meets your needs. The documentation for this product is available on the NetIQ Web site in HTML and PDF formats on a page that does not require you to log in. If you have suggestions for documentation improvements, click **Add Comment** at the bottom of any page in the HTML version of the documentation posted at www.netiq.com/documentation. You can also email Documentation-Feedback@netiq.com. We value your input and look forward to hearing from you.

Contacting the Online User Community

NetIQ Communities, the NetIQ online community, is a collaborative network connecting you to your peers and NetIQ experts. By providing more immediate information, useful links to helpful resources, and access to NetIQ experts, NetIQ Communities helps ensure you are mastering the knowledge you need to realize the full potential of IT investments upon which you rely. For more information, visit community.netiq.com.

1 Understanding the Multi-Domain Active Directory Driver

The NetIQ Identity Manager driver for Multi-Domain Active Directory supports provisioning of multiple domains in an Active Directory forest. This driver also supports cross domain object synchronization for a forest. The driver simplifies the overall deployment and integration of the entire Active Directory forest with the Identity Manager solution.

The existing Active Directory driver is actively in use and suffices most of the Identity Manager deployment scenarios. The Multi-Domain Active Directory driver enables your enterprise with multiple domain support.

The driver provides the following key features:

- Supports object synchronization across domains for a forest
- Supports user provisioning and group membership through entitlements
- Provides bidirectional password synchronization
- Supports nested group membership synchronization
- Supports automatic Domain Controllers (DC) discovery for domains
- Supports DC failover
- Supports PowerShell Cmdlets and Exchange mailbox provisioning
- Provides scalability using separate messaging queuing system for each domain
- Supports Permission Collection and Reconciliation Service (PCRS)

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

- "Key Terms" on page 11
- "Data Transfers Between Systems" on page 13
- "Support for Standard Driver Features" on page 14
- "Default Driver Configuration" on page 18
- "Checklist for Enabling User Synchronization" on page 22

Key Terms

- "Identity Manager" on page 12
- "Connected System" on page 12
- "Identity Vault" on page 12
- "Identity Manager Engine" on page 12
- "Multi-Domain Active Directory Driver" on page 12

- "Driver Shim" on page 13
- ".NET Remote Loader" on page 13

Identity Manager

NetIQ 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 Identity Vault to store shared information, and uses Identity Manager engine for policy-based management of the information as it changes in the vault or connected system. Identity Manager runs on the server where Identity Vault and Identity Manager engine are located.

Connected System

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

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 the NetWare Core Protocol (NCP), 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.

Identity Manager Engine

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

Multi-Domain Active Directory Driver

The Multi-Domain Active Directory driver provisions and synchronizes objects and password across multiple domains in a forest. The driver addresses the need for configuring multiple driver instances to synchronize with multiple domains. The driver also supports PowerShell Cmdlets that eliminates the need of installing separate PowerShell and Exchange services.

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 Multi-Domain Active Directory driver is `DXMLMADDriver.dll`.

`DXMLMADDriver.dll` is implemented as a .NET Windows DLL file.

.NET Remote Loader

A Remote Loader enables a driver shim to execute outside of the Identity Manager engine, remotely on a different machine. The Remote Loader is a service that executes the driver shim and passes the information between the shim and the Identity Manager engine. The Multi-Domain Active Directory driver shim requires the .NET Remote Loader 64-bit version.

The .NET Remote Loader is an Identity Manager service that executes the driver shim and passes the information between the driver shim and the Identity Vault. Install the driver shim on the server running the .NET Remote Loader. This server should be a member server that belongs to a domain in the same forest.

You can create a secure communication between the .NET Remote Loader and the Identity Manager engine. For more information, see "Creating a Secure Connection to the Identity Manager Engine" in the *NetIQ Identity Manager Driver Administration Guide*.

When you use the .NET Remote Loader with the driver shim, two network connections exist:

- Between the Identity Manager engine and the .NET Remote Loader
- Between domain controller and the driver shim

Data Transfers Between Systems

The driver supports data transfer on both the channels, the Publisher and the Subscriber channels.

The Publisher channel controls the data transfer as follows:

- Reads events from the configured domains.
- Submits that information to the Identity Vault.

The Subscriber channel controls the data transfer as follows:

- Watches for the events from the Identity Vault objects.
- Makes changes to Active Directory based on the event data.

You can configure the driver filter so that both Active Directory and the Identity Vault are allowed to update attribute(s). In this configuration, the most recent change determines the attribute value, except for merge operations that are controlled by the filters and the merge authority.

Support for Standard Driver Features

The sections below contain information about the key driver features.

- "Supported Operations" on page 14
- "Remote Platforms" on page 14
- "Multi-Domain Support" on page 15
- "PowerShell Command Support" on page 15
- "Entitlements and Permission Collection and Reconciliation Service" on page 15
- "Automatic Domain Controller Discovery and Failover" on page 17
- "Domain Controller Failover" on page 18
- "Password Synchronization Support" on page 18
- "Data Synchronization Support" on page 18
- "Nested Group Synchronization Support" on page 18
- "Scalability" on page 18
- "Multiple Active Directory User Account Support" on page 18

Supported Operations

The Multi-Domain Active Directory driver performs the following operations on the Publisher and Subscriber channels:

- **Publisher Channel:** Add, Modify, Delete, Migrate, Move, Query operations and password synchronization.
- **Subscriber Channel:** Add, Modify, Delete, Migrate, and Query operations, Password Set/Reset operations only on User objects, execution of PowerShell Cmdlets using policies, and Move operation across domains within the same forest.

Remote Platforms

The Multi-Domain Active Directory driver is a .NET Remote Loader only driver. The driver uses the Remote Loader service to run on a Windows domain controller or Windows member server. The Remote Loader service for the Multi-Domain Active Directory driver can be installed on the following Windows platforms:

- Windows Server 2022 (64-bit)
- Windows Server 2019 (64-bit)
- Windows Server 2016 (64-bit)
- Windows Server 2012 R2 (64-bit)
- Windows Server 2012 (64-bit)

For more information about remote installations, see "Deploying the Multi-Domain Active Directory Driver" on page 24.

For additional information about system requirements, see "Planning Your Installation " in the *NetIQ Identity Manager Setup Guide for Windows*.

Multi-Domain Support

The Multi-Domain Active directory driver provisions and synchronizes objects across all domains in a forest. The driver simplifies integrating Microsoft Active Directory with Identity Manager.

PowerShell Command Support

The Multi-Domain Active Directory driver establishes remote PowerShell sessions with the preferred DC for each domain. PowerShell commands are executed for a domain provided by the associations of the object or the LDAP DN of the domain in the XDS document. You can use Identity Manager policies, stylesheets, and ECMA to add the commands. The driver shim executes the commands in the respective domains.

The driver establishes a remote PowerShell session with the preferred domain controller for each domain. If an account is used across multiple domains, then the number of sessions will not exceed the number of configured domains. If Exchange is used, a second remote session is established with that Exchange server to execute exchange commands. This second PowerShell session with Exchange is created within the first PowerShell session and does not count toward the total number of PowerShell sessions being used by the server.

Remove-psession is a cleanup call and it is done by the driver at the time of shutdown.

Entitlements and Permission Collection and Reconciliation Service

Entitlements

The Multi-Domain Active Directory driver supports entitlements. By default, it supports UserAccount, Group, and ExchangeMailbox.

When using entitlements, an action such as provisioning an account in the target 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. 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.

Permission Collection and Reconciliation Service (PCRS)

The Multi-Domain Active Directory driver also supports Permission Collection and Reconciliation Service (PCRS) that allows you to create and manage the relationship of identities with resource assignments. PCRS helps you to create custom entitlements to map with the resources in the Identity Vault. You can dynamically create resources with custom entitlements holding permission values from Active Directory and permission assignments between Identity Manager resource/entitlement model and Active Directory.

The driver uses PCRS to map entitlements with resources and automatically assign those entitlements to users when permissions change in Active Directory. The driver content includes an enhanced entitlement package that supports the following entitlements:

- **ExchangeMailbox:** This entitlement grants or denies a Microsoft Exchange mailbox for the specified user.
- **Group:** This entitlement grants or denies membership to a group in Active Directory. When the entitlement is revoked, Identity Manager removes the user from the group.
- **UserAccount:** This entitlement grants or denies an Active Directory account for the specified user. When this entitlement is granted, the Multi-Domain Active Directory driver provides an enabled logon account. When this entitlement is revoked, the driver either disables or deletes the logon account, depending on the driver configuration.

The driver performs the following actions when PCRS is enabled:

- Reconcile resource assignments between the connected systems and the Identity Vault
- Provide a way to create customized entitlements and resources specific to your domain

The driver supports multiple domain connections, which can have multiple dynamic resources. The Role Based Provisioning Module (RBPM) considers each of the domains configured in the connection objects as separate Logical Identifier (LLID) systems. Each logical system requires a unique dynamic resource. RBPM creates a dynamic resource for each LLID and thereby for each domain that the driver is connecting to. The entitlement value source for this dynamic resource is bound to the LLID.

The `PermissionOnboarding` job is a standard Identity Manager job and is available in the entitlement package. During the driver startup, the `PermissionOnboarding` job runs and queries Active Directory for resource updates. When the driver performs resource onboarding, this process creates the `EntitlementLLIDMapping` mapping table. The `PermissionOnboarding` job populates this table with the ResourceDNs of the dynamic resources created for LLIDs for each custom entitlement. This mapping table includes the mapping between an entitlement, its LLIDs, and the ResourceDN.

In the User Application, the **Resource Name** field in the Roles and Resources tab displays all the default entitlements and the custom entitlements. You can select the desired resource in this tab. The Details page under the Entitlements tab shows the dynamic value of the resource. The LLID is automatically mapped to the resource when the resource is created.

When creating custom entitlements, the driver can still use the CSV file to map the Active Directory entitlements with the corresponding resources in the Resource Catalog. After you create, deploy, and start the driver, the driver automatically reads the `PermissionNameToFile` mapping table. The CSV file information that the driver requires to create the custom entitlement is available in the `PermissionNameToFile` mapping table. The driver consumes the entitlements values from the CSV file and creates the custom entitlements.

RBPM creates new resources with the entitlement values from the CSV file and displays the new custom entitlement and the corresponding resource object in the Resource Catalog. When the permission assignments change in Active Directory, the driver policies consume the modified permission values and update the Resource Catalog.

NOTE: If you are not using group onboarding, ensure that you do not include the static group resources in the `StaticEntitlementValueMap` table.

If an administrator assigns a resource to a user in the User Application, then that change reflects in Active Directory, and similarly, if an Active Directory administrator makes a change to the user permission, the corresponding resource is updated with the permission assignment.

You can turn this functionality on or off using the Entitlement GCVs included with the driver.

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 *NetIQ Identity Manager Entitlements Guide*.

Prerequisites

Before continuing, ensure that you go through the prerequisites needed for enabling this functionality. For general prerequisites, see "Prerequisites " in *Synchronizing Permission Changes from the Connected Systems* in the *NetIQ Identity Manager Driver Administration Guide*.

Also, you need to set up administrative user accounts and configure a password policy for them. For more information, see "Setting Up Administrative User Accounts" and "Setting Up Administrative Passwords" in "Synchronizing Permission Changes from the Connected Systems " in the *NetIQ Identity Manager Driver Administration Guide*.

To use the new functionality included in the Multi-Domain Active Directory driver, you can create a new driver with the latest packages. For more information about creating a driver, see "Creating the Driver in Designer" on page 39.

CSV File Format

The Multi-Domain Active Directory driver can consume the entitlement information from the CSV file, which is present on the server where Identity Manager is installed. The CSV file must contain values of the Active Directory system permission information in the format specified below. The Active Directory administrator should maintain a separate CSV file for every custom entitlement.

For example, a CSV file can contain details about issuing parking passes to the employees for the **ParkingPass** entitlement. A CSV file that holds **ParkingPass** entitlement details represents this information in the following format:

```
North, North Lot, North Parking Lot
```

where **North** is the entitlement value, **North Lot** is the display name in the User Application for the entitlement value **North**, and **North Parking Lot** is the description of the entitlement value, which is displayed in the User Application. Optionally, you can also add an additional field for LLID name that allows you to configure the entitlement value for a specific domain.

Automatic Domain Controller Discovery and Failover

The Multi-Domain Active Directory driver supports automatic Domain Controller (DC) discovery during the driver start up. The driver either automatically discovers the nearest DC or connects with the preferred DC as per the configurations.

Domain Controller Failover

The Multi-Domain Active Directory driver supports automatic domain controller failover. If the driver is running and the connection with preferred domain controller fails, driver rests for the wait period specified before it tries to re-establish the connection with a secondary DCs.

Password Synchronization Support

The Multi-Domain Active Directory driver synchronizes passwords on both Subscriber channel and Publisher channel. For more information, see Chapter 6, “Synchronizing Passwords,” on page 57.

Data Synchronization Support

The Multi-Domain Active Directory driver synchronizes User objects, Group objects, containers, and Exchange mailboxes in the default configuration and can be customized to use additional classes and attributes.

Nested Group Synchronization Support

The Multi-Domain Active Directory driver synchronizes group memberships across domains when a group is added as the member of another group.

Scalability

The Multi-Domain Active Directory driver creates separate messaging queues for each of the synchronized domains. These message queues are processed simultaneously by the driver shim and changes are synchronized with the respective domains parallel.

Multiple Active Directory User Account Support

The Multi-Domain Active Directory driver does not support creation of multiple user accounts for the same eDirectory object on multiple domains in the same forest. A single driver instance can maintain only a one-one mapping between an eDirectory user object and any Active Directory domain account. If you require management of multiple Active Directory forests then you must have one Multi-Domain Active Directory driver per forest.

Default Driver Configuration

The Multi-Domain Active Directory driver is shipped with default packages. When the driver is created in Designer, a set of policies and rules are created suitable for synchronizing with Active Directory in a sample environment. If your requirements for the driver are different from the default policies, you need to modify the default policies to meet your business requirements.

- “User Object Name Mapping” on page 19
- “Data Flow” on page 19

User Object Name Mapping

Management utilities for the Identity Vault, such as Identity Console and Designer, 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.

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 Multi-Domain Active Directory driver.

- "Filters" on page 19
- "Schema Mapping" on page 19
- "Name Mapping Policies" on page 21
- "Active Directory Logon Name Policies" on page 22

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," on page 19
- Table 1-2, "Mapped Group Attributes," on page 20
- Table 1-3, "Mapped Organizational Unit Attributes," on page 20
- Table 1-4, "Mapped Organization Attributes," on page 20
- Table 1-5, "Mapped Locality Class," on page 20
- Table 1-6, "Mapped Non-Class Specific Attributes," on page 20

Table 1-1 *Mapped User Attributes*

eDirectory - User	Active Directory - user
DirXML-MDADAliasName	userPrincipalName
sAMAccountName	CN
L	PhysicalDeliveryOfficeName
Physical Delivery Office Name	l
nspmDistributionPassword	nspmDistributionPassword

Table 1-2 Mapped Group Attributes

eDirectory - Group	Active Directory - group
DirXML-MDADAliasName	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 Identity Console 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
Description	description
DirXML-EntitlementRef	DirXML-EntitlementRef
DirXML-EntitlementResult	DirXML-EntitlementResult
Facsimile Telephone Number	facsimiletelephoneNumber
Full name	displayName
Given Name	givenName
Group Membership	memberOf
Initials	initials

eDirectory	Active Directory
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 Multi-Domain Active Directory driver packages 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 Identity Console, the object name and the user logon name are the same.

If you create some users in Active Directory by using MMC, and then create other objects in the Identity Vault or another connected system that is synchronized with the Identity Vault, the object can look odd in the opposite console and might fail to be created in the opposite system. However, you can use the name mapping policies to avoid this problem.

The Full Name Mapping policy is used to manage objects in Active Directory by using the MMC conventions. When this policy 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. However, this name might be inconvenient for login to the Vault.

Using the Name Mapping policies is controlled through Global Configuration Values. For information, see "Global Configuration Values" on page 105.

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. The 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 the 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 looking at 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 "Global Configuration Values" on page 105.

Checklist for Enabling User Synchronization

Use the following checklist to verify that you complete the following tasks in order to have a complete solution with the driver.

- Ensure that you have installed the software mentioned in "Driver Prerequisites" on page 23.
- Install the driver object. For more information, see Chapter 3, "Installing the Driver Files," on page 37.
- Create and configure the driver object. For more information, see Chapter 4, "Creating a New Driver," on page 39.

2 Preparing Multi-Domain Active Directory

Use the information in this section as you prepare to install the Multi-Domain Active Directory driver:

- "Driver Prerequisites" on page 23
- "Deploying the Multi-Domain Active Directory Driver" on page 24
- "Securing Driver Communication" on page 26
- "Creating an Administrative Account" on page 29
- "Configuring System Permissions" on page 30
- "Windows Message Queuing Permissions" on page 31
- "Becoming Familiar with Driver Features" on page 31

Driver Prerequisites

The Multi-Domain Active Directory must be deployed only on a Windows Server (64-bit). You can deploy the driver on a server that supports .NET Remote Loader 64-bit version and Active Directory driver configuration.

Ensure that you have completed the following prerequisites before installing the Multi-Domain Active Directory driver:

Table 2-1 *Prerequisites*

Prerequisites for Identity Manager Engine Server
◆ Identity Manager 4.6 or later ◆ Identity Manager Designer 4.6 or later ◆ Managed System Gateway Driver 4.0.2.0 or later This driver version contains the necessary software for data collection to work with the Multi-Domain Active Directory driver. You must download it from the Downloads (https://download.novell.com/index.jsp) Web site.

Prerequisites for Multi-Domain Active Directory Driver

- Windows Server 2022, Windows Server 2019, Windows Server 2016 (64-bit), Windows Server 2012 R2 (64-bit) or later, and Windows Server 2012 (64-bit).
- Microsoft .NET Framework 4.5 or later
- Identity Manager 4.6 (64-bit) .NET Remote Loader or later.

  NOTE: Identity Manager 4.7 .NET Remote Loader requires Microsoft .NET Framework 4.6 at a minimum. The .NET Framework is required to be additionally installed on Microsoft Windows Server 2022, Windows Server 2019, Windows Server 2016, and 2012 R2 (64-bit). It is automatically installed on Microsoft Windows Server 2016.
- Windows Message Queuing feature.
- Microsoft Windows Management Framework 4.0.

  NOTE: The Management Framework installs Microsoft Windows PowerShell version 4.0 on the Windows server.

NOTE: The domain functional level for the Active Directory driver is supported on all supported Windows Server platforms.

The server hosting the Multi-Domain Active Directory driver must be a member of an Active Directory domain in the target forest.

If you want to synchronize Exchange accounts, see Chapter 8, “Provisioning Exchange Mailboxes,” on page 75.

Deploying the Multi-Domain Active Directory Driver

The Multi-Domain Active Directory driver shim must run on one of the supported Windows platforms. You can install the Multi-Domain Active Directory driver on either the domain controller or a member server. NetIQ recommends that you install the driver on the Windows Member Server to benefit the driver failover capability. If the driver is installed on a domain controller, failover for the hosted domain is not supported.

You can only run the Multi-Domain Active Directory driver either as an application or a service.

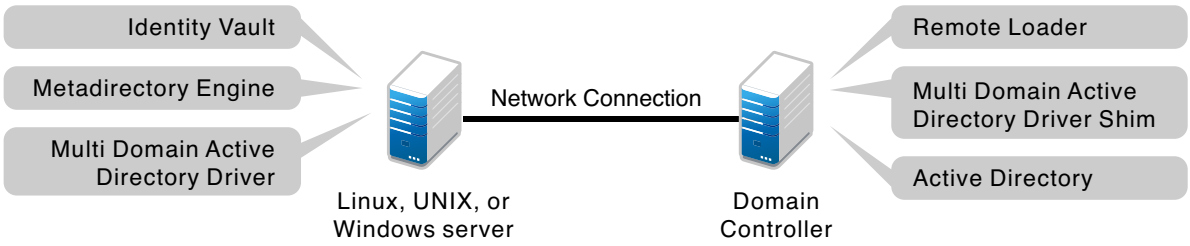
Before you start the driver installation, determine where you want to install the driver.

- “Remote Installation on Windows and Other Platforms” on page 24
- “Remote Installation on a Windows Member Server” on page 25

Remote Installation on Windows and Other Platforms

In a remote installation, you can install the .NET Remote Loader and the driver shim on the Active Directory domain controller and the Identity Vault and the Identity Manager engine on a separate server.

Figure 2-1 Remote Loader and Driver on the Domain Controller



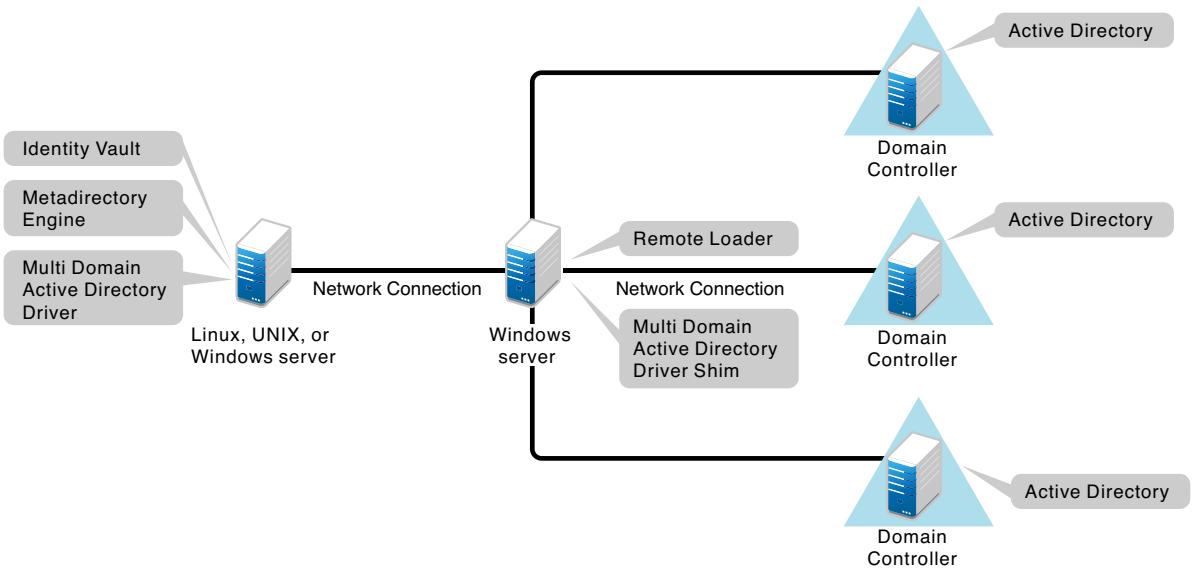
This configuration is attractive if the Identity Vault and the Identity Manager engine are installed on a platform other than one of the supported versions of Windows.

Both types of remote installations eliminate the performance impact of hosting the Identity Vault and the Identity Manager engine on the domain controller.

Remote Installation on a Windows Member Server

NetIQ recommends that you use a three-server configuration. This configuration ensures the driver failover capability for the Multi-Domain Active Directory driver.

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



In this figure, the two Windows servers are member servers of the domain.

Securing Driver Communication

The major security recommendations are in the areas of authentication, encryption, and use of the Remote Loader.

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 you implement your driver, pay close attention to security.

- "Authentication Methods" on page 26
- "Encryption Using SSL" on page 26

Authentication Methods

Authentication identifies the driver shim to Active Directory and, potentially, the local computer.

Table 2-2 *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 forest.	The server hosting the driver must be a member of the forest.
Simple	Used when the server hosting the driver shim is not a member of the forest.	The driver can be installed on a server that is not a member of the forest.	Some provisioning services are unavailable, such as Exchange mailbox provisioning, password synchronization, moving users across domains, and auto discovery of domain controllers.

NOTE: Multi Domain Active Directory driver uses **Negotiate** as the default authentication method. When the Multi Domain Active Directory driver's basic configuration file is imported to create a new driver, the authentication method is set to **Negotiate** by default. If you want to use **Simple** authentication, change the authentication setting on the driver's property page after the driver is created.

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 you are accessing Active Directory remotely by using an Multi-Domain Active Directory driver shim on a member server, you need to set up SSL between the Multi-Domain Active Directory driver shim and Active Directory. You do this by setting the SSL parameter to **Yes** on the driver configuration. See Step 5, in "SSL Connection between the Multi-Domain Active Directory Driver and the Domain Controller" on page 27.

If you are using the Remote Loader on the domain controller, you can set up SSL between the Identity Manager engine and the Remote Loader. For additional information on SSL and Remote Loaders, see "Creating a Secure Connection to the Identity Manager Engine" in the *NetIQ Identity Manager Driver Administration Guide*.

SSL Connection between the Multi-Domain Active Directory Driver and the Domain Controller

To establish SSL connections to an Active Directory server, you must configure SSL. This involves setting up a CA, then creating, exporting, and importing the necessary certificates. This is only needed if the Remote Loader is running on a member server. However, if the driver is running on a domain controller, no additional configuration is required after you enable driver SSL.

- "Setting Up a Certificate Authority" on page 27
- "Creating, Exporting, and Importing Certificates" on page 27
- "Verifying the Certificate" on page 28

Setting Up a Certificate Authority

Most organizations already have a CA. 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 CA that the issuing CA of this certificate chains to.

If you do not have a CA in your organization, you must establish one. NetIQ, Microsoft, and several other third parties provide the tools necessary to do this. Establishing a CA is beyond the scope of this guide. For more information about the NetIQ solution, see Certificate Server Administration in the *eDirectory Administration Guide*.

Creating, Exporting, and Importing Certificates

After you have a CA, 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 the 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 certification authority" (http://support.microsoft.com/kb/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 CA that issued the certificate.

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

   For more information on establishing trust for certificates, see "Policies to establish trust of root certification authorities" (http://technet.microsoft.com/en-us/library/cc775613(v=ws.10).aspx), in the Microsoft TechNet Library (http://technet.microsoft.com/library/bb625087.aspx).
5. In Identity Console, edit the driver properties and in the **Driver Settings** tab, change the **Use SSL for LDAP connection between Driver Shim and AD (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 Multi-Domain 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`

For example, you would use the following command if your server is configured for port 636.

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

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 .NET Remote Loader and Identity Manager

When you enable SSL between Engine and .NET Remote Loader, ensure that the certificate is added to the certificate store before starting the driver for the first time.

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

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.

The Multi-Domain Active Directory driver allows you to set up an administrative account in two ways:

- Using Single Administrative Account for the Forest
- Using Individual Administrative Accounts for Each Domain

When setting up a single administrative account for the forest, ensure that the account is a member of `Enterprise Admins` group. An administrator can create proxy account with minimum permission to operate the driver. Administrator can create account for the entire forest or individual accounts for all configured domains.

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 be a member of the `Administrators` group and 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.

Permissions for Remote PowerShell Execution

To establish a remote PowerShell session, ensure that the following prerequisites are met:

- Enable the remote PowerShell (if not enabled) by executing the `Winrm quickconfig` on the configured domain controllers.
- The user should have sufficient permission to run the remote PowerShell. To provide the required permission, execute the following command in the Powershell framework:

  ```
  Cmd = Set-PSSessionConfiguration -name Microsoft.PowerShell -
  ShowSecurityDescriptorUI
  ```

NOTE: This process needs to be followed for each domain controller as these PowerShell settings do not replicate.

Permissions for Exchange Provisioning

To provision Exchange mailboxes, your Identity Manager account must have "Act as part of the Operating System" permission for the logon account and must be a member of the "`Organizational Management`" group.

Permissions for Inter-domain Moves

To enable an object move between two domains, the domain administrative account configured in the driver for the source domain must also be a member of `Domain Admins` group of the destination domain.

Configuring System Permissions

In order to retrieve a user's password on the Publisher channel, the driver requires system permissions in addition to Active Directory permissions.

Identity Manager also configures specific permissions for its own internal components. On domain controllers, the PWFilter component runs using SYSTEM privileges, so the local system account should have full permissions to the `HKEY_LOCAL_MACHINE\SOFTWARE\Novell\PwFilter\Data` registry key, as well as any sub-keys.

The driver shim runs using SYSTEM privileges by default, so the system account should also have full permissions to the `HKEY_LOCAL_MACHINE\SOFTWARE\Novell\PassSync\Data` registry key, as well as any sub-keys. If the driver is run using any other account, that account should be given full permissions to the `HKEY_LOCAL_MACHINE\SOFTWARE\Novell\PwFilter\Data` registry key, as well as any sub-keys. The account should also be a member of the `Administrators` group.

NOTE: The driver automatically provides default permissions to both PWFilter and the driver shim. Modifying these permissions can affect the functionality of the driver and should be performed with caution.

Windows Message Queuing Permissions

When you start the driver for the first the time, the driver creates multiple instances of Windows Message Queues to store events for the respective domains. These queues are created using the credentials with which the Multi-Domain Active Directory driver is running.

In application mode, usually this is the credentials with which the user is logged in. You must ensure that all the subsequent driver startup must happen using this credentials or with other credentials having similar access rights to the Windows Message Queues.

In service mode, usually this is the system account of the Windows server. You must ensure that all the subsequent driver startup must happen using this credentials or with other credentials having similar access rights to the Windows Message Queues.

Switching between the application and service modes can result in state that will render the messaging queues to be inaccessible by the driver. In this scenario, you can start the driver using the **Run As** option with the credentials that has the access permissions to the queues.

Becoming Familiar with Driver Features

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

- "Schema Changes" on page 31
- "Structuring eDirectory Container Hierarchy" on page 31
- "Moving Cross Domain Objects" on page 32
- "Automatic Failover" on page 32
- "Multivalue Attributes" on page 33
- "Using Custom Boolean Attributes to Manage Account Settings" on page 33
- "Provisioning Exchange Mailboxes" on page 34
- "Expiring Accounts in Active Directory" on page 34
- "Driver Response Behavior" on page 35

Schema Changes

The Multi-Domain Active Directory driver introduces a path syntax multi-valued `DirXML-MDADContext` and `DirXML-MDADAliasName` attributes to support multiple Multi-Domain Active Directory drivers. The `DirXML-MDADContext` allows you to store the user's current AD context (LDAP DN). The `DirXML-MDADAliasName` allows you to store the current AD logon attribute (sAMAccountName or-- userPrincipalName).

Structuring eDirectory Container Hierarchy

The Multi-Domain Active Directory driver provides two ways of organizing Active Directory objects if spread across multiple domain in the forest. The two ways are as follows:

Mirrored

In the Mirrored mode, each domain is mapped to a separate container created in edirectory under the user container in `idv.dit.data.users`. This container stores all the directory objects belonging to that particular domain. An object must be created within this hierarchy for it to be synchronized to the Active Directory domain. The driver automatically matches the objects created in this container with the corresponding Active Directory container configured in the **Synchronization Settings** under the **Configuration** tab in the Driver Properties page. For more information, see "Synchronization Settings" on page 108. The path to this container must be updated in **Domain Container** in the Synchronization Settings page.

When using User Account entitlements, the user objects residing in **Domain Container** can only be assigned to that particular domain.

Flat

In the Flat mode, all domains are mapped to a single user container in `idv.dit.data.users`. This container stores the directory objects belonging to all domains. In this mode, policy customization is required to match the users to the corresponding Active Directory container configured in the **Synchronization Settings** under the **Configuration** tab in the Driver Properties page. For more information, see "Synchronization Settings" on page 108. The path to `idv.dit.data.users` must be updated in **Domain Container** in the Synchronization Settings page.

When using User Account entitlement, the user objects are automatically synchronized to the domain that is present in the assignment value.

Moving Cross Domain Objects

When Multi-Domain Active Directory driver is configured in the **Mirrored** mode, a move between eDirectory domain containers is interpreted as a move between the respective Active Directory domains.

When the driver does not mirror the forest domain structure, but instead uses the **Flat** mode to synchronize, then an user account entitlement reassignment from one domain value to another domain value is interpreted as a move between these two domains.

Automatic Failover

In case of a Domain Controller (DC) failure, the drivers does the automatic failover in the following ways:

1. The driver allows the DC a period of time equal to **Time to Wait** interval to come back online.
2. After the **Time to Wait** interval expires, the driver automatically tries to connect with another domain controller. The secondary server can be configured or allowed to automatically locate during driver startup.

Multivalue Attributes

When the Multi-Domain 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 multivalued 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.

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.

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-uACAccountDisable	0x0002	Read-write
dirxml-uACDontExpirePassword	0x10000	Read-write
dirxml-uACEncryptedTextPasswordAllowed	0x0080	Read-write
dirxml-uACHomedirRequired	0x0008	Read-write
dirxml-uACInterdomainTrustAccount	0x0800	Read-only
dirxml-uACNormalAccount	0x0200	Read-only
dirxml-uACPasswordCantChange	0x0040	Read-only
dirxml-uACScript	0x0001	Read-write
dirxml-uACPasswordNotRequired	0x0020	Read-write
dirxml-uACServerTrustAccount	0x2000	Read-only
dirxml-uACWorkstationTrustAccount	0x1000	Read-only
dirxml-uACLockout	0x0010	Read-write

For troubleshooting tips relating to the userAccountControl attribute, see "The Active Directory Account Is Disabled after a User Add on the Subscriber Channel" on page 92.

Provisioning Exchange Mailboxes

The Multi-Domain Active Directory driver can be configured to provision Exchange mailboxes as well as Active Directory accounts. The Multi-Domain Active Directory driver can provision Exchange Server 2013 and Exchange Server 2016, and Exchange Server 2019 mailboxes. For information on configuring the driver to provision the Exchange mailboxes, see Chapter 8, "Provisioning Exchange Mailboxes," on page 75.

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.

Driver Response Behavior

The engine returns a status message for all the operations submitted to the driver queue and restores the operation-data to this status message.

The Multi-Domain Active Directory driver receives two status document, one when the event is queued and the another when the queued event is processed. The first status contains a type attribute set to `driver-queue` while the second status element does not contain any type attributes. This helps to differentiate the processing of policies for operation-data between the first and second status documents.

3 Installing the Driver Files

The Multi-Domain Active Directory driver can only be run from the .NET Remote Loader (64-bit) installed on a supported Windows platform. For information about the supported platforms, see "Planning Your Installation " in the *NetIQ Identity Manager Setup Guide for Windows*.

The following section explains the installation process for Multi-Domain Active Directory driver:

- "Preparing for Driver Installation" on page 37
- "Installing the Driver Files" on page 37
- "Configuring the Multi-Domain Active Directory Driver" on page 37

Preparing for Driver Installation

Before installing the driver, ensure that you have installed all the prerequisites for the Windows server. For more information, see "Driver Prerequisites" on page 23.

Installing the Driver Files

The Identity Manager server installation process installs the .NET Remote Loader and the Multi-Domain Active Directory driver:

1. Run `install.exe` from the Identity Manager framework installer installation folder.
2. Accept the license agreement and click **Next**.
3. Select **NetIQ Identity Manager Connected System Server (.NET).**

   When you select the .NET Remote Loader, the installer automatically installs the Multi-Domain Active Directory driver.
4. Select the .NET Remote Loader installation directory and continue the installation.

Configuring the Multi-Domain Active Directory Driver

To configure a new driver shim instance:

1. Create the driver using the Designer tool. For more information, see Chapter 4, "Creating a New Driver," on page 39.
2. Use the .NET Remote Loader graphical interface to configure the Multi-Domain Active Directory Driver.

   2a Browse to the installation directory of the .NET Remote Loader and run `rlconsole.exe` to open the GUI console. The .NET Remote Loader GUI is similar to the traditional Remote Loader GUI.

   2b Click Add, then specify the Description, Driver (`DXMLMADDriver.dll`) and other parameters in the page that displays.

2c To configure the Multi-Domain Active Directory driver as an application, deselect the **Establish a Remote Loader service for this driver instance**.

NOTE: You can configure the Multi-Domain Active Directory driver as an application or as a service.

2d Click **OK**. A prompt displays asking you if you want to start the Remote Loader. You can start the driver now or later.

When the driver is started as an application, a trace window opens. It doesn't open if the driver is started as a service.

3 Set up password synchronization. For more information, see Chapter 6, "Synchronizing Passwords," on page 57.

4 If using Exchange, prepare your domains for mailbox provisioning. For more information, see Chapter 8, "Provisioning Exchange Mailboxes," on page 75.

4 Creating a New Driver

After the Multi-Domain 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 37), you can create the driver in the Identity Vault. You achieve this by installing the driver packages and then modifying the driver configuration to suit your environment. The following sections provide instructions to create a new driver.

- “Creating the Driver in Designer” on page 39
- “Activating the Driver” on page 49
- “Adding Packages to an Existing Driver” on page 50

Creating the Driver in Designer

An administrator can use the Designer to create the Multi-Domain Active Directory driver. You can select the required domains and configure the driver.

After you create and configure the driver, deploy it to the Identity Vault. After you deploy the driver, add the domain connection objects to the Subscriber options in the Driver Parameters page. You need to make the changes in the driver object in eDirectory using Designer compare or Deploy option. After performing these actions, you can start the driver.

NOTE: The Multi-Domain Active Directory driver is package enabled and you must install and deploy the driver using Designer. You can use Identity Console to view and edit the deployed configuration. However, NetIQ recommends that you use Designer tool to perform any changes.

- “Importing the Current Driver Packages” on page 40
- “Installing the Driver Packages” on page 40
- “Configuring Domain Connections for Multi-Domain Active Directory Driver” on page 45
- “Configuring the Driver” on page 47
- “Deploying the Driver” on page 48
- “Starting the Driver” on page 49

Importing the Current Driver Packages

Driver packages can be updated at any time and are stored in the Package Catalog. Packages are initially imported into the Package Catalog when you create a project, import a project, or convert a project. It is important to verify that you have the latest packages imported into the Package Catalog before you install the driver.

To verify you have the latest packages imported into the Package Catalog:

1. Open Designer.
2. In the toolbar click **Help** > **Check for Package Updates**.
3. Click **OK** if there are no package update

   or

   Click **OK** to import the package updates. If prompted to restart Designer, click **Yes** and save your project, then wait until Designer restarts.
4. Click **OK** to import the selected packages, then click **OK** in the successfully imported packages message.
5. After the current packages are imported, continue with "Installing the Driver Packages" on page 40.

Installing the Driver Packages

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

1. In Designer, open your project.
2. In the Modeler, right-click the driver set where you want to create the driver, then select **New >Driver**.

   NOTE: You can also drag and drop the Multi-Domain Active Directory driver icon from the Designer palette.

   The Driver Configuration Wizard displays.
3. Select **Multi-Domain Active Directory Base** from the list of base packages, then click **Next**.
4. Select the optional features to install for the Multi-Domain Active Directory driver. All options are selected by default. The options are:
   - **Default Configuration:** This package contains the default configuration information for the Active Directory driver. Always leave this option selected.
   - **Entitlements and Exchange Support:** This package contains configuration information for synchronizing Exchange Mailbox accounts and policies that enable account creation and auditing for the Active Directory driver. If you want account creation and auditing enabled, verify that this option is selected.

     For general information about entitlements, see the *NetIQ Identity Manager Entitlements Guide*.

     This package also contains policies for quick onboarding of custom entitlements and dynamic resource creation. This package contains GCVs to control the resource mapping. Select this package if you want to enable the Permission Collection and Reconciliation

Service (PCRS) feature for this driver. For more information, see "Synchronizing Permission Changes from the Connected Systems " in the *NetIQ Identity Manager Driver Administration Guide*.

- **Password Synchronization:** This packages contains the policies that enable the Active Directory driver to synchronize passwords. If you want to synchronize passwords, verify that this option is selected. For more information, see the *NetIQ Identity Manager Password Management Guide*.
- **Data Collection:** This package contains the policies that enable the driver to collect data for reports. If you are using Identity Reporting, verify that this option is selected. For more information, see the *Administrator Guide to NetIQ Identity Reporting*.
- **Account Tracking:** This package contains the policies that enable you to track accounts for reports. If you are using Identity Reporting, verify that this option is selected. For more information, see the *Administrator Guide to NetIQ Identity Reporting*.
- **Audit Entitlements:** This package contains the entitlement policies that help the users to access resources in connected systems. For general information about entitlements, see the *NetIQ Identity Manager Entitlements Guide*.
- **Managed Information System:** This package contains the policies that enable the driver to collect data for reports for all the managed systems. If you are using Identity Reporting, verify that this option is selected. For more information, see the *Administrator Guide to NetIQ Identity Reporting*.

5 Click **Next**.

6 (Conditional) If there are package dependencies for the packages you selected to install, you must install them to install the selected package. Click **OK** to install the package dependencies listed.

7 (Conditional) If more than one type of package dependency must be installed, you are presented with these packages separately. Continue to click **OK** to install any additional package dependencies.

8 (Conditional) If not already configured, fill in the following fields on the Common Settings Advanced Edition page, then click **Next**:

- **User Application Provisioning Services URL:** Specify the User Application Identity Manager Provisioning URL. For example, `http://localhost:port/IDMProv`.
- **User Application Provisioning Services Administrator:** Specify the DN of the User Application Administrator user. This user should have the rights for creating and assigning resources. For example, `CN=name.OU=unit.O=data`. For more information, see "Setting Up Administrative User Accounts" in the *NetIQ Identity Manager Driver Administration Guide*.
- **User Application Provisioning Service Account DN:** Specify the DN of the User Application Provisioning Service Account. This is required only if you use Access Review.

NOTE: This page is only displayed if you installed the Common Settings Advanced Edition package.

---

9 (Conditional) If not already configured, fill in the following fields on the Common Settings page, then click **Next**:

NOTE: The Common Settings page is only displayed if the Common Settings package is a dependency.

- **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 want a unique location for this driver, set the value for all drivers on this page. After the driver is created, change the value on the driver's Global Configuration Values page.

- **Group Container:** Select the Identity Vault container where Active Directory groups 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 want a unique location for this driver, set the value for all drivers on this page. After the driver is created, change the value on the driver's Global Configuration Values page.

10 On the Driver Information page, specify a name for the driver, then click **Next**.

11 On the Remote Loader page, fill in the following fields to configure the driver to connect using the Remote Loader, then click **Next**:

- **Connect to Remote Loader:** By default, the driver is configured to connect using the Remote Loader. Click **Next** to continue. Otherwise, fill in the remaining fields to configure the driver to connect by using the Remote Loader.
- **Host Name:** Specify the hostname or IP address of the server where the driver's Remote Loader service is running.
- **Port:** Specify the port number where the Remote Loader is installed and is running for this driver. The default port number is 8090.
- **KMO:** Specify the Key Name of the Key Material Object (KMO) that contains the keys and certificate the Remote Loader uses for an SSL connection. This parameter is only used when you use SSL for connections between the Remote Loader and the Identity Manager engine.
- **Other parameters:** Specify any other parameters required to connect to the Remote Loader. Any parameters specified must use a key-value pair format, as follows: `paraName1=paraValue1 paraName2=paraValue2`
- **Remote Password:** Specify the Remote Loader's password as defined on the Remote Loader. The Identity Manager server (or Remote Loader shim) requires this password to authenticate to the Remote Loader
- **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 Identity Manager server.

12 Click **Next**.

13 (Conditional) On the Entitlements Name to CSV File Mappings page, click the **Add Name to File Mapping** icon to populate the page with the entitlement configuration options.

Identity Manager uses the CSV file to maps Active Directory entitlements into corresponding resources in the Identity Manager catalog.

NOTE: This page is only displayed if you installed the Entitlements and Exchange Mailbox Support package.

The information that you specify in this page is used for creating the permission catalog. Fill in the following fields, then click **Next**:

- **Entitlement Name:** Specify a descriptive name for the entitlement to map it to the CSV file that contains the Active Directory entitlement details.

  Entitlement Name is the name of the entitlement. This parameter corresponds to the Entitlement Assignment Attribute in Active Directory. For example, you could define an entitlement called **ParkingPass**.

  This parameter is used to create a resource in the User Application.

- **Entitlement Assignment Attribute:** Specify a descriptive name for the assignment attribute for an entitlement.

  Entitlement Assignment Attribute holds the entitlement values in Active Directory. For example, you could have an attribute called **Parking**.

  You must add this parameter to **Field Names** in the Driver Parameters page or modify it in driver settings after creating the driver.

- **CSV File:** Specify the location of the CSV file. This file must be located on the same server as the driver. This file contains the values for the application entitlements.

- **Multi-valued?:** Set the value of this parameter to **True** if you want to assign resources and entitlements multiple times with different values to the same user. Otherwise, set it to **False**.

NOTE: After creating the driver, you can modify **Entitlement Name to CSV File Mapping** from **PermissionNameToFile** mapping.

---

14 On the Synchronization Settings page, click the icon and fill in the following fields to configure the driver's domain synchronization settings for each domain, then click **Next**:

- **Domain DNS Name:** Specify the DNS name of the Active Directory domain managed by this driver. For example, `domain.com`.

- **Domain Container:** Specify the container where user objects reside in the Identity Vault. By default, the container is `data\users`. If you want to mirror the target domain, you must first create a separate container within the hierarchy of your default `users` container and specify the container value in this field. For example, `data\users\container`. If you do not want to mirror the target domain to the Identity Vault, use the default `users` container. Only those users added into these containers are considered for synchronization with the Identity Vault.

- **Active Directory User Container:** Specify the container where user objects reside in Active Directory. For example, `CN=users,DC=domain,DC=company,DC=com`.

  NOTE: You can create custom containers also. An example for a custom container path is `OU=custom,DC=domain,DC=company,DC=com`. If you use a flat placement rule, this is the container where the users reside. If you use a mirrored placement rule, this is the root container.

- **Subscriber Channel Placement Type:** Select the desired form of placement for the Subscriber channel. This option determines the Subscriber channel Placement policies.
  - **mirrored:** Places objects hierarchically within the base container
  - **flat:** Places objects only in the base container

- **Publisher Channel Placement Type:** Select the desired form of placement for the Publisher channel. This option determines the Publisher channel Placement policies.
  - **mirrored:** Places object hierarchically within the base container
  - **flat:** Places objects only in the base container

15 (Conditional) On the General Information page, fill in the following fields to define your Active Directory system, then click **Next**:

- **Name:** Specify a descriptive name for this Active Directory system. The name is displayed in reports.
- **Description:** Specify a brief description for this Active Directory system. The description is displayed in reports.
- **Location:** Specify the physical location of this Active Directory system. The location is displayed in reports.
- **Vendor:** Leave Microsoft as the vendor of Active Directory. This information is displayed in reports.
- **Version:** Specify the version of this Active Directory system. The version is displayed in the reports.

NOTE: This page is only displayed if you installed the Managed System package.

---

16 (Conditional) On the System Ownership page, fill in the following fields to define the ownership of the Active Directory system, then click **Next**:

- **Business Owner:** Select a user object in the Identity Vault that is the business owner of the Active Directory system. This can only be a user object, not a role, group, or container.
- **Application Owner:** Select a user object in the Identity Vault that is the application owner of the Active Directory system. This can only be a user object, not a role, group, or container.

NOTE: This page is only displayed if you installed the Managed System package.

---

17 (Conditional) On the System Classification page, fill in the following fields to define the classification of the Active Directory system, then click **Next**:

- **Classification:** Select the classification of the Active Directory system. This information is displayed in the reports. The available options are:
  - Mission-Critical
  - Vital
  - Not-Critical
  - Other

    If you select **Other**, you must specify a custom classification for the Active Directory system.
- **Environment:** Select the type of environment the Active Directory system provides. The available options are:
  - Development
  - Test
  - Staging

- Production
- Other

  If you select **Other**, you must specify a custom classification for the Active Directory system.

NOTE: This page is only displayed if you installed the Managed System package.

---

18 Review the summary of tasks that will be completed to create the Multi-Domain Active Directory driver, then click **Finish**.

The driver is now created. You can modify the configuration settings, by continuing with the next section, "Configuring Domain Connections for Multi-Domain Active Directory Driver" on page 45 and "Configuring the Driver" on page 47. To deploy the driver, continue to "Deploying the Driver" on page 48.

Configuring Domain Connections for Multi-Domain Active Directory Driver

The Multi-Domain Active Directory driver require you to perform additional configuration in addition to the basic driver configuration. The NetIQ Identity Manager Designer provides Multi-Domain Active Directory driver editor for configuring these additional configurations. You can use the Multi-Domain Active Directory driver editor to accomplish the following tasks:

- Add forest and configure domain connections for the Multi-Domain Active Directory driver.
- Configure the driver with multiple domains within the same forest. The editor allows you to select the domains that you want to synchronize with Identity Manager.

IMPORTANT: ◆Root domains are mandatory for the configuring the driver. When adding the forest, the Designer tool automatically identifies the root domain.

- NetIQ recommends that you create the driver instance on a member server.
- You can configure only one instance of the driver for a forest on the machine where the driver shim is running.

- Configure a Primary Domain Controller (DC) and a list of alternate DCs for each domain or select **Auto discover** option to let the driver automatically identify the alternate domain controllers.

  In case of a primary DC failure, the driver tries to establish connection with an alternate DC.

Adding Forests to the Multi-Domain Active Directory Driver

You must first add the forests to configure the domain connections.

1 Open your project in Designer.

2 In the Modeler, right-click the Multi-Domain Active Directory driver icon and select **Multi-Domain Active Directory Configuration**.

  The Multi-Domain Active Directory Configuration Editor displays.

3 Click the icon to add a new forest.

4 In the Add Forest pop up window, fill in the following fields:

- **Forest Short Name:** Specify the forest name. Ensure that you specify a logical forest name that is accepted by the Identity Vault.
- **Global Catalog Server:** Specify the global catalog server address. You can specify the port number along with the IP address. For example, `IP Address:port`. The default port for clear text is 3268 and for SSL is 3269.
- **User:** Specify the username in LDAP format. For example, `CN=name,OU=employee,O=department`.
- **Password:** Specify the global catalog server password.
- **Secure Connection:** Select this option to establish a secure connection with the global catalog server.

5 Click **OK**.

NOTE: This creates a new forest and adds all the domains associated with the forest. By default, the root domain is added automatically. Designer displays multiple domains in the **Available Domains** list in the **Forest Configuration** tab.

---

6 Repeat step 4 through step 6 to create multiple forests for the Multi-Domain Active Directory driver.

Configuring the Domain Connections

After adding the forest, follow these steps to configure the domain connections for each forest.

1 In the **Forest Configuration** tab, select the desired domain from the **Available Domains** and move to the **Selected Domain** list.

The selected domains also display in the Forest tree view.

2 Select the domain from the Forest tree view and proceed with the domain configuration.

3 In the **Domain Configuration** tab, fill in the following fields:

- **Domain:** Displays the selected domain name.
- **User:** Specify the administrative username. Use LDAP format for **Simple** authentication.
- **Wait Period:** Specify the interval that you want the driver to wait before re-establishing the connection with the next available domain controller during domain discovery failover. If you do not specify the wait period, the default value is five minutes.

  NOTE: NetIQ recommends that you specify the failover wait period that is more than the minimum time required for replication between the Active Directory servers.

- **Domain Controllers:** Specify the domain controller configuration. The options are:
  - **Auto Discover:** The Multi-Domain Active Directory driver supports automatic DC discovery during runtime. Select this option to automatically discover the nearest DCs. This option should not be used with Simple authentication.
  - **Configure Manually:** Select this option to configure the preferred and secondary domain controllers. To configure manually, select the desired domain controllers from the **Available Dcs** and move them to the **Selected DCs** list. In this scenario, ensure that the preferred DC is available during the driver start up.

NOTE: Ensure that at least few of the domain controllers for the target domain are available during the driver start.

- **Exchange-MDB:** Select the desired exchange mailbox database (MDB) that you want to provision to users in this specific domain from the **Available Exchange-MDB** list and move them to the **Selected Exchange-MDB** list. You can specify more than one mailbox database.

  NOTE: Selecting more than one Exchange server allows the driver to failover Exchange operations if the primary Exchange server is offline.

- **Trace File:** Specify the trace file. All the driver traces for this domain will be logged in this specified file. If you leave this field blank, the driver trace will be logged in the default remote loader trace file.
- **Trace Level:** Specify the trace level.
- **Trace File Size:** Specify the size of the trace file.

4 For the changes to take effect, click **Save** on the Designer toolbar.

IMPORTANT: After configuring the connection objects, deploy the connection object with the driver to the Identity Vault. After deploying it, link the connection objects to the Subscriber options in the driver configuration page.

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.

You can configure the driver with entitlements or with entitlements disabled. Ensure that you configure the mandatory driver properties by using the Driver Configuration and Global Configuration Values (GCVs) settings.

To edit the driver properties:

1 Open your project.

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

3 Select **Driver Configuration** and configure the configuration properties. For more information see, "Driver Configuration" on page 99.

4 Click **GCVs > Entitlements** and review the following settings:

  NOTE: These settings are only displayed if you installed the Entitlements package.

  - **Use User Account Entitlement:** Ensure the value of this parameter is set to **true** to enable the driver to manage user account permissions using the User Account entitlement. By default, the value is set to **true**.

- **Use Group Entitlement:** Ensure the value of this parameter is set to **true** to enable the driver to manage group memberships using the Group entitlement. By default, the value is set to **true**.

IMPORTANT: ◆If the values for **Use User Account Entitlement** and **User Group Entitlement** parameter is set to **true**, user and group membership synchronization is managed using the entitlements configuration method. In the entitlements configuration method, when you assign resources to a user, the entitlements and the polices determine to which active directory domain the user must be synchronized to from the Identity Vault.

- If the values for **Use User Account Entitlement** and **User Group Entitlement** parameter is set to **false**, user and group membership synchronization is managed using the non-entitlement configuration method. The **Domain Container** you specify in the synchronization settings in the **Configuration** section helps to synchronize users from Identity Vault to the Active Directory domain by using the Domain container approach. Each user container in the Identity Vault is mapped with a unique active directory domain. You can customize this to meet your business requirement.

- **Exchange Mailbox Provisioning:** Ensure the value of this parameter is set to **Use Exchange Mailbox Entitlement** to enable the driver to provision Exchange mailboxes. By default, the value is set to **Use Exchange Mailbox Entitlement**.

5 Click **Apply**.

6 Modify any other settings as necessary. For more information about other settings, see Appendix B, “Driver Properties,” on page 99.

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 require customization. The default policies and rules are discussed in “Default Driver Configuration” on page 18.

7 Click **OK** when finished.

8 Continue with “Deploying the Driver” on page 48.

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 success 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 user account called `DriversUser`. 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**.

Starting the Driver

When a driver is created, it is stopped by default. Identity Manager is an event-driven system and will start caching events as soon as the driver is deployed. These cached events will be processed once the driver is started.

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 71.

Activating the Driver

The Identity Manager driver for Multi-Domain Active Directory is part of the Identity Manager Integration Module for Microsoft Enterprise. This integration module includes the following drivers:

- NetIQ Identity Manager Driver for SharePoint
- NetIQ Identity Manager Driver for Office 365
- NetIQ Identity Manager Driver for Multi-Domain Active Directory

This integration module requires a separate activation. After purchasing the integration module, you will receive activation details in your NetIQ Customer Center.

If you create a new Multi-Domain Active Directory driver in a driver set that already includes an activated driver from this integration module, the new driver inherits the activation from the driver set.

If you create the driver in a driver set that has not been previously activated with this integration module, the driver will run in the evaluation mode for 90 days. You must activate the driver with this integration module during the evaluation period; otherwise, the driver will be disabled.

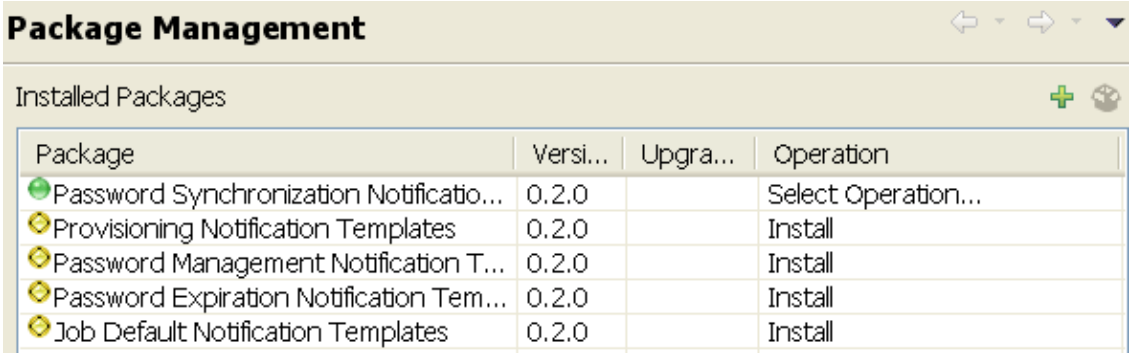
If driver activation has expired, `dstrace` displays an error message indicating that you need to reactivate the driver to use it. For information on activation, refer to in the Activating Identity Manager in the *NetIQ Identity Manager Overview and Planning Guide*.

Adding Packages to an Existing Driver

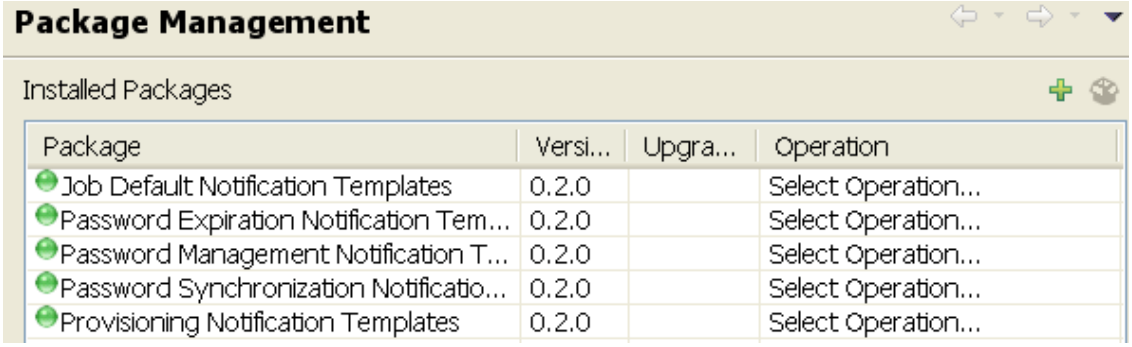
You can add new functionality to an existing driver by adding new packages to an existing driver.

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

   This option is only displayed on drivers. By default, only the packages that can be installed on the selected driver are displayed.
5. Click **Apply** to install all of the packages listed with the Install operation.



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



9 Repeat Step 1 through Step 8 for each driver where you want to add the new packages.

5 Upgrading an Existing Driver

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

Supported Upgrade Paths

You can upgrade from 5.0 version of the Multi-Domain Active Directory driver to 5.1.0 version.

What's New in Version 5.1.0

The driver supports Subscriber Service channel. This channel enables you to separately process the out-of-band queries without interrupting the normal flow of cached events. For example, the Subscriber Service channel can separately process code map refresh, data collection, and queries triggered from dxcmd. This helps to improve the performance of the driver. For more information, see Improving Driver Performance Using Subscriber Service Channel in the *NetIQ Identity Manager Driver Administration Guide*.

Upgrading the Driver

The driver upgrade process involves upgrading the installed driver packages and updating the driver files. The driver patch file contains the software to update the driver files.

- "Upgrading the Installed Packages" on page 53
- "Applying the Driver Patch" on page 54

Upgrading the Installed Packages

1. Download the latest available packages.

   To configure Designer to automatically read the package updates when a new version of a package is available, click **Windows > Preferences > NetIQ > Package Manager > Online Updates** in Designer. For more information, see the *NetIQ Designer for Identity Manager Administration Guide*.

2. Upgrade the installed packages.

   2a Open the project containing the driver.

   2b Right-click the driver for which you want to upgrade an installed package, then click **Driver > Properties**.

   2c Click **Packages**.

   If there is a newer version of a package, there is check mark displayed in the Upgrades column.

   2d Click **Select Operation** for the package that indicates there is an upgrade available.

2e From the drop-down list, click **Upgrade**.

2f Select the version that you want to upgrade to, then click **OK**.

NOTE: Designer lists all versions available for upgrade.

2g Click **Apply**.

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

Depending on which package you selected to upgrade, you must fill in the required information to upgrade the package.

2i Read the summary of the packages that will be installed, then click **Finish**.

2j Review the upgraded package, then click **OK** to close the Package Management page.

For detailed information, see the Upgrading Installed Packages in the *NetIQ Designer for Identity Manager Administration Guide*.

Applying the Driver Patch

The driver patch updates the driver files.

This section provides a general procedure for updating the driver files. For instructions about updating the driver to a specific version, search for the driver patch that you want to upgrade to in the Patch Finder Download Page and follow the instructions from the Readme file that accompanies the driver patch release.

Prerequisites

Before installing the patch, complete the following steps:

1. Take a back-up of the current driver configuration.
2. Stop the driver instance.
3. Stop the .NET Remote Loader service.
4. In a browser, navigate to the NetIQ Patch Finder Download Page.
5. Under **Patches**, click **Search Patches**.
6. Specify **Identity Manager *nn* Multi-Domain Active Directory driver nn** in the search box.
7. Download and unzip the contents of the patch file to a temporary location on your server.

Applying the Patch

1. Update the driver files:

   1a Navigate to the `<Extracted Driver Patch File Temporary Location>\windows` directory.

   1b Copy the following files to `<IdentityManager installation>\RemoteLoader.NET\lib`. folder.

   - `DXMLMADDDriver.dll`
   - `MADWrapper.dll`

1c Unblock the `.dll` files that you copied in Step 1.

Microsoft applies some security restrictions on the files copied to a Windows server which leaves the files in a blocked state. To unblock the files, right-click each file name and open the file with the Properties page and unblock the file.

2 Start the .NET Remote Loader service.

3 Start the driver instance.

6 Synchronizing Passwords

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

- "Securing Driver Connections" on page 57
- "Setting Up Password Synchronization Filters" on page 58
- "Retrying Synchronization after a Failure" on page 66
- "Disabling Password Synchronization on a Driver" on page 69
- "Diagnosing Password Synchronization Issues" on page 69

For information on troubleshooting password synchronization, see "Tips on Password Synchronization" on page 90.

Securing Driver Connections

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 remote loader runs on a domain controller:** The driver does not require connection security between the remote loader and Active Directory. The driver supports bi-directional password synchronization.
- **The remote loader runs on a member server:** The driver requires connection security between the remote loader and Active Directory, using either SSL or signing and sealing. The driver supports bi-directional password synchronization.
- **The remote loader runs on a server outside the forest:** Use the Simple authentication method to create the connection. The driver requires connection security using SSL between the remote loader and Multi Domain Active Directory. The driver supports password synchronization only on the Subscriber channel.

Configure the authentication method and enable SSL or signing and sealing in the driver parameters. For more information, see "Driver Parameters" on page 102.

Setting Up Password Synchronization Filters

The Multi-Domain Active Directory driver is configured to run on either a Windows member server or a domain controller. 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 the supported Windows APIs that encrypts the passwords, and then forwards the inputs to the driver. The driver eventually updates the Identity Vault with the password changes.

NOTE: ◆You do not need to install a password filter on a read-only domain controller.

- Password filter allows you to specify multiple hosts.

To simplify installation and administration of password filters, an Identity Manager PassSync utility is added to the Control Panel when the Remote Loader 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:

- "Allowing Remote Access to the Registry" on page 58
- "Not Allowing Remote Access to the Registry" on page 62

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 Multi-Domain Active Directory driver is configured to run.

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

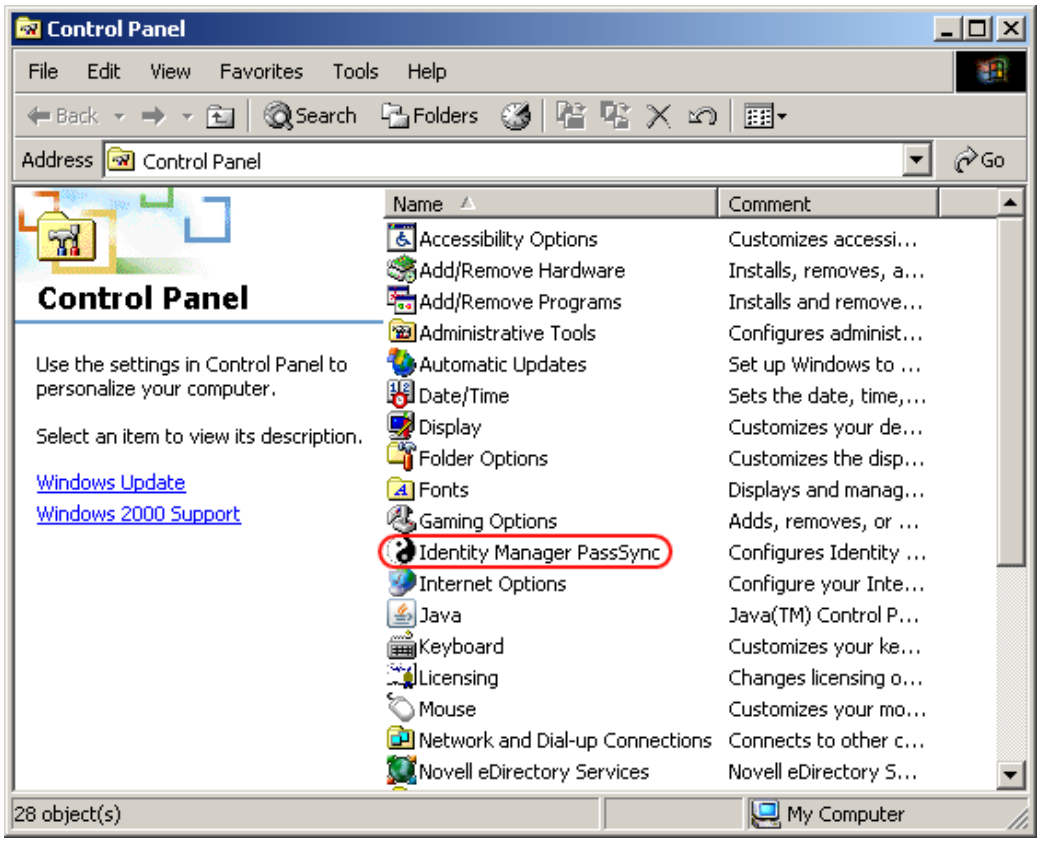
   - 137: NetBIOS name service
   - 138: NetBIOS datagram service
   - 139: NetBIOS session service

   A firewall could prevent the ports from being accessible remotely.

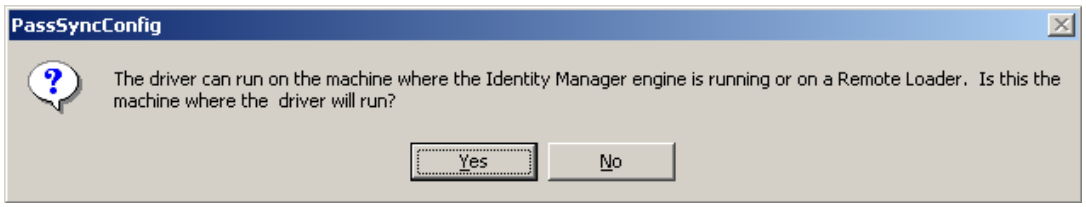
2 Log in with an administrator account on the computer where the driver is installed.

3 At the computer where the driver is installed, click **Start > Control Panel > Identity Manager PassSync.**

NOTE: Because there may be security policies in place that could block the PassSync utility from running, we recommend you run the utility using an account with Administrator privileges.



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

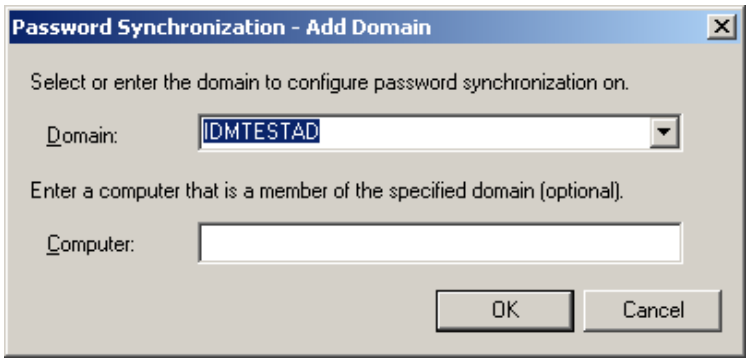


You only receive this prompt the first time you run the utility. After you complete the configuration, you are not shown this prompt again unless you remove this domain from the list.

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.

6 If no domains are listed, or if a 1208 error is displayed, you must manually type the domain name.



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.

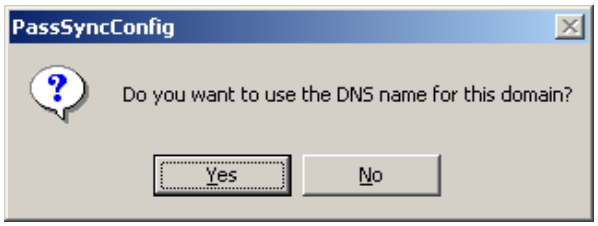
7 (Optional) Specify a computer in the domain, then click **OK**.

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.

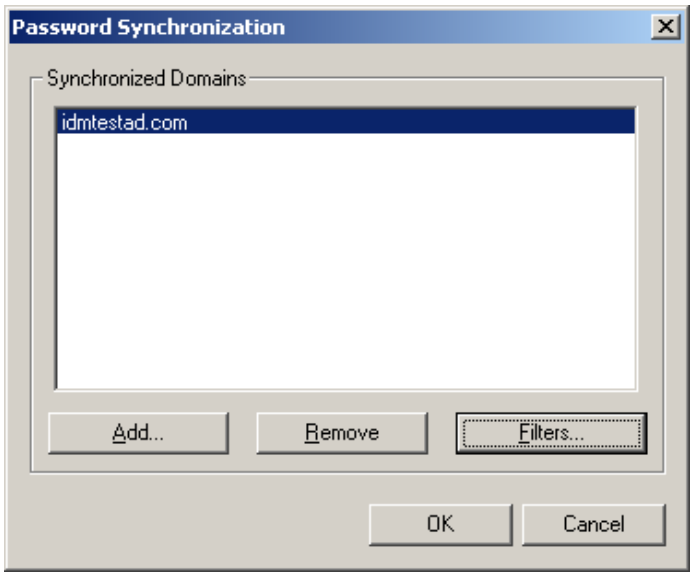
If you receive an error message indicating that PassSync can't locate a domain, specify a name.

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

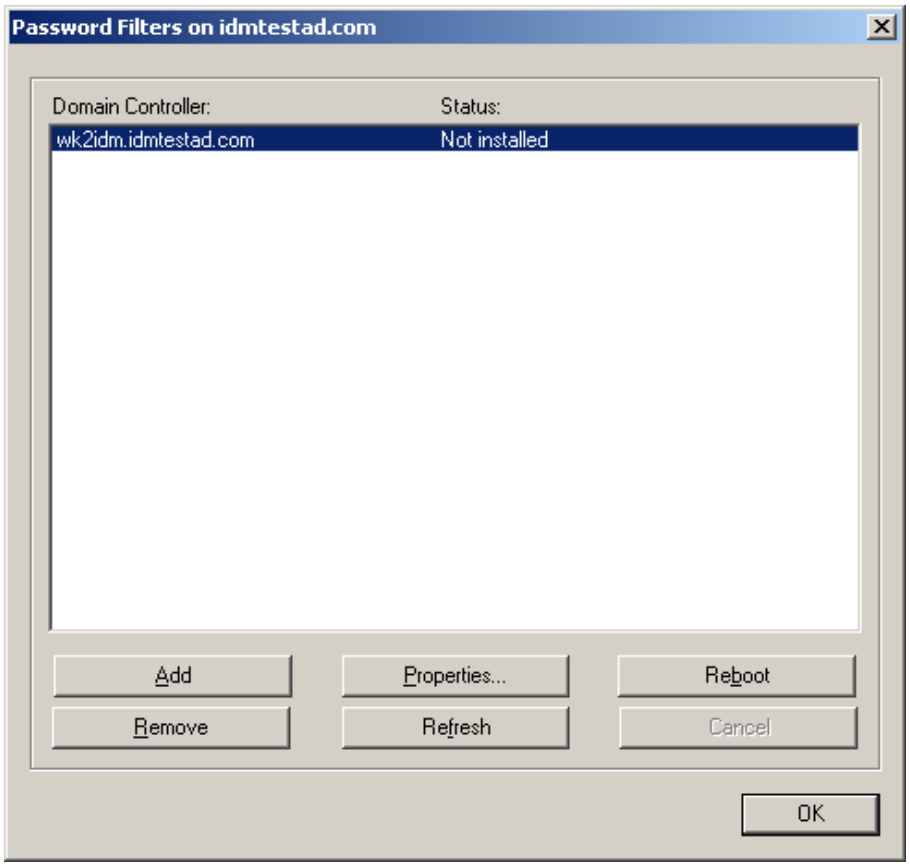


You can select **No**, but the DNS name provides more advanced authentication and the ability to more reliably discover domains in bigger installations. However, the choice depends on your environment.

9 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**.

10 To install the filter, click **Add**, then 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.

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

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

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 remote loader service 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 the procedure in this section, you install the Remote Loader 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 Multi-Domain Active Directory driver installed on it.

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

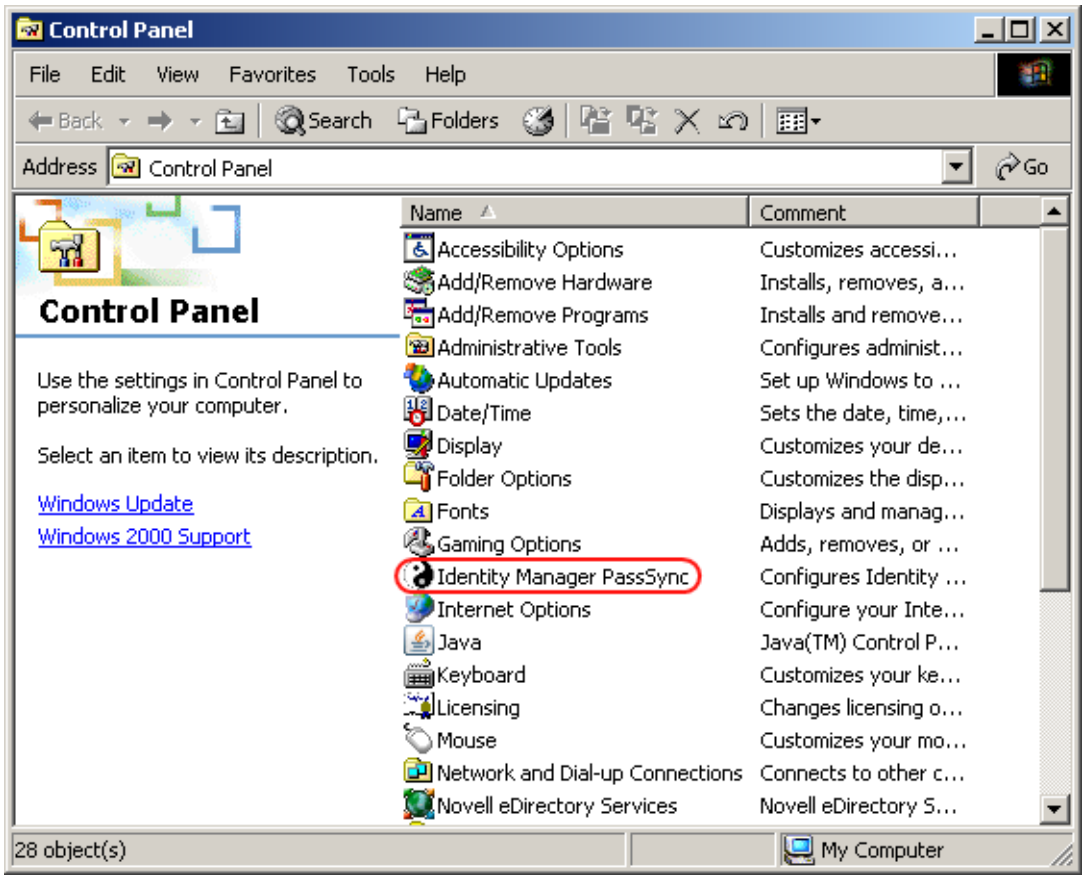
   A firewall could prevent the ports from being accessible remotely.

2. On the domain controller, install only the .NET Remote Loader (64-bit). For more information, see "Reviewing the Ports Used by the Identity Manager Components" *NetIQ Identity Manager Setup Guide for Windows*.

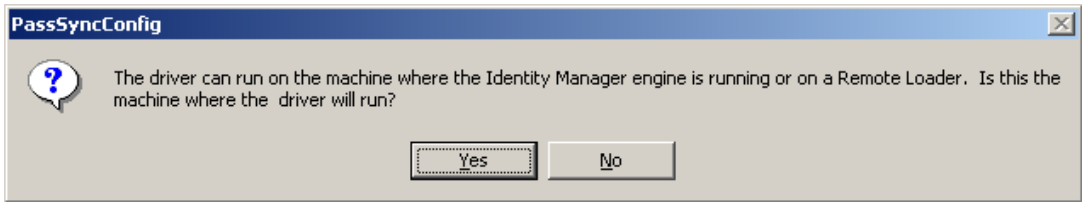
   Installing the driver installs the Identity Manager PassSync utility.

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

   NOTE: Because there may be security policies in place that could block the PassSync utility from running, we recommend you run the utility using an account with Administrator privileges.

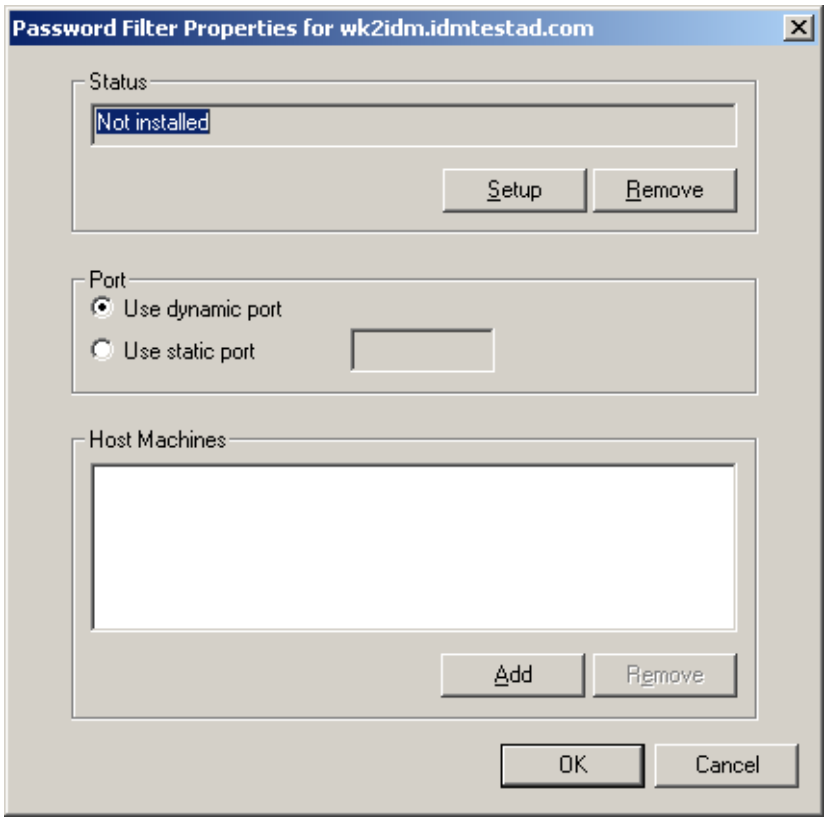


4 In the dialog box that displays, click **No** to specify that this machine is not running the Multi-Domain Active Directory driver.



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.

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.

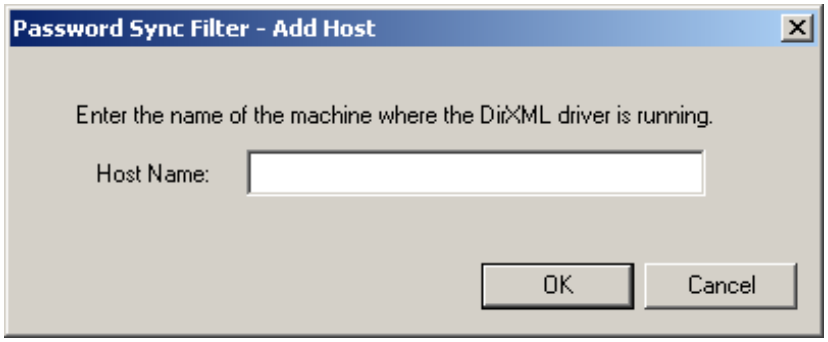


5 Click the **Setup** button to install the password filter, `pwFilter.dll`.

6 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.

7 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.

8 Verify that the information specified in Step 5 through Step 7 is correct, then click **OK**.

9 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.

10 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.

11 Repeat Step 2 through Step 10 for each domain controller that you want to participate in password synchronization.

12 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 by using the Windows Client. This should initiate the synchronization process.

Retrying Synchronization after a Failure

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

- "Retrying after an Add or Modify Event" on page 66
- "Password Expiration Time" on page 66

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.

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 five 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 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 67
- "Scenario: Increasing the Expiration Time" on page 67
- "Scenario: Never Meeting Requirements" on page 68
- "Scenario: E-Mail Notifications" on page 68

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 Multi-Domain 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 Multi-Domain 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. As per the configured UP policy, passwords that does not meet the policy requirement criteria will be reset to the Distribution password. 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.

Disabling Password Synchronization on a Driver

You can disable password synchronization on a driver by setting the **Password Sync Timeout** parameter to 0.

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 103 for more information about this parameter.

Diagnosing Password Synchronization Issues

You can use the PassSync Troubleshooting tool to diagnose issues that you encounter during password synchronization. This tool is a standalone executable that collects the following information to help you analyze synchronization issues:

- All Domain Controller information of the environment
- Password filter installation status and version
- RPC connection status
- Registry information created at `SOFTWARE\NOVELL\PWFILTER\DATA`

7 Managing Active Directory Groups and Exchange Mailboxes

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

- "Managing Groups" on page 71
- "Managing Microsoft Exchange Mailboxes" on page 72

Managing Groups

The Active Directory group class defines two types of groups and three types of 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 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.

The Multi-Domain Active Directory driver supports assigning permissions across multiple domains in the same forest.

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. 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
GROUP_TYPE_SECURITY_ENABLED	Security	0x80000000

Managing Microsoft Exchange Mailboxes

The Multi-Domain 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.

There are 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 that 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 resource that holds the permission 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.

NOTE: When configuring Exchange Server accounts, you can also set the value of the **Exchange HomeMDB** parameter to `defer`. With this value specified, the driver does not include the `-database` parameter when constructing the PowerShell command. This enables Exchange to load balance the databases in which email accounts are created.

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.

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 synchronization to assign the mailbox to the user in Active Directory.

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.

8 Provisioning Exchange Mailboxes

The Multi-Domain Active Directory driver can be configured to provision Active Directory accounts as well as Exchange accounts.

The driver can synchronize Exchange Server 2013, Exchange Server 2016 and Exchange Server 2019 accounts. The administrative account in the domain connection configuration need to have the necessary permissions and privileges to provision exchange mailboxes.

- "Setting Up Domains for Exchange Provisioning" on page 75
- "Setting Up Exchange Server Permissions" on page 76
- "Supported Operations on Exchange Mailboxes" on page 77

Setting Up Domains for Exchange Provisioning

The driver supports multiple domains. By default, while installing exchange some domains are prepared for exchange provisioning. If there are other domains that need to be prepared for exchange provisioning, you must prepare them before configuring the Multi-Domain Active Directory driver. This step creates additional containers and security groups, and sets permissions so that Exchange can access them.

Before setting up the domains for exchange provisioning, ensure that you set the required permissions for the user who provisions the domains.

To prepare all the domains in the forest, run the command:

```
Setup.exe /PrepareAllDomains /IAcceptExchangeServerLicenseTerms
```

To prepare only selected domains in the forest, run the command:

```
Setup.exe /PrepareDomain:<FQDN of the domain you want to prepare> /
IAcceptExchangeServerLicenseTerms
```

For more information about preparing Active Directory domains for exchange provisioning, see "Prepare Active Directory and domains" (https://technet.microsoft.com/en-us/library/bb125224(v=exchg.160).aspx#Step3).

Setting Up Exchange Server Permissions

The homeMDB attribute is set during initial configuration, but you can change the setting by modifying the driver policy.

To configure the driver to synchronize an Exchange Server account:

1. Verify that the authentication account for the domain has enough rights to create, delete, or move Exchange accounts.
2. Specify the configuration parameters to provision the Exchange mailboxes, when you are creating a driver object. See Table 8-1 for a list of Exchange parameters. See Chapter 4, "Creating a New Driver," on page 39 for information on how to create the driver object.

Table 8-1 *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 you are using entitlements, an external service such as the Workflow service or roles 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 you are 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.
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.

Supported Operations on Exchange Mailboxes

NOTE: Following Operations are not supported by Multi Domain Active Directory Driver:

- Creating Mailboxes in Exchange Online.
- Creating Users and Group in Azure AD.

For the above mentioned Operations you need to use "NetIQ Office 365 and Azure Active Directory Driver". For more information refer to: NetIQ Office 365 and Azure Active Directory Driver Implementation Guide. (https://www.netiq.com/documentation/identity-manager-48-drivers/msazure_ad/data/netiq-idm-driver-for-office365-and-msazure.html)

The Multi-Domain Active Directory driver provides support for the Exchange Server 2013, Exchange Server 2016 and Exchange Server 2019.

The Multi-Domain Active Directory driver creates, moves, and disables Exchange Server mailboxes. The cmdlets supported by the Active Directory driver to create, move, and disable mailboxes are `Enable-Mailbox`, `New-MoveRequest`, and `Disable-Mailbox`. The cmdlets use the following parameters in the Multi-Domain Active Directory driver:

- **Enable-Mailbox:** -Identity, -Alias, -Database, -DomainController
- **Disable-Mailbox:** Identity, -DomainController, -Confirm
- **New-MoveRequest:** -Identity, -TargetDatabase, -DomainController, -Confirm

For more information on PowerShell support in Identity Manager, see Chapter 9, "Configuring PowerShell Support," on page 79.

To provision Exchange Server mailboxes, you must complete the following steps:

- "Configuring the Driver" on page 77
- "Configuring the Driver to Support Database Load Balancing" on page 78
- "Support for Multiple Exchange Server in the Environment" on page 78

Configuring the Driver

You need to modify the existing driver object to enable mailbox provisioning.

Configuring Exchange Mailboxes in Designer

To configure exchange mailboxes, use the Multi-Domain Active Directory Configuration Editor in the Designer tool to set the value of the **homeMDB** parameter.

For more information, see "Configuring Domain Connections for Multi-Domain Active Directory Driver" on page 45.

Configuring the Driver to Support Database Load Balancing

The Multi-Domain Active Directory driver supports the database load balancing feature included in Exchange Server. You can use the driver to auto-provision exchange server accounts and enable Exchange to load balance accounts across the databases in your Exchange environment.

To enable load balancing, use either Designer or Identity Console to set the value of the **homeMDB** parameter.

For more information about load balancing in Exchange Server, see "Load Balancing" (http://technet.microsoft.com/en-us/library/jj898588(v=exchg.150).aspx).

Configuring Exchange Support for an Existing Driver in Designer

1. Right-click the Multi-Domain Active Directory driver in the Modeler, then select **Driver** > **Properties**.
2. Select **GCVs**.
3. Select the **Entitlements** tab.
4. Click **Exchange Mailbox Provisioning** and select **Use Policy**.
5. Set the value of the **Exchange homeMDB** parameter to `defer`.
6. Click **OK**.

Configuring Exchange Support for an Existing Driver in Identity Console

1. In Identity Console, click the **IDM Administration** tile.
2. On the Driver Dashboard, locate the driver, then click the driver icon to display the driver's properties page.
3. In the **Global Config Values** tab, click **Exchange Mailbox Provisioning** and select **Use Policy**.
4. Set the value of the **Exchange HomeMDB** parameter to `defer`.
5. Save the configuration.

Support for Multiple Exchange Server in the Environment

The Multi-Domain Active Directory supports Exchange Server 2013, Exchange Server 2016 and Exchange Server 2019. It also works in an environment where Exchange server 2013 and 2016 co-exist.

You can configure multiple MDB for each domain. Depending upon the entitlements provisioned for the user, mailboxes are created on the provisioned database.

9 Configuring PowerShell Support

Identity Manager provides support for managing Active Directory and Microsoft Exchange using Windows PowerShell cmdlets.

- "Overview of PowerShell Functionality" on page 79
- "System Requirements" on page 79
- "Implementing PowerShell Cmdlets in the Multi-Domain Active Directory Driver" on page 80

Overview of PowerShell Functionality

PowerShell is a shell-based automation framework created by Microsoft that allows users to manage the internal functions of other Microsoft products, including Active Directory and Exchange. PowerShell uses special .NET classes called cmdlets to perform various processing actions on objects in your Active Directory or Exchange environments. Identity Manager can use PowerShell cmdlets to perform post-processing on events by sending cmdlets to the Multi-Domain Active Directory driver using one or more policies.

For more information about PowerShell, see the following resources:

- "Getting Started with Windows PowerShell" (https://docs.microsoft.com/en-us/powershell/scripting/getting-started/getting-started-with-windows-powershell?view=powershell-6)
- "Windows PowerShell Owner's Manual" (http://technet.microsoft.com/library/ee221100.aspx)
- "A Task-Based Guide to Windows PowerShell Cmdlets" (http://technet.microsoft.com/en-us/scriptcenter/dd772285.aspx)

System Requirements

The system requirements to run PowerShell cmdlets is the same as the driver prerequisites. For more information, see "Driver Prerequisites" on page 23.

Identity Manager only supports using PowerShell cmdlets for Active Directory and Microsoft Exchange with Windows PowerShell 4.0 or later. However, Multi-Domain Active Directory driver supports out of the box Exchange Mailbox provisioning functionality using previous versions of PowerShell.

Implementing PowerShell Cmdlets in the Multi-Domain Active Directory Driver

To call cmdlets, create a rule which adds the PSExecute containing the PowerShell command string.

The Multi-Domain Active Directory driver looks for the PSExecute attribute in the input XDS code and reads any cmdlets embedded in a <value/> tag. The Active Directory server executes the commands sequentially using a programmatic PowerShell interface. For particular domains, these cmdlets are remotely executed on the Preferred DC configured for the domain.

NOTE: When including the PSExecute attribute in an Add or Modify event policy, you must adhere to the XDS format, or the driver ignores the embedded cmdlets.

Sample Active Directory Policy Rule with Cmdlets

The following is a sample rule created in an Multi-Domain Active Directory driver policy that allows an administrator to disable a newly-created user account in Active Directory using the Disable-ADAccount cmdlet.

```
<rule>
  <description>Adding PSExecute to Disable New User Account</description>
  <conditions>
   <and>
    <if-operation mode="regex" op="not-equal">query|status</if-operation>
   </and>
  </conditions>
  <actions>
   <do-set-local-variable name="identityname" scope="policy">
    <arg-string>
     <token-xpath expression='./add-attr[@attr-name="sAMAccountName"]/
value/text()'/>
    </arg-string>
   </do-set-local-variable>
   <do-set-dest-attr-value name="PSExecute">
    <arg-value type="string">
     <token-text xml:space="preserve">Disable-ADAccount -Identity </token-
text>
     <token-local-variable name="identityname"/>
    </arg-value>
   </do-set-dest-attr-value>
  </actions>
 </rule>
```

Note that the example rule above is used for including a PowerShell cmdlet in an Add event. You can also create rules to include PowerShell cmdlets in other types of events.

Specifically, the PSExecute attribute must include XDS code similar to the following example, where the <value> tag includes the PowerShell command string:

```
   <modify-attr attr-name="PSExecute">
    <add-value>
     <value type="string">New-ADUser -SamAccountName rbigby
-Name "Robin Bigby -AccountPassword (ConvertTo-SecureString -AsPlainText
"NetIQ1234" -Force) -Enabled $true -Path
'ou=active,ou=workforce,dc=w2012r2vm,dc=com'</value>
    </add-value>
   </modify-attr>
```

Available Active Directory and Exchange Cmdlets

PowerShell includes a wide variety of cmdlets and functions. The Multi-Domain Active Directory driver supports only Active Directory and Exchange PowerShell modules and cmdlets.

For information about using Active Directory PowerShell cmdlets, see "Active Directory Cmdlets in Windows PowerShell" (http://technet.microsoft.com/en-us/library/ee617195.aspx). For information about using Exchange cmdlets, see "Cmdlets" (http://technet.microsoft.com/en-us/library/aa996589.aspx).

NOTE: You can use the `PSExecute` attribute only with Active Directory and Exchange Cmdlets.

Creating Active Directory Policies with Cmdlets

To use cmdlets in Identity Manager, first use Designer to create a new policy in the Multi-Domain Active Directory driver. For more information about creating policies in Designer, see *Policies in Designer* (http://www.netiq.com/documentation/identity-manager-46/policy_designer/data/bookinfo.html) and *Understanding Policies for Identity Manager* (http://www.netiq.com/documentation/identity-manager-47/policy/data/bookinfo.html).

After you create a new policy, add a rule to the policy that includes an `add destination attribute value` action to create the `PSExecute` attribute, which calls one or more PowerShell cmdlets. You can include several cmdlet strings in multiple `value` tags for a single `PSExecute` attribute, as necessary.

To configure the rule using the Policy Builder, complete the following steps:

1. In Designer, right-click the policy in the Outline view and select **Edit**.
2. In the Policy Builder, select the location where you want to create the `PSExecute` attribute.
3. In the toolbar, click **Rule** and select **Action** > **Insert Action After**.
4. In the Do field under **Define new action below**, select **add destination attribute value**.
5. Specify `PSExecute` as the attribute name.
6. In the **Select mode** field, select **add to current operation**.
7. In the **Select object** field, select **Current object**.
8. In the **Specify value type** field, select **string**.
9. In the **Enter string** field, specify the PowerShell command string you want to use, enclosed in quotation marks.
10. Click **OK**.
11. Save the policy.

When specifying the PowerShell command string, you can include other variables configured in separate actions within the rule, as necessary.

For example, for the sample policy provided in "Sample Active Directory Policy Rule with Cmdlets" on page 80, you first add a rule to define the variable `identityname` as the name of the user account you want to disable using a PowerShell cmdlet, and then you specify the following string for the PSExecute variable, which uses the new `identityname` variable in the PowerShell command string:

```
"Disable-ADAccount -Identity"+Local Variable("identityname")
```

NOTE: You can also configure a policy to execute a specified cmdlet by modifying the XML directly, in the XML Source tab of the Policy Builder.

Verifying Active Directory Cmdlet Execution

When a PowerShell cmdlet runs successfully, the Active Directory returns a specific `success` event in the output XML, with the type `powershell`. After you run a cmdlet, check the output XML file for the following event:

```
<status level="success" event-id="rj-idmdt-
196#20150601120830#1#2:35663026-fba3-4d78-8ebd-26306635a3fb"
type="powershell"></status>
```

If the PowerShell cmdlet does not run successfully, the driver instead logs an `error` event in the output XML. The error event is similar to the following, including the reason for the failure:

```
<status level="error" event-id="rj-idmdt-196#20150218031220#1#1:cbe79a6d-
9f69-4a65-a1a1-6d9ae7cb699f" type="powershell"> Cannot
process command because of one or more missing mandatory parameters:
Identity. </status>
```

NOTE: If you execute multiple cmdlets in a single rule and one of the cmdlets does not run successfully, the driver does not execute any subsequent cmdlets in the rule and only logs the error event for the failed cmdlet. The driver does not log error events for the subsequent cmdlets, even though they did not run successfully, because the driver does not run those cmdlets after the failure occurs.

---

10 Security Best Practices

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

- "Security Considerations" on page 83
- "Default Configuration of the Security Parameters" on page 83
- "Recommended Security Configurations for the Simple Authentication Method" on page 85

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

Security Considerations

The following security considerations are implemented for Identity Manager Multi-Domain Active Directory driver:

- Driver synchronizes passwords from Active Directory (AD) domain controllers using encryption
- Data transfer between engine and Remote Loader must be performed using SSL
- Windows Cryptography API is used to encrypt Active Directory Passwords before synchronization
- Communication between driver shim and AD servers happens in a secure mode if the driver is running on a member server
- Driver uses `Microsoft System.Security SecureString` to encrypt User Passwords in memory
- Driver requires authorized AD accounts to read and make changes in AD

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 Identity Console:

1. In Identity Console, click the **IDM Administration** tile.
2. On the Driver Dashboard, locate the driver, then click the driver icon to display the driver's properties page.
3. Click **Configuration > Driver Parameters**.
4. Review the driver parameters in Table 10-1, 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 10-1, and decide if you need to make any changes.

Table 10-1 *Security Parameters*

Security Parameter	Description
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.
Digitally sign communications	This setting enables signing on a Kerberos or NTLM v2 authenticated connection between the driver shim and the Active Directory database. Signing ensures that a malicious computer is not intercepting data. 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. Select **Yes** to digitally sign the communication between the driver shim and Active Directory. Do not use this option with SSL. Select **No** if you do not want to sign communication between the driver shim and the Active Directory database.
Digitally sign and seal communications	This setting enables encryption on a Kerberos or NTLM v2 authenticated connection between the driver shim and the Active Directory database. Sealing encrypts the data so that it cannot be viewed by a network monitor. Sealing only works when you use the Negotiate authentication method and the underlying security provider selects NTLM v2 or Kerberos for its protocols. Select **Yes** to digitally encrypt communication between the driver shim and the Active Directory database. Do not use this option with SSL. Select **No** if you do not want to sign and seal communication between the driver shim and the Active Directory database.

Security Parameter	Description
Use SSL for LDAP connection between Driver Shim and AD	Select **Yes** to digitally encrypt communication between the driver shim and the Multi Domain Active Directory database. SSL requires that the Microsoft server running the driver shim imports the domain controller's server certificate. For more information, see "Microsoft Security Compliance Manager" (http://technet.microsoft.com/en-us/library/cc677002.aspx) for Windows Server 2012 R2 or later. 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 IDMPowerShell service and Password Set support. The driver performs a local logon. The authentication account must have the proper rights assignment. For more information, see "Creating an Administrative Account" on page 29. If **No** is selected, the driver performs a network logon only.
Encryption Password	Enter the encryption password for passing the encrypted messages in the Windows Messaging Queue.

Recommended Security Configurations for the Simple Authentication Method

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

Table 10-2 *Recommended Security Configuration for the Simple Authentication Method*

Parameter	Description and Recommended Setting
Authentication ID	Provide the authentication information while configuring the domain connections for the driver. Leave the field blank.
Authentication Context	Provide the authentication information while configuring the domain connections for the driver. Leave the field blank.
Password	Provide the authentication information while configuring the domain connections for the driver. Leave the field blank.
Digitally sign communications	Select **No**.
Digitally sign and seal communications	Select **No**.

Parameter	Description and Recommended Setting
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 "Microsoft Security Compliance Manager" (http://technet.microsoft.com/en-us/library/cc677002.aspx) for Windows Server 2012 R2 or later.

11 Troubleshooting

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

- "Data Protection from Unauthorized Access" on page 87
- "Error Displays When Stopping the Remote Loader Service" on page 87
- "Issues with User Account Reconciliation in Resource Catalog" on page 88
- "Changes Are Not Synchronizing from the Publisher or Subscriber" on page 88
- "Using Characters Outside the Valid NT Logon Names" on page 88
- "Synchronizing c, co, and countryCode Attributes" on page 88
- "Synchronizing Operational Attributes" on page 89
- "Password Complexity on Windows Server" on page 89
- "Tips on Password Synchronization" on page 90
- "Where to Set the SSL Parameter" on page 91
- "Password Filter Synchronization State Definitions" on page 91
- "The Active Directory Account Is Disabled after a User Add on the Subscriber Channel" on page 92
- "Restoring Active Directory" on page 93
- "Setting LDAP Server Search Constraints" on page 93
- "Error Messages" on page 94
- "Driver Binaries Fail to Load on Windows 2012 Devices" on page 95
- "Setting a Password in Multi-Domain Active Directory Driver Resets the eDirectory Password Expiration Date to the Current Date" on page 96
- "Troubleshooting Driver Processes" on page 96

Data Protection from Unauthorized Access

The driver mandates the user authentication by using authorized Active Directory accounts to make Active Directory updates. A SSL connection is supported between the Identity Manager engine and .NET Remote Loader and between the driver and the Active Directory server.

Error Displays When Stopping the Remote Loader Service

The driver returns an error when you try to stop the service using **Windows > Service** option in the .NET Remote Loader. There is no functionality loss and it is safe to ignore this message.

Issues with User Account Reconciliation in Resource Catalog

If PCRS is enabled and you try to add a user with incorrect data, the user provisioning to the Active Directory fails. However, the resource catalog displays the new user account. To workaround this issue, verify and correct the reasons for user add failure and try to synchronize the user again. If the issue persists, manually remove the assignment from the Catalog.

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 "Creating an Administrative Account" on page 29.

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

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

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.

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 11-1 *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.

Synchronizing Operational Attributes

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

For more information about operational attributes and attributes in general in Active Directory, see "Active Directory Operational Attributes" (http://msdn.microsoft.com/en-us/library/windows/desktop/aa772169%28v=vs.85%29.aspx) and "Attributes" (http://msdn.microsoft.com/en-us/library/windows/desktop/ms675089%28v=vs.85%29.aspx).

Password Complexity on Windows Server

Passwords must meet criteria that the password policies specify.

Complexities and requirements in Windows 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.

For information about the password complexity requirements for the supported Windows Server platforms, see "Password must meet complexity requirements" (http://technet.microsoft.com/en-us/library/hh994562%28v=ws.10%29.aspx).

For information about managing passwords in eDirectory, see the *Password Management Administration Guide* (https://www.netiq.com/documentation/password_management33/pwm_administration/data/bookinfo.html).

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 Server 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 server.

Tips on Password Synchronization

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

- The Identity Manager 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 Identity Manager 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 Identity Manager 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.

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 Multi-Domain Active Directory driver to provide the initial password for a user when the Multi-Domain 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 Multi-Domain 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 does not come with the Add event, but comes in a subsequent event. A user is added to eDirectory in two stages. The object is created in the initial Add event and then the password is set for this user. In the Create rule in the Subscriber channel, there is a suggested rule to veto if the `nspmDistributionPassword` operational attribute is not available. This causes the initial Add event to end with a veto, and the subsequent Modify event ends with only the `modify-attr attr-name="nspmDistributionPassword"` attribute, which turns the Modify event into a synthetic Add event. Because the initial Add event was vetoed, the password Modify event is converted into another Add event, but this time it can complete.

Where to Set the SSL Parameter

SSL is used for securing communication in two different ways:

- **For securing communication between the Remote Loader and the engine:** This is activated by specifying the string `kmo="<name of SSL Cert>"` in the Remote Loader connection parameters of the driver configuration. For more information, see "Creating a Secure Connection to the Identity Manager Engine" in the *NetIQ Identity Manager Driver Administration Guide*.
- **For securing communication between the driver shim and the domain controller:** If you select the **Use SSL** option, this setting is done in the driver configuration for securing communication between the Remote Loader and the domain controller when the driver shim is installed on a member server instead of a domain controller.

The SSL parameter in the driver configuration is for SSL connection between the Active Directory driver and Active Directory. It is not for SSL connection between the Identity Manager engine and the Remote Loader. See "Encryption Using SSL" on page 26.

Password Filter Synchronization State Definitions

The SyncState attribute provides information about the hosts to which passwords have been sent. Each bit in the SyncState value is set if the password has been successfully sent to the corresponding host in the Host Names list.

The SyncState value is located in the `HKEY_LOCAL_MACHINE\SOFTWARE\Novell\PwFilter\Data\` registry key.

Table 11-2 Password Filter Synchronization State Values

Host Name Value of Password Filter Key	Synchronization State
If there is one host name	◆ Sync State 00 - Password is not sent to any host.
If there are two host names	◆ Sync State 00 - Password is not sent to any host. ◆ Sync State 01 - Password is sent only to the first host. ◆ Sync State 02 - Password is sent only to the second host.
If there are three host names	◆ Sync State 00 - Password is not sent to any host. ◆ Sync State 01 - Password is sent only to the first host. ◆ Sync State 02 - Password is sent only to the second host. ◆ Sync State 03 - Password is sent only to the first and second hosts. ◆ Sync State 04 - Password is sent only to the third host. ◆ Sync State 05 - Password is sent only to the first and third hosts. ◆ Sync State 06 - Password is sent only to the second and third hosts.

You can see more than six synchronization states if there are four or more hosts in the Hosts Name list.

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.

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. Delete the `state` file from the .NET Remote Loader server.
4. Restore Active Directory.
5. Set the Active Directory driver to Manual or Automatic startup, then select the **Do not automatically synchronize the driver** option.
6. Start the driver.
7. Re-migrate to find unassociated objects.

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:\>
```

Error Messages

The following sections contains a list of common error messages.

- "LDAP_SERVER_DOWN" on page 94
- "LDAP_AUTH_UNKNOWN" on page 95
- "Error initializing connection to DirXML: SSL library initialization error: error:00000000:lib(0) :func(0) :reason(0)" on page 95

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 the authentication context, and you have disabled non-kerberos authentication to Active Directory. kerberos requires a DNS name for the authentication context.

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

Possible Cause: The driver initiates a failover.

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: Something is wrong with the certificate that was imported to the driver shim server, or no certificate was imported. Either import a certificate, or generate a new certificate and import it.

Action: Wait for the specified wait period time.

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 retry-ldap-auth-unknown driver parameter 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 hide="true" to hide="false".
4. Access the driver parameters. See "Driver Parameters" on page 102 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 server and a self-signed certificate in DER format, the connection fails. The certificate must have a Base64 format for the SSL connection to work.

Driver Binaries Fail to Load on Windows 2012 Devices

Right-click the following binary files, and then select **Properties** > **Unblock**.

- DXMLMADDriver.dll
- MADWrapper.dll
- ADDriver.dll
- log4net.dll

Setting a Password in Multi-Domain Active Directory Driver Resets the eDirectory Password Expiration Date to the Current Date

Whenever you set a password in Active Directory driver, the password syncs to Identity Manager, as expected. However, this also resets password expiration date in eDirectory to the current date and time. Because of this, a user with a future password expiration date in eDirectory now has an expired password.

To workaround this issue, perform the following steps:

1. Click the upper-right corner of the Active Directory driver, then click **Edit** properties.
2. Navigate to **Driver Properties > Global Configuration Values**.
3. In Password synchronization policy tab, ensure that you `deselect` the **If password does not comply, enforce Password Policy on the connected system by resetting user's password** option.

This ensures that the eDirectory password expiration date is not reset whenever you set passwords in Active Directory.

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 *NetIQ Identity Manager Driver Administration Guide*.

A Known Issues

The following known issue exists for this version of the driver:

- "Loss of Existing Resource Assignments During Upgrade" on page 97
- "Issue with Enabling Out of Band Sync for Attributes with Distinguished Name Syntax" on page 97
- "Applying the Latest Driver Package Does Not Change the Default Setting of Enable Service Channel ECV" on page 97
- "32-Bit pwfilter.dll File Is Not Available" on page 98

Loss of Existing Resource Assignments During Upgrade

Issue: The driver package upgrade process sets Add Logical System information to Entitlement Values to No. To view the permissions in a readable format, set this option to Yes. However, this change results in loss of all the existing resource assignments. This occurs due to the change in the entitlement values after code map refresh. (`Bug 1081672`)

Workaround: There is no workaround at this time.

Issue with Enabling Out of Band Sync for Attributes with Distinguished Name Syntax

If you enable Out of Band Sync for attributes with Distinguished Name Syntax (For example: Manager), there might be event loss in some cases. To avoid event loss, do not enable Out of Band Sync for such attributes.

Applying the Latest Driver Package Does Not Change the Default Setting of Enable Service Channel ECV

Issue: If you upgraded to Identity Manager 4.7 and updated the base packages for your driver, the package update process does not overwrite the default setting (`False`) of **Enable Service Channel** ECV.

This issue does not occur when you create a new driver.

Workaround: Manually change the ECV for the driver.

To change the ECV in Designer:

1. In Modeler, right-click the driver line.
2. Select **Properties > Engine Control Values**.
3. Click the tooltip icon to the right of **Engine Controls for Server**.

If a server is associated with the Identity Vault, and if you are authenticated, the engine control values display in the large pane.

4 Change the value for **Enable Subscriber Service Channel**.

5 Click **OK**.

6 For the change to take effect, deploy the driver to the live Identity Vault.

To change the ECV in Identity Console:

1 In Identity Console, click the **IDM Administration** tile.

2 On the Driver Dashboard, locate the driver, then click the driver icon to display the driver's properties page.

3 Click the **Configuration** tab.

4 Expand the **Engine Control Values** section.

5 Change the value for **Enable Subscriber Service Channel**.

6 Save.

7 For the change to take effect, restart the driver.

32-Bit pwfilter.dll File Is Not Available

You cannot install the password filter on a 32-bit Domain Controller.

There is no workaround at this time.

B Driver Properties

This section provides information about the Driver Configuration and Global Configuration Values properties for the Multi-Domain 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 *NetIQ Identity Manager Driver Administration Guide* for information about the common properties.

The information is presented from Identity Console's perspective. If a field is different in Designer, it is marked with a Designer icon.

- "Driver Configuration" on page 99
- "Global Configuration Values" on page 105

Driver Configuration

In Identity Console:

1. Click the **IDM Administration** tile.
2. On the Driver Dashboard, locate the driver, then click the driver icon to display the driver's properties page.

In Designer:

1. Open a project in the Modeler.
2. Right-click the driver line, then click **Properties**.
3. Click **Driver Configuration**.

The following sections describe driver configuration in details:

- "Driver Module" on page 100
- "Driver Object Password" on page 100
- "Authentication" on page 100
- "Startup Option" on page 101
- "Driver Parameters" on page 102
- "Subscriber Settings" on page 104
- "Publisher Settings" on page 105
- "ECMAScript (Designer Only)" on page 105
- "Global Configurations (Designer Only)" on page 105

Driver Module

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

The Multi-Domain Active Directory driver `dll` is: `DXMLMMADDriver.dll`.

Connect to Remote Loader: Used when the driver is connecting remotely to the connected system. The options are:

- **Java:** Specify the name of the java class.
- **Native:** Specify the name of the DLL file.
- **Connect to remote Loader:** Select this option to specify the remote loader client configuration.

  Designer includes one sub-option:

  - **Remote Loader client configuration for documentation:** Includes information on the Remote Loader client configuration when Designer generates documentation for the driver.

Driver Object Password

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

Authentication

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

Authentication ID: Provide the authentication information while configuring the domain connections for the driver. Leave the field blank.

Connection Information (Designer only): Specify the IP address or name of the server the application shim should communicate with. If you are synchronizing Exchange mail boxes, you must specify the full qualified name of the domain controller. For example: `myserver.company.com`.

Authentication context: Provide the authentication information while configuring the domain connections for the driver. Leave the field blank.

Remote Loader Connection Parameters: Used only if the driver is connecting to the application through the Remote Loader.

In Identity Console, enter `hostname=xxx.xxx.xxx.xxx port=xxxx secureprotocol=TLS version enforceSuiteB=`*`true/false`* `kmo=certificatename.`

- `hostname` specifies the IP address of the Remote Loader server.
- `port` specifies the TCP/IP port on which the Remote Loader listens for connections from the remote interface shim. The default port for the Remote Loader is 8090.
- `secureprotocol` specifies the version of the TLS protocol that the Remote Loader uses to connect to the Identity Manager engine. Identity Manager supports TLSv1, TLS v1_1, and TLSv1_2 versions only.

- `enforceSuiteB` specifies whether the Remote Loader uses Suite B for communicating with the Identity Manager engine. To use Suite B, specify `enforceSuiteB=true.` The communication supports only TLS version 1.2 version. Communication is not established if the connection has non-Suite B authentication algorithms.
- The `kmo` entry is optional. Use it only when an SSL connection exists between the Remote Loader and the Identity Manager engine.

  For example: `hostname=10.0.0.1 port=8090 kmo=IDMCertificate`

Specify the additional parameters in the **Other parameters** field.

Driver Cache Limit (kilobytes): Specify the maximum event cache file size (in KB). If it is set to zero, the file size is unlimited. select **Unlimited** option to set the file size to unlimited in Designer.

Application Password: Use the **Set Password** option to set the application authentication password.

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

The kmo entry is optional. It is only used when there is an SSL connection between the Remote Loader and the Identity Manager engine. For example, `hostname=10.0.0.1 port=8090 kmo=IDMCertificate`. Specify the additional parameters in the **Other parameters** field.

Remote loader password: Use this option to update the remote loader password.

Startup Option

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

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

Manual: The driver does not start when the Identity Manager server is started. The driver must be started through Designer or Identity Console.

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

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

Driver Parameters

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

The driver setting parameters are divided into the following categories:

- "Authentication Options" on page 102
- "Access Options" on page 103
- "Advanced Options" on page 103

Authentication Options

Show authentication options: Enables you to see and change the authentication options for the driver. The options are **show** or **hide**. These parameters control how the Multi-Domain Active Directory driver authenticates to the Active Directory domain controller.

Authentication Method: The Multi-Domain Active Directory supports **Negotiate** authentication method. Negotiate uses Microsoft's security package to negotiate the logon type. Typically kerberos or NTLM is selected. **Simple** authentication 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. The communication is in clear text, but signing ensures that the communication is not tampered with enroute to the destination. It reduces the chance 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. You can use this option with the **Digitally sign and seal communications** option.

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. You can use this option with the **Digitally sign communications option**.

Use SSL for LDAP connection between Driver Shim and Active Directory: Select **Yes** to digitally encrypt communication between the driver shim and the Multi Domain Active Directory database.

This option can be used with the Negotiate or Simple authentication method. SSL requires that the Microsoft server running the driver shim imports the domain controller's server certificate. For more information, see "Microsoft Security Compliance Manager".

Logon and impersonate: Select **Yes** to log on and impersonate the driver authentication account for Identity Manager PowerShell service and Password Set support. The driver performs a local logon. The authentication account must have the proper rights assignment. For more information, see "Creating an Administrative Account" on page 29.

If **No** is selected, the driver performs a network logon only.

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.

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 you have domain controllers distributed across multiple subnets, then it is recommended to set the timeout value to a minimum of 15 minutes to allow Active Directory replication to complete.

If this value is set to 0, password synchronization is disabled for this driver.

If this value is set to -1, passwords never expire. It can reach a maximum value of 2147483647 minutes.

The default value is 5 minutes.

DC Passwords TimeToLive (minutes): Specify the time limit in minutes for the passwords to be stored in the Domain Controller registry.

This allows the passwords that are stored in the Domain Controller registry to time out if the password does not synchronize to the driver within the specified time.

If this value is set to -1, passwords will never be deleted from the registry.

The default value is -1.

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.

Advanced Options

Show advanced options: Select **show** to display the advanced configuration options for the driver.

Enable Deletion of protected objects in Windows AD Server: Select **Yes** to delete the protected objects that are created through MMC in Windows Active Directory Server. Select **No** for protecting these objects from accidental deletion.

Retry LDAP Auth unknown error: Ordinarily, the driver shim returns a fatal error when encountering an LDAP-AUTH_UNKNOWN error that causes the driver to shut down. If you want the driver to retry the LDAP bind request, select **Yes**.

Enable DirSync Incremental Values: The Publisher channel usually receives all the values of a multi-valued attribute. Enabling this option reports only the added or deleted values during the poll interval. This requires 2003 Forest functional mode or above. This option is hidden by default. It can be modified by selecting the **Edit XML** option in the Driver configuration tab.

Subscriber Settings

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

Domain Connections Options

Show domain connection options: Select **show** to display the show domain connections options for the driver. All configured domain connection details display in this section. Click ✚ to add a new instance of domain template data to add domain connection option.

Connection DN: Specify the name of the domain connection. Enter the credentials for accessing the Identity Vault.

Connection Password: Set the connection password.

Queue Encryption Password

Encryption Password: Specify the key to encrypt the events before saving the message queue.

Exchange Options

Show Exchange Management Options: Select **show** to display the Microsoft Exchange options. These parameters control whether the driver shim uses the Identity Manager PowerShell service 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.

Enable Exchange mailbox provisioning: Select **enable**d to provision Exchange Mailbox accounts.

- **Allow Exchange mailbox move:** Select **Yes** to enable the driver to intercept modifications to the Active Directory homeMDB attribute and call the Identity Manager PowerShell service 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:** Select **Yes** to enable the driver to intercept removals of the Active Directory homeMDB attribute and call the Identity Manager PowerShell service to delete the mailbox.

  Select **No** if you don't want to delete the mailbox account when the Active Directory account is deleted.

Publisher Settings

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

Heart Beat Interval: Specify the time period at which the heart beat document is issues by the driver shim.

Polling Interval: Specify the interval value at which the driver shim reports the changes on the Publisher channel.

ECMAScript (Designer Only)

Displays an ordered list of ECMAScript resource objects. The objects contain extension functions for the driver that Identity Manager loads when the driver starts. You can add additional ECMAScript objects, remove existing files, or change the order the objects are executed.

Global Configurations (Designer Only)

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

Global Configuration Values

Global configuration values (GCVs) 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 several predefined 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 Identity Console:

1. Click the **IDM Administration** tile.
2. On the Driver Dashboard, locate the driver, then click the driver icon to display the driver's properties page.
   - **2a** Select the **Configuration** tab.
   - **2b** Expand the **Global Config Values** section.

To add a GCV to the driver set:

1. On the Driver Dashboard, click the upper right corner of the driver set to display the Action menu.
2. Select **Driver Set Properties**.
3. On the **Driver Set Configuration** tab, expand the **Global Config Values** section.
4. Save the values.

To access the driver's GCVs in Designer:

1 Open a project in the Modeler.

2 Right-click the Multi-Domain Active Directory driver icon or line, then select **Properties > Global Configuration Values.**

   or

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

The global configuration values are organized as follows:

- "Managed System Information" on page 106
- "Password Synchronization" on page 107
- "Configuration" on page 108
- "Account Tracking" on page 109
- "Entitlements" on page 110

Managed System Information

These settings help Identity Reporting to generate the reports. The following are the sections in the **Managed System Information** tab:

- "General Information" on page 106
- "System Ownership" on page 106
- "System Classification" on page 107
- "Active Directory Domain Configuration" on page 107
- "Connection and Miscellaneous Information" on page 107

General Information

Name: Specify a descriptive name for this Active Directory system. This name is displayed in the reports.

Description: Specify a brief description of this Active Directory system. This description is displayed in the reports.

Location: Specify the physical location of this Active Directory system. This location is displayed in the reports.

Vendor: Select Microsoft as the vendor of the Active Directory system. This information is displayed in the reports.

Version: Specify the version of this Active Directory system. This version information is displayed in the reports.

System Ownership

Business Owner: Browse to and select the business owner in the Identity Vault for this Active Directory system. You must select a user object, not a role, group, or container.

Application Owner: Browse to and select the application owner in the Identity Vault for this Active Directory system. You must select a user object, not a role, group, or container.

System Classification

Classification: Select the classification of the Active Directory system. This information is displayed in the reports. The options are:

- Mission-Critical
- Vital
- Not-Critical
- Other

  If you select **Other**, you must specify a custom classification for the Active Directory system.

Environment: Select the type of environment the Active Directory system provides. The options are:

- Development
- Test
- Staging
- Production
- Other

  If you select **Other**, you must specify a custom classification for the Active Directory system.

Active Directory Domain Configuration

Logical Instances: Click the icon to add multiple logical instances of the managed system.

Connection and Miscellaneous Information

Connection and miscellaneous information: This options is always set to **hide**, so that you don't make changes to these options. These options are system options that are necessary for reporting to work. If you make any changes, reporting stops working.

Password Synchronization

These GCVs enable password synchronization between the Identity Vault and the Active Directory system.

In Designer, you must click the icon next to a GCV to edit it. This displays the Password Synchronization Options dialog box for a better view of the relationship between the different GCVs.

In Identity Console, to edit the Password management options, follow the steps given below:

1. Click the **IDM Administration** tile.
2. On the Driver Dashboard, locate the driver, then click the driver icon to display the driver's properties page.
3. Select the **Configuration** tab.

4 Expand the **Global Config Values** section.

5 Select the **Password Synchronization** tab.

For more information about how to use the Password Management GCVs, see "Configuring Password Flow" in the *NetIQ Identity Manager Password Management Guide*.

Connected System or Driver Name: Specify the name of the Active Directory system or the driver name. This value is used by the e-mail notification template to identify the source of the notification message.

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: If **True**, uses the password from the connected system to set the non-reversible NDS password in eDirectory.

Publish passwords to Distribution Password: If **True**, uses the password from the connected system to set the NMAS Distribution Password used 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.

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

Configuration

The following GCVs contain configuration information for the Active Directory driver. They are divided into the following categories:

- "Synchronization Settings" on page 108
- "Name Mapping Policy" on page 109

Synchronization Settings

Configure Domains: In the synchronization settings, click to configure domains that needs to be synchronized.

- **Domain DNS Name:** Specify the DNS name of the Active Directory domain managed by this driver. For example, multidomain.com.
- **Domain Container:** Specify the eDirectory container name where the synchronized user objects reside in the eDirectory.
- **Active Directory User Container:** Specify the container where user objects reside in the Active Directory. If you are using a flat Placement rule, this is the container where the users are placed. If you are using a mirrored Placement rule, this is the root container.

- **Subscriber Channel Placement Type:** Specify the type of placement for the Subscriber Channel. Select **Flat** to strictly place objects within the base container. Select **Mirrored** to hierarchically place objects within the base container. This is used to determine the Subscriber Channel Placement policies.
- **Publisher Channel Placement Type:** Specify the type of placement for the Publisher Channel. Select **Flat** to strictly place objects within the base container. Select **Mirrored** to hierarchically place objects within the base container. This is used to determine the Publisher Channel Placement policies.

Name Mapping Policy

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).

User 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 the 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 the 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:** This option is useful when you want to generate userPrincipalName from the user logon name plus a hard-coded string defined in the policy.
- **None:** This option is useful when you do not want to control userPrincipalName or when you want to implement your own policy.

Account Tracking

Account tracking is part of Identity Reporting. For more information, see the *Administrator Guide to NetIQ Identity Reporting*.

Enable account tracking: Set this to **True** to enable account tracking policies. Set it to **False** if you do not want to execute account tracking policies.

Object Class: Add the object class to track. Class names must be in the application namespace.

Identifiers: Add the account identifier attributes. Attribute names must be in the application namespace.

Status attribute: Specify the name of the attribute in the application namespace to represent the account status.

Status active value: Value of the status attribute that represents an active state.

Status inactive value: Value of the status attribute that represents an inactive state.

Subscription default status: Select the default status that the policies assume when an object is subscribed to the application and the status attribute is not set in the Identity Vault. The options are:

- Active
- Inactive
- Undefined
- Uninitialized

Publication default status: Select the default status that the policies assume when an object is published to the Identity Vault and the status attribute is not set in the application. The options are:

- Active
- Inactive
- Undefined
- Uninitialized

Entitlements

There are multiple sections in the **Entitlements** tab. Depending on which packages you installed, different options are enabled or displayed.

- "Entitlements and Exchange Configuration" on page 110
- "Data Collection" on page 111
- "Role Mapping" on page 112
- "Resource Mapping" on page 112
- "Entitlement Extensions" on page 112

Entitlements and Exchange Configuration

For more information about entitlements, see "Entitlements and Permission Collection and Reconciliation Service" on page 15.

Use User Account Entitlement: Entitlements act like an On/Off switch to control account access. Enable the driver for entitlements to create accounts, and remove/disable it when the account entitlement is granted to or revoked from users. If you select **True**, user accounts in Active Directory can be controlled by using entitlements.

- **Enable Login Disabled attribute sync:** Specify whether the driver syncs the changes made to the `Login Disabled` attribute in the Identity Vault even if the User Account entitlement is enabled.
- **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**.
- **Parameter Format:** Select the parameter format the entitlement agent must use. The options are **Identity Manager 4** or **Legacy**.

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.

- **Parameter Format:** Select the parameter format the entitlement agent must use. The options are **Identity Manager 4** or **Legacy**.

Exchange Mailbox Provisioning: Specify the method to provision exchange mailboxes.

- Select **Disable Exchange Provisioning** to disable the Exchange Provisioning.
- Select **Use Exchange Mailbox Enablement** to enable the driver to manage Exchange Mailboxes based on the driver's Exchange Mailbox Entitlement, in Active Directory.

  Parameter Format: Select the parameter format the entitlement agent must use. The options are **Identity Manager 4** or **Legacy**.
- Select **Use Policy** to enable the driver to manage Exchange Mailboxes based on the driver's policies, in Active Directory.

  Exchange HomeMDB: Specify the exchange mailbox to provision the objects.

Data Collection

Data collection enables the Identity Report Module to gather information to generate reports. For more information, see the *Administrator Guide to NetIQ Identity Reporting*.

Enable data collection: Select **Yes** to enable data collection for the driver through the Data Collection Service by the Managed System Gateway driver. If you are not going to run reports on data collected by this driver, select **No**.

Allow data collection from user accounts: Select **Yes** to allow data collection by the Data Collection Service through the Managed System Gateway driver for the user accounts.

Allow data collection from groups: Select **Yes** to allow data collection by the Data Collection Service through the Managed System Gateway driver for groups.

Allow data collection from Exchange mailboxes: Select **Yes** to allow data collection by the Data Collection Service through the Managed System Gateway driver for Exchange mailboxes.

NOTE: The APP_NAME attribute displays the configured domain connection name for the Multi-Domain Active Directory driver as the attribute value in the Data Collection Query response. However, for the Active Directory driver the domain names specified in the Managed System package configuration displays in the query response.

Role Mapping

Identity Applications allow you to map business roles with IT roles.

Enable role mapping: Select **Yes** to make this driver visible to Identity Applications.

Allow mapping of user accounts: Select **Yes** if you want to allow mapping of user accounts in Identity Applications. An account is required before a role, profile, or license can be granted through Identity Applications.

Allow mapping of groups: Select **Yes** if you want to allow mapping of groups in Identity Applications.

Allow mapping of Exchange mailboxes: Select **Yes** if you want allow mapping of Exchange mailboxes in Identity Applications.

Resource Mapping

Identity Applications allow you to map resources to users. For more information, see the *NetIQ Identity Manager - User's Guide to the Identity Applications*.

Enables resource mapping: Select **Yes** to make this driver visible to Identity Applications.

Allow mapping of user accounts: Select Yes if you want to allow mapping of user accounts in Identity Applications. An account is required before a role, profile, or license can be granted.

Allow mapping of groups: Select **Yes** if you want to allow mapping of groups in Identity Applications.

Allow mapping of Exchange mailboxes: Select **Yes** if you want to allow mapping of Exchange mailboxes in Identity Applications.

Entitlement Extensions

User account extensions: The content of this field is added below the entitlement elements in the EntitlementConfiguration resource object.

Group extensions: The content of this field is added below the entitlement element in the EntitlementConfiguration resource object.

Exchange mailbox extensions: The content of this field is added below the entitlement element in the EntitlementConfiguration resource object.

C Migrating Users Per Domain

The Multi-Domain Active Directory driver supports user migration from Active Directory domain to the Identity Vault. Migration is only possible at a domain level. You can migrate only one domain at a time. If you have multiple domains in a forest, you must migrate each domain separately. The driver does not support object migration at the forest level.

You can use the Identity Console to migrate the user objects.

To migrate user objects:

1. In Identity Console, click the **IDM Administration** tile.
2. On the Driver Dashboard, locate the driver, then click the driver icon to display the driver's properties page.
3. Click the **Data Transformation and Synchronization** tab.
4. Click the **Migrate into IDVault** tab.
5. Click **Edit migration criteria** and add the User as class.
6. Select **Show all attributes from all classes** and select CN as attribute, then select the users that you want to migrate.

   To migrate all users, specify `*` for the CN attribute. For example, `CN=*`.
7. Select **MDAD instance** and provide the domain DN in LDAP format from where you want to migrate the user objects to Identity Vault.

   Example of domain DN in LDAP format: `ou=LKG, dc=PreKG, and dc=local`.
8. Click **Migrate into IDVault.**

D Trace Levels

The Multi-Domain Active Directory driver supports the following trace levels:

Table D-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, plus template instantiations

NOTE: If the Multi-Domain Active Directory driver uses the Remote Loader, the driver logs only driver shim trace messages on the Remote Loader, while the Identity Manager server logs the engine trace messages.

E Microsoft Windows Events

The Multi-Domain Active Directory driver logs the following events to the Password Synchronization event log in Microsoft Windows environments:

Table E-1 *Logged Password Synchronization Events*

Event Description	Type	Explanation
The password filter has been fully initialized. Domain Name = *DomainName*, Computer Name = *ComputerName*, Host Name = *HostName*	Information	Identity Manager successfully initialized the PassSync utility.
The password filter could not initialize its registry values.	Error	Identity Manager could not open the registry key `/HKLM/SOFTWARE/NOVELL/PWFILTER`.
The password synchronization notification for user *UserName* failed	Error	Pwfilter could not send a password notification to the PassSync utility for this user account.
The password for user *UserName* could not be changed.	Error	Identity Manager could not change the password for the specified user account.
The password filter RPC server failed to load.	Error	The PassSync remote procedure call (RPC) server could not initialize. Check whether RPC services are running on the server.
The password for user *UserName* in directory *DirectoryName* was not synchronized because the password change timed out.	Error	Identity Manager could not synchronize passwords for the specified user because the key exceeded its time to live as set in the driver.
The Cryptographic Service Provider has defaulted to *CSPProvider*. Encryption will be downgraded to the standards of this provider. Execution of the password synchronization server will not be affected. If higher encryption standards are required, please contact your network administrator.	Warning	Identity Manager has defaulted to using the base Cryptographic Service Provider (CSP) specified in the event description.
A request to allocate *RequestedSize* bytes of memory failed. Tag value = *TagValue*.	Error	Identity Manager could not allocate the requested memory.

Event Description	Type	Explanation
Driver NOT synchronizing passwords with the domain controller	Error	Identity Manager and the Multi-Domain Active Directory driver are not synchronizing passwords with this domain controller (DC). This may be due to pwfilter not being installed or being installed incorrectly.
Driver is synchronizing passwords with the domain controller	Information	Identity Manager and the Multi-Domain Active Directory driver are successfully synchronizing passwords with this DC