NetIQ® Identity Manager

Driver for Office 365 and Azure Active Directory Implementation Guide

November 2022

Legal Notice

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

About this Book and the Library

The *Identity Manager Driver for Office 365 and Azure Active Directory Implementation Guide* explains how to install and configure the Identity Manager Driver for Azure Active Directory.

Intended Audience

This book provides information for administrators implementing Identity Manager, application server developers, Web services administrators, and consultants.

Other Information in the Library

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

- Identity Manager Documentation Website (https://www.netiq.com/documentation/identity-manager-48/)
- Identity Manager Driver Documentation Website (https://www.netiq.com/documentation/identity-manager-48-drivers/)

Contents

About this Book and the Library 3

About NetIQ Corporation 9

About This Guide 11

1 Understanding the Office 365 and Azure Active Directory Driver 13

Understanding How the Driver Works 13
Data Transfers Between Systems 15
Driver Features 16
- Supported Operations 16
- Schema Extension 18
- License Handling 19
- Hybrid Mode 19
- Entitlements 20
- Group Cache Clear Interval 20

Planning to Install the Driver 22
- Options for Installing the Driver Shim 22

2 Installing Azure AD 5.1.7 23

Preparing for Installation 23
- Prerequisites 23
- Installing the Driver and the Identity Manager Exchange Service 31
- Securing Driver Communication 33

Creating a New Driver Object 38
- Creating the Driver Object in Designer 38
- Activating the Driver 47

Updating the Driver Files 47

3 Installing Azure AD 5.1.6 49

Preparing for Installation 49
- Prerequisites 49
- Installing the Driver and the Identity Manager Exchange Service 56
- Securing Driver Communication 58

Certificate Based Authentication support for Identity Manager Exchange Service 63
- Creating Client Certificate for IDM Exchange Service 63
- Verifying and Starting the Identity Manager Exchange Service 64
- Verifying the Provisioning of Exchange Mailbox 65

Creating a New Driver Object 65
- Creating the Driver Object in Designer 65
- Activating the Driver 73

Updating the Driver Files 73

4 Installing Azure AD 5.1.5 75

Preparing for Installation 75

Securing Driver Communication 75
Prerequisites 80
Identity Manager Exchange Service 86
Installing the Driver and the Identity Manager Exchange Service 86
Verifying and Starting the Identity Manager Exchange Service 87
Verifying the Provisioning of Exchange Mailbox 87
Creating a New Driver Object 88
Creating the Driver Object in Designer 88
Activating the Driver 96
Updating the Driver Files 96

5 Installing Azure AD Driver prior to 5.1.5 99

Securing Driver Communication 99
Securing Communication with Azure AD Graph 99
Securing Communication with Identity Manager Exchange Service 100
Installing the Driver Files 101
Preparing for Installation 101
Installing the Driver and the Identity Manager Exchange Service 106
Verifying and Starting the Identity Manager Exchange Service 107
Verifying the Provisioning of Exchange Mailbox 107
Creating a New Driver Object 107
Creating the Driver Object in Designer 107
Activating the Driver 115
Upgrading an existing Driver 115
What’s New in Version 5.1.0.0? 115
Working with MapDB 3.0.5 115
Upgrading the Driver 116

6 Procedure for upgrading from Azure AD Driver 5.1.5 to 5.1.6 or later 119

How to upgrade to Azure AD Driver 5.1.6 119
Upgrading the Installed Packages 121

7 Procedure for upgrading to Azure AD Driver 5.1.5 123

How to upgrade to Azure AD Driver 5.1.5 123
Upgrading the Installed Packages 125
Post Upgrade Tasks 126

8 Procedure for upgrading from Azure AD Driver 5.1.4 or earlier to 5.1.6 or later 127

How to upgrade to Azure AD Driver 5.1.6 127
Upgrading the Installed Packages 130
Post Upgrade Tasks 131

9 Transitioning from Existing Office 365 Driver to New Azure AD Driver 133

Preparing for Migrating Identities from Azure AD to Identity Vault 133
Migrating Identities. 133
Transitioning Assignments Through User Application 134

10 Understanding Identity Manager Exchange Service 137

11 Troubleshooting 139

Synchronizing country and usageLocation Attributes 139
Azure AD Password Complexity 140
Restoring the Driver to Current State 140
No Trusted Certificate Found Exception 141
Exchange Error During Driver Restart 141
Setting the set-executionPolicy to RemoteSigned in the Powershell 141
Email Address is Set Incorrectly for Groups that are Provisioned to a Different Valid Domain 141
Revoking Roles and Licenses in Hybrid Mode. 141
Setting Primary SMTP Address With EmailAddress Attribute Displays An Error 142
Mapping company Attribute with companyName Attribute Displays An Error. 142
Issue with the Size of PowerShell Log File 142
License Dependency in Developer Pack 142
User Name Cannot Contain Some Special Characters 142
Restoring a Mailbox or Mail User Displays a Warning Message 143
Random Errors While Connecting to the Exchange Portal 143
Adding a Graph User to Group Fails 144
No Trusted Certificate Found Exception 144
Driver Fails to Connect to Microsoft Graph API Due to Invalid Certificate Error 144
Driver Fails to Connect to Office 365 due to the MS Exchange License Issues 145
Driver Shuts Down if Role Assignment Fails in Azure Application. 145
Channel Exception on Too Many Requests 146
Error While Adding Users as Members to a Group in Azure 146
Certificate based authentication fails with “Refresh token and Client Secret must be provided to get
the new Access Token exception” Error when driver is configured to use proxy server. 146

A Driver Properties 147

Driver Configuration 147
Driver Module 148
Driver Object Password. 148
Authentication 148
Startup Option 149
Driver Parameters 149
Global Configuration Values. 152
Password Synchronization. 152
Driver Configuration 153
Account Tracking 154
Exchange Role Entitlement 154
Entitlements. 155
Managed System Information 158

B Known Issues and Limitations 161

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   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.

About This Guide

This guide explains how to install and configure the Identity Manager Driver for Office 365 and Azure Active Directory.

Audience

This guide is intended for administrators implementing Identity Manager, application server developers, Web services administrators, and consultants.

Feedback

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

Documentation Updates

For the most recent version of this document, see the Identity manager Driver Documentation Website (https://www.netiq.com/documentation/identity-manager-48-drivers/msazure_ad/data/netiq-idm-driver-for-office365-and-msazure.html).

Additional Documentation

For information on Identity Manager, see the Identity Manager Documentation Website (https://www.netiq.com/documentation/identity-manager-48-drivers).

1 Understanding the Office 365 and Azure Active Directory Driver

The Identity Manager Driver for Office 365 and Azure Active Directory (Azure AD driver) allows you to seamlessly provision and deprovision users, group memberships, exchange mailboxes, roles, and licenses to Azure AD cloud. You can also configure the driver to integrate with Identity Manager Service for Exchange Online (Identity Manager Exchange Service) for synchronizing Office 365 attributes.

Microsoft Office 365 is deprecating the Basic authentication method. To configure the driver with modern authentication, it is recommended to upgrade your driver to 5.1.3 or later. Prerequisites for the upgrade are explained in the following documents: "Prerequisites for the Driver" on page 80, "Prerequisites for Identity Manager Exchange Service" on page 81, and "Prerequisites for OAuth 2.0" on page 81. Once these prerequisites are met, proceed with the steps outlined in "Prerequisites for Support of Modern Authentication" on page 85.

Also, the performance improvement feature **Group cache clear interval** is introduced in the Subscriber channel from Driver version 5.1.4 and later. For more information, see "Group Cache Clear Interval" on page 20.

In general, you can perform the following tasks by using the driver:

- Synchronize users and groups on Publisher and Subscriber channels
- Provision and deprovision mail and mailbox users, distribution and mail enabled security and Office 365 groups
- Assign and revoke roles, group membership, and licenses using entitlements
- Extend the Azure AD schema
- Synchronize passwords from the Identity Vault

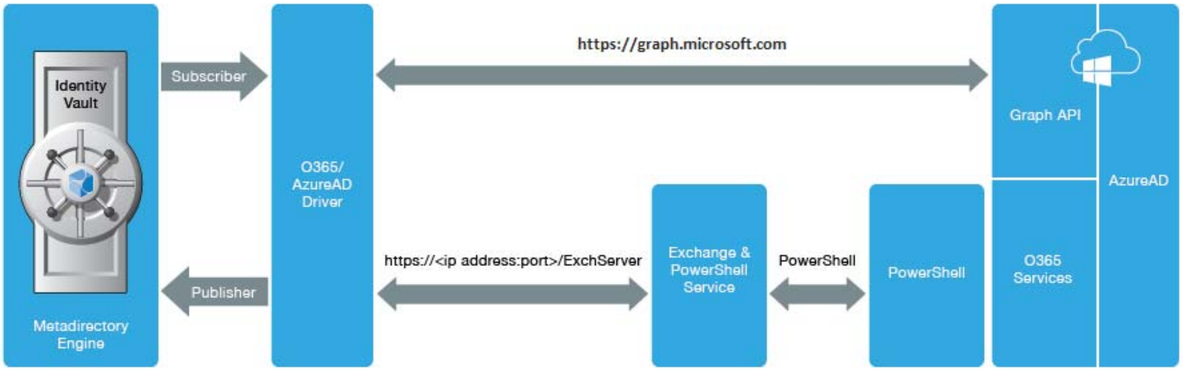
This section contains high-level conceptual information about the Azure AD driver.

- "Understanding How the Driver Works" on page 13
- "Data Transfers Between Systems" on page 15
- "Driver Features" on page 16
- "Planning to Install the Driver" on page 22

Understanding How the Driver Works

The following figure shows the data flow between Identity Manager and the Azure AD driver:

Figure 1-1



Azure AD Driver

The Azure AD driver allows you to seamlessly provision and deprovision users, group memberships, exchange mailboxes, roles, SKU and licenses to Azure AD (cloud). The driver synchronizes the user identity information between the Identity Vault and Azure AD and keeps this information consistent at all times.

Identity Manager Service for Exchange Online

The Azure AD driver uses the Identity Manager Exchange Service to provision or deprovision user mailboxes, mail users, create or remove distribution lists and security groups on Office 365 Exchange Online. For more information on configuring the service, see Chapter 10, “Understanding Identity Manager Exchange Service,” on page 137.

PowerShell

The Azure AD driver uses PowerShell for executing Exchange operations such as creation of Exchange mailbox, mail users, and groups.

Internet Protocols

The Azure AD driver uses the following Internet protocols to exchange data between Identity Manager and Azure AD.

- **REST (Representational State Transfer):** An HTTP-based protocol for exchanging messages over the network. It supports POST, PUT, GET, PATCH, DELETE methods to communicate with the application logic.

- **HTTPS (Hypertext Transfer Protocol):** An HTTP protocol over SSL (Secure Socket Layer) as a sub-layer under the regular HTTP application layering. HTTPS encrypts and decrypts user page requests as well as the pages that are returned by the web server.

  Azure AD processes a request and returns a REST response to the driver shim. The shim receives the response as an array of bytes and converts it to an XML document before passing it back to the driver policies. The input transformation style sheet processes the response and converts it into appropriate XDS that is reported back to the Identity Manager engine.

Identity Manager Engine

The Identity Manager engine uses XDS, a specialized form of XML (Extensible Markup Language), to represent events in the Identity Vault. Identity Manager passes the XDS to the driver policy which can consist of basic policies, DirXML Script, and XSLT (Extensible Stylesheet Language Transformation) style sheets. The Azure AD driver uses REST protocol to handle the HTTP transport of data between the Identity Vault and Azure AD.

The Subscriber channel receives XDS command documents from the Identity Manager engine, converts them to Azure AD API (Application Program Interface) calls, and executes them. The driver shim translates the XDS to XML payload on the Subscriber channel and then invokes the appropriate REST endpoints exposed by Azure AD for Object CRUD (Create, Read, Update, and Delete) operations.

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 passes the information between the shim and the Identity Manager engine.

For the Azure AD driver, you can choose to install the driver shim on the server where the Remote Loader is running.

Data Transfers Between Systems

The Azure AD driver supports data transfer on 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 Azure AD based on the event data.

You can configure the driver filter to allow both Azure AD and Identity Vault to modify the attribute(s). In this configuration, the most recent change determines the attribute value, except for merge operations that are controlled by filters and the merge authority.

The Exchange schema uses a different casing than the Azure AD schema where the first character of an Exchange schema attribute is uppercase, which is lowercase in Azure AD schema.

Driver Features

The Azure AD driver supports the following features:

- “Supported Operations” on page 16
- “Schema Extension” on page 18
- “License Handling” on page 19
- “Hybrid Mode” on page 19
- “Entitlements” on page 20
- “Group Cache Clear Interval” on page 20

Supported Operations

By default, the Azure AD Driver synchronizes User and Group Objects, and Exchange Mailboxes. You can customize the driver to synchronize additional classes and attributes.

The driver supports the following operations on the Publisher Channel:

- Add, Modify, Delete, And Query Operations.
- Migrate Operation Only Through Azure Ad Attributes. Exchange Attributes Are Not Supported For Migration.

NOTE: The Password Synchronization is not supported in the Publisher Channel.

Attributes that are not supported by MS Graph API on Publisher Channel and Query are as listed Below.

- “photo”
- “physicalDeliveryOfficeName”
- “LitigationHoldEnabled”
- “ArchiveStatus”
- “ServerLegacyDN”
- “Type”

Unsupported Attributes for Query User:

- “Birthday”
- “Hiredate”
- “Mailboxsettings”
- “Deviceenrollmentlimit”
- “Aboutme”
- “Interests”
- “Mysite”
- “Pastprojects”

- "Schools"
- "Skills"

Unsupported Attributes for Query Group:

- AutoSubscribeNewMembers
- hasMembersWithLicenseErrors
- allowExternalSender
- autoSubscribeNewMember
- hideFromAddressLists
- hideFromOutlookClients
- isSubscribedByMail
- unseenCount

The following Exchange Groups can be added through the Publisher Channel:

- Office 365 Group
- Distribution Group
- Security Group
- Mail Enabled Security Group
- Unified / Microsoft 365 Group

NOTE: You need to write a policy to set a value for **Equivalent To Me** while modifying a group membership.

The driver supports the following Operations on the Subscriber Channel:

- Add, Modify, Delete, Migrate, And Query Operations On Users And Groups.
- Add Or Delete Licenses Only On User Objects.
- Set Or Reset A Password Only On User Objects.
- Execute Powershell Cmdlets Using Policies.
- Assign Or Revoke Roles And Licenses In Hybrid Mode.

The following Exchange Groups can be added through the Subscriber Channel:

- Distribution Group
- Security Group
- Office 365 Group
- Mail Enabled Security Groups
- Unified/Microsoft 365 Groups

Support for SharePoint Online sites

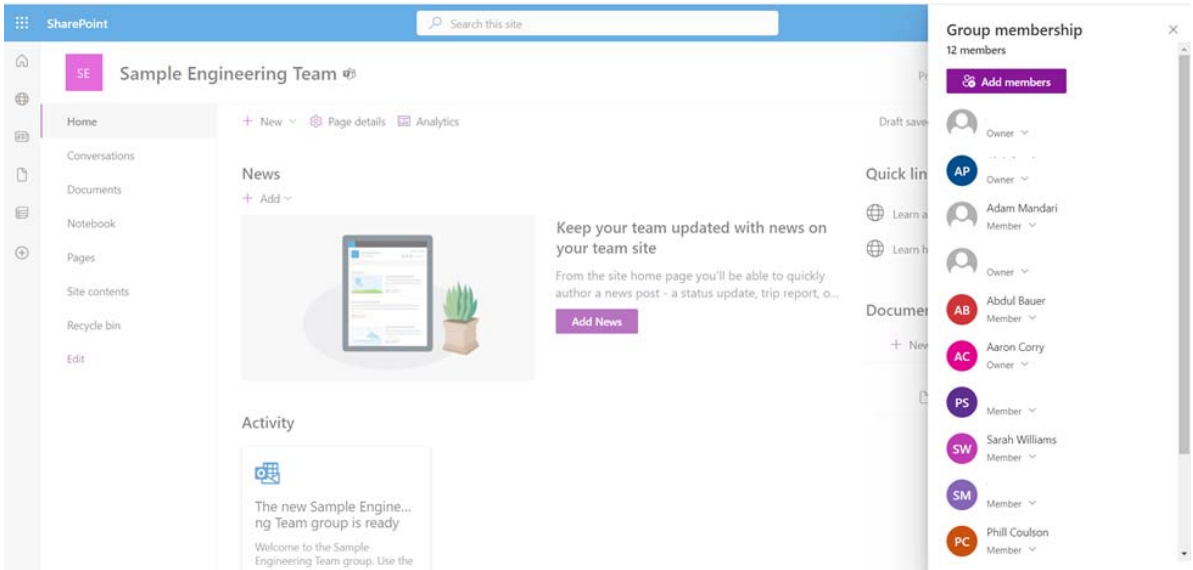
A SharePoint team site connects you and your team to shared content and resources. When you create a team site, a Microsoft 365 Group is automatically created. Any users you add to your team site are automatically added to your Microsoft 365 Group. Similarly, when you create a Microsoft 365 Group in Outlook, a SharePoint in Microsoft 365 team site is automatically created and any users you add to your Microsoft 365 Group get added to the team site.

NOTE: Only SharePoint Team site is supported but not the SharePoint Communication site.

Access the SharePoint Team site using the link: `https://<tenant>.sharepoint.com/sites/<group>`.

For example: https://asnlab.sharepoint.com/sites/SampleEngineeringTeam (https://asnlab.sharepoint.com/sites/SampleEngineeringTeam).

Figure 1-2



Keep your team updated with news on your team site

From the site home page you'll be able to quickly author a news post - a status update, trip report, o...

The new Sample Engine... ng Team group is ready

Welcome to the Sample Engineering Team group. Use the

Schema Extension

The Azure AD driver allows you to extend the Azure AD schema to include different types of attributes such as Integer, Boolean, and String using the driver parameters. For example, You can add, remove, or change extension attributes on a User or A Group Class with the allowed attribute types. For more information, see the Microsoft (https://docs.microsoft.com/en-us/graph/extensibility-overview) website.

License Handling

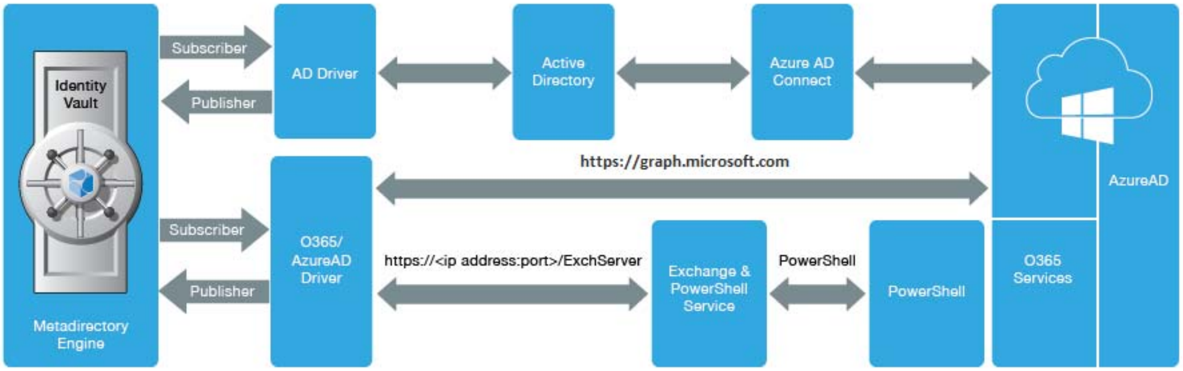
In the Office 365 driver, you created custom licenses to disable specific service plans. For example, to disable Office 365 ProPlus from your enterprise plan, you specified this string in the driver configuration. `OFFICESUBSCRIPTION`. For more information, see the *NetIQ Identity Manager Driver for Office 365 Implementation Guide*.

In the Azure AD driver, license handling is simplified. The driver lists all the available service plans for your subscription in Identity Applications. Each service plan can be individually assigned or revoked for a user through the License Entitlement resource. For example, if you want to assign only Office 365 ProPlus to a user, select `OFFICESUBSCRIPTION` from the list of service plans. The driver also supports assigning or revoking multiple plans to a user.

Hybrid Mode

The following figure shows the hybrid mode deployment scenario:

Figure 1-3



In this deployment, Azure AD Connect integrates on-premise Active Directory with Azure Active Directory. Azure Active Directory does not allow modifications on user and group objects that were synchronized through Azure AD Connect. However, it allows you to provision roles and licenses to the users. To accomplish this, you must deploy an Azure AD driver in hybrid mode. To synchronize identities from the Identity Vault to on-premise Active Directory, you must have an AD driver in your environment.

NOTE: If you have upgraded to the 5.0.1 version of the driver, the group membership entitlement and exchange role entitlement are supported in hybrid mode. For more information on entitlements supported in 5.0.1, see Exchange Role Entitlement.

To provision roles and licenses, set the driver in hybrid mode in Designer or iManager. For more information, see “Configuring the Driver” on page 93.

The driver performs the following actions when operating in hybrid mode:

- When a user is provisioned to AD through AD driver's user account entitlements and the user is synchronized to Azure AD through Azure AD Connect, the driver updates the user association in the Identity Vault.
- When a user is deleted from Azure AD, the driver removes the association for the user from the Identity Vault.
- When a user is granted or revoked roles or licenses through entitlements, the driver grants or revokes roles or licenses after an association is created for the user.
- When an account entitlement is revoked for a user in AD, the driver removes the association for the user from the Identity Vault.

NOTE: You cannot add users, delete users, and modify user attributes through the publisher channel when you operate the Azure AD driver in hybrid mode. However, the Azure AD driver will update the associations accordingly.

In hybrid mode, the AD driver's account tracking takes precedence over Azure AD driver. The password synchronization to the Identity Vault is handled by AD driver in hybrid mode.

Entitlements

The Azure AD driver supports entitlements. By default, it supports UserAccount, Group, Licenses, SKU and Roles entitlements.

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 resources in Azure AD. You can use entitlements to grant the rights to an account in Azure AD, to control group membership, roles, and licenses.

Group Cache Clear Interval

The **Group cache clear interval** is a feature introduced in the Subscriber channel, to improve the driver’s performance while adding or removing group’s members. This parameter can be specified in hours. For example, if you specify 1 in this field, it indicates 1 hour, and 2 indicates 2 hours, etc.

NOTE: ◆This feature is applicable for Driver versions 5.1.4 and later.

- The Driver also requires an upgrade of the base package to `MFAZUREBASE_1.0.3.20210225161038.jar` and above, to configure this feature.

The Group cache clear interval feature is applicable only for the following Group types:

- Distribution Lists
- Mail Enabled Security Groups
- Unified/Microsoft 365 Groups

When you add or remove a member to a group for the first time, the driver cache is updated based on the calls made to Azure application. For subsequent modifications to the group membership, the driver makes one call to the application to find the group and retrieves all the other required information from the in-memory cache, thereby reducing the time taken to process the changes of the group membership.

Clearing and rebuilding the cache is an event driven process. The group cache build time starts when an event is processed, and the cache is built as subsequent events occur. Upon processing subsequent events, the driver checks for the value set in the **Group cache clear interval** parameter in driver configuration. If these subsequent events are encountered:

- within the specified duration, the group cache is maintained
- post the specified duration, the cache is cleared and rebuilt. The Group cache build time is also reset.

For example, the driver is started at 9:00 AM with the **Group cache clear interval** specified in hours as **1**, in Driver configuration. Driver processes any event on Subscriber channel at 9:10 AM then this time will be the cache build start time. Post this, the driver processes add or remove members to *AzureUnified* (Unified type Group) or *AzureDistribution* (Distribution list type Group) then all the information fetched from the application while processing the modify event gets added to Group Cache, and is maintained until the Group cache clear interval is hit, that is till 10:10 AM and any event processed by Subscriber Channel post 10:10 AM will trigger the group cache to clear, rebuilt and reset the Group cache build start time.

To see the group cache related messages, you must set the trace level to 7.

To specify this value in Designer, double click the Azure Driver’s connector line, and navigate to **Driver Configuration > Driver Parameters > Subscriber Options**.

Specify a numerical value (for example, -1, 0, 1, 2, etc.) in the **Group cache clear interval** field. The numerical value specified here corresponds to hours.

- **-1**: indicates the cache feature is disabled.
- **0**: indicates that the cache is cleared only upon driver restart.
- **1**: indicates the cache feature is enabled and cleared every one hour.
- **2**: indicates the cache feature is enabled and cleared every two hours.

NOTE: The Group cache feature works only when the member of a group is processed in an event. If any other attribute of a group (for example, description) is modified along with the member addition or removal, this feature does not have any impact.

Planning to Install the Driver

This section provides information for planning the installation and configuration for the driver.

- "Options for Installing the Driver Shim" on page 22

Options for Installing the Driver Shim

You can install the driver shim on the Identity Manager server or the Remote Loader server.

For more information about the platforms supported with Identity Manager or the Remote Loader, see the NetIQ Identity Manager Technical Information website (https://www.netiq.com/products/identity-manager/advanced/technical-information/).

The Remote Loader loads the driver and communicates with the Identity Manager engine on behalf of the driver installed on the remote server. For information about configuring the Identity Manager drivers with the Remote Loader, see Configuring the Remote Loader and Drivers in the *NetIQ Identity Manager Setup Guide for Windows*.

2 Installing Azure AD 5.1.7

The Identity Manager Driver for Office 365 and Azure Active Directory (Azure AD driver) allows you to seamlessly provision and deprovision users, group memberships, exchange mailboxes, roles, and licenses to Azure AD cloud. You can also configure the driver to integrate with Identity Manager Service for Exchange Online (Identity Manager Exchange Service) for synchronizing Office 365 attributes.

In this release, we are supporting National Cloud Deployment (NCD) and Business to Customer(B2C).

There are some configuration scenarios that need to be considered:

- For NCD: While configuring Azure AD Driver for NCD, all the standard options will be supported and It is require to provide MS Graph API Base and Token URLs.
- For B2C: While configuring Azure AD Driver for a B2C tenant, only the subscriber channel is supported. However, please be aware that the Publisher Channel needs to be disabled and Exchange Online service will not be supported in this deployment scenario.

NOTE:

- It is recommended to update the driver base package to version 1.0.6.20230627192740, but not mandatory unless you are connecting to B2C or National Cloud Deployment tenants.
- It is recommended to update as it provides fix for Defect 763002 - Azure AD shim 5.1.6 GET 404 not found - attribute 'print' no longer supported in queries added the print attribute in Unsupported User Attributes by MS Graph APIs in the driver configuration.

Perform the following set of procedure to install Azure AD 5.1.7:

- "Preparing for Installation" on page 49
- "Certificate Based Authentication support for Identity Manager Exchange Service" on page 63
- "Creating a New Driver Object" on page 65
- "Updating the Driver Files" on page 47

Preparing for Installation

Prerequisites

This section provides the prerequisites, considerations, and system setup needed to install the driver:

- "Prerequisites for the Driver" on page 24
- "Prerequisites for the User Account to be Configured in the Driver" on page 24
- "Prerequisites for Identity Manager Exchange Service" on page 25
- "Prerequisites for OAuth 2.0" on page 25

- "Assigning the Rights to the Application" on page 28
- "Prerequisites for Support of Modern Authentication" on page 30

NOTE: - In case of fresh deployment of the driver with Identity Manager 4.8.6 or earlier then make sure that following jars are present:
    - `asm-1.0.2.jar`
    - `content-type-2.2.jar`
    - `nimbus-jose-jwt-9.23.jar`
    - `oauth2-oidc-sdk-9.39.jar`
    - `msal4j-1.12.0.jar`
    - `slf4j-log4j12-1.7.33.jar`
    - `common-2.49.jar`
    - `json-smart-2.4.8.jar`

    These jars are bundled with Azure 5.1.6 driver zip file and can be found under common folder after extracting the zip file, For more details check "Updating the Driver Files" on page 73 of Chapter 3, "Installing Azure AD 5.1.6," on page 49.
- If you are running Identity Manager 4.8.7, then the latest version of the above mentioned jars will be bundled with it.

Prerequisites for the Driver

The driver requires the following applications:

- Identity Manager 4.8.4 or later
- Identity Manager Designer 4.8.4 or later
- Identity Manager REST driver 1.1.2.0400 or later

Prerequisites for the User Account to be Configured in the Driver

You must ensure that the user account you are configuring in the driver has the following Roles or Permissions in the Azure application:

- At a minimum, the user account must have the following roles:
    - User Administrator
    - Exchange Administrator
    - Privileged Role Administrator
    - Teams Administrator
- The multi-factor authentication for the user account is disabled.
- A user account has conditional access. For more information on providing conditional access, see What is Conditional Access.

NOTE: You may need to provide additional roles based on their requirement, For more information check New built-in roles (https://techcommunity.microsoft.com/t5/microsoft-entra-azure-ad-blog/16-new-built-in-roles-including-global-reader-now-available-in/ba-p/900749).

Prerequisites for Identity Manager Exchange Service

- Microsoft Windows Server 2016, or Microsoft Windows Server 2019.
- Install .NET Framework 4.7.2 or later.
- Microsoft Visual C++ 2017 Redistributable packages for Visual Studio.

  Download the packages from the Microsoft Downloads website.
- Upgrade to PowerShell 5.1 or later.
- Update PowerShellGet to the latest version using Install-Module PowerShellGet -Force-5
- Install the ExchangeOnlineManagement and Microsoft Graph modules by executing the below commands in PowerShell:
  - `Install-Module -Name ExchangeOnlineManagement`
  - `Install-Module Microsoft.Graph -Scope AllUsers`

  NOTE: If you get “Unable to resolve package source” error, then run `[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12` command before running the Install-Module commands.
- Windows Azure AD Module for Windows Powershell on the computer where you will install Windows Powershell service.

  Perform the following steps to upgrade PowerShell to the latest version:
  1. Open a Windows PowerShell console.
  2. Run the following Install-Module cmdlet or Install-Script cmdlet:
     - If it is a module: `Install-Module -Name <moduleName> -RequiredVersion <version>`

       For example, `Install-Module -Name MSOnline`. Refer to www.powershellgallery.com (https://www.powershellgallery.com/packages/MSOnline/1.1.183.66)
     - If it is a script: `Install-Script -Name <scriptName> -RequiredVersion <version>`

Identity Manager Exchange Service can be run on a user configured port. However, the service cannot be used with any other REST client tools.

Prerequisites for OAuth 2.0

The driver uses OAuth 2.0 protocol to authenticate to Azure AD. To support this protocol for authentication, you need to have a proxy application for the Azure AD driver on Azure AD. The Client ID and Client Secret allotted to the application will be later used in the Azure AD driver configuration. For more information about Azure Active Directory Application Proxy, see Microsoft Azure documentation.

Creating a Proxy Application on Azure AD

A proxy application is created in the Azure Portal. Creating a proxy application involves the following steps:

1. Registering an application and obtaining a client ID. For more information see, Registering an Application.
2. Generating an application password or the client secret. For more information see, Certificates and Secrets (https://learn.microsoft.com/en-us/entra/identity-platform/quickstart-register-app#add-credentials).
3. Configuring API permissions (Delegated and Application permissions). Set the delegated and application permissions as shown in the following table. For more information see, Add permissions to access web APIs.

Table 2-1 *List of Application type of APIs*

API	Type	Description	Admin Consent
Application.Read.All	Application	Read all applications	Grant Admin Consent
Application.ReadWrite.All	Application	Read and write all applications	Grant Admin Consent
AuditLog.Read.All	Application	Read all audit log data	Grant Admin Consent
Channel.Create	Application	Create channels	Grant Admin Consent
Channel.Delete.All	Application	Delete channels	Grant Admin Consent
Channel.ReadBasic.All	Application	Read the names and descriptions of all channels	Grant Admin Consent
ChannelMember.Read.All	Application	Read the members of all channels	Grant Admin Consent
ChannelMember.ReadWrite.All	Application	Add and remove members from all channels	Grant Admin Consent
ChannelSettings.Read.All	Application	Read the names, descriptions, and settings of all channels	Grant Admin Consent
ChannelSettings.ReadWrite.All	Application	Read and write the names, descriptions, and settings of all channels	Grant Admin Consent
Device.Read.All	Application	Read all devices	Grant Admin Consent
Device.ReadWrite.All	Application	Read and write devices	Grant Admin Consent
Directory.Read.All	Application	Read directory data	Grant Admin Consent
Directory.ReadWrite.All	Application	Read and write directory data	Grant Admin Consent
Domain.ReadWrite.All	Application	Read and write domains	Grant Admin Consent

API	Type	Description	Admin Consent
Group.Create	Application	Create groups	Grant Admin Consent
Group.Read.All	Application	Read all groups	Grant Admin Consent
Group.ReadWrite.All	Application	Read and write all groups	Grant Admin Consent
GroupMember.Read.All	Application	Read all group memberships	Grant Admin Consent
GroupMember.ReadWrite.All	Application	Read and write all group memberships	Grant Admin Consent
Team.Create	Application	Create teams	Grant Admin Consent
Team.ReadBasic.All	Application	Get a list of all teams	Grant Admin Consent
TeamMember.Read.All	Application	Read the members of all teams	Grant Admin Consent
TeamMember.ReadWrite.All	Application	Add and remove members from all teams	Grant Admin Consent
TeamMember.ReadWrite NonOwnerRole.All	Application	Add and remove members with non-owner role for all teams	Grant Admin Consent
TeamSettings.Read.All	Application	Read all teams' settings	Grant Admin Consent
TeamSettings.ReadWrite.All	Application	Read and change all teams' settings	Grant Admin Consent
User.Read.All	Application	Read all users	Grant Admin Consent
User.ReadWrite.All	Application	Read and write all users' full profiles	Grant Admin Consent
UserAuthenticationMethod.Read.All	Application	Read all user authentication methods	Grant Admin Consent
UserAuthenticationMethod.ReadWrite.All	Application	Read and write all users authentication methods	Grant Admin Consent

Table 2-2 *List of Delegated type of APIs*

API	Type	Description	Admin Consent
AuditLog.Read.All	Delegated	Read audit log data	Grant Admin Consent
Directory.AccessAsUser.All	Delegated	Access directory as the signed in user	Grant Admin Consent
Directory.Read.All	Delegated	Read directory data	Grant Admin Consent
Directory.ReadWrite.All	Delegated	Read and write directory data	Grant Admin Consent
Group.Read.All	Delegated	Read all groups	Grant Admin Consent

API	Type	Description	Admin Consent
Group.ReadWrite.All	Delegated	Read and write all groups	Grant Admin Consent
GroupMember.Read.All	Delegated	Read group memberships	Grant Admin Consent
GroupMember.ReadWrite.All	Delegated	Read and write group memberships	Grant Admin Consent
RoleManagement.Read.All	Delegated	Read role management data for all RBAC providers	Grant Admin Consent
RoleManagement.Read.Directory	Delegated	Read directory RBAC settings	Grant Admin Consent
UserAuthenticationMethod.Read	Delegated	Read user authentication methods	Grant Admin Consent
UserAuthenticationMethod.Read.All	Delegated	Read all user’s authentication methods	Grant Admin Consent
UserAuthenticationMethod.ReadWrite	Delegated	Read and write user authentication methods	Grant Admin Consent
UserAuthenticationMethod.ReadWrite.All	Delegated	Read and write all user authentication methods	Grant Admin Consent
User.Read	Delegated	Sign in and read user profile	Grant Admin Consent
User.Read.All	Delegated	Read all users' full profiles	Grant Admin Consent
User.ReadBasic.All	Delegated	Read all users' basic profiles	Grant Admin Consent
User.ReadWrite	Delegated	Read and write access to user profile	Grant Admin Consent
User.ReadWrite.All	Delegated	Read and write all users' full profiles	Grant Admin Consent

NOTE: You may need to provide additional permissions based on their requirement.

The Client ID and Client Secret can now be used for driver configurations or any other REST clients.

Assigning the Rights to the Application

1 In the server where you have installed the exchange service, login to PowerShell and connect to the Office 365 Exchange Online service, using the following command:

```
Connect-MSolService
```

2 To obtain the Client ID for your application, replace <AppPrincipalId> with the Client ID that you obtained from Creating a Proxy Application on Azure AD and run the following commands in PowerShell.

```
Get-MsolServicePrincipal | ft DisplayName, <AppPrincipalId> -AutoSize
```

```
$ClientIdWebApp = '<AppPrincipalId>'
```

```
$webApp = Get-MsolServicePrincipal –AppPrincipalId $ClientIdWebApp
```

3 Assign the `Company Administrator` rights to your application using the Client ID obtained in Step 2 by running the following command:

```
Add-MsolRoleMember -RoleName "Company Administrator" -RoleMemberType
ServicePrincipal -RoleMemberObjectId $webApp.ObjectID
```

The `Company Administrator` role will give you rights to delete the directory objects.

IMPORTANT: The `Company Administrator` role in Identity Manager is mapped to the `Global Administrator` role in Azure AD.

Ensure that the account used by the driver to connect to the Exchange Online service has the correct roles to load and execute the following cmdlets:

- New-Mailbox
- Set-Mailbox
- Get-Mailbox
- Remove-Mailbox
- New-MailUser
- Set-MailUser
- Get-MailUser
- Remove-MailUser
- Set-User
- Get-User
- New-DistributionGroup
- Set-DistributionGroup
- Set-Group
- Get-DistributionGroup
- Get-Group
- Remove-DistributionGroup
- Add-DistributionGroupMember
- Remove-DistributionGroupMember
- Get-DistributionGroupMember
- Add-RoleGroupMember
- Remove-RoleGroupMember
- Get-RoleGroupMember
- New-UnifiedGroup
- Get-UnifiedGroup

- Set-UnifiedGroup
- Remove-UnifiedGroup
- Add-UnifiedGroupLinks
- Remove-UnifiedGroupLinks
- Get-UnifiedGroupLinks
- Get-MsolUser
- Set-MsolUser

Absence of the required roles prevents the driver from executing the cmdlets that require those roles.

Prerequisites for Support of Modern Authentication

As Microsoft Office 365 is deprecating the **Basic** authentication, you must now configure the driver with modern authentication method. You must also ensure to have the earlier mentioned prerequisites (“Prerequisites for the Driver” on page 80, “Prerequisites for Identity Manager Exchange Service” on page 81, and “Prerequisites for OAuth 2.0” on page 81) met, and then proceed with the following prerequisites.

The following prerequisites are specific to modern authentication. It is highly recommended to upgrade the driver version 5.1.x to 5.1.3 or later to support modern authentication.

Installing the Microsoft Exchange Online PowerShell V3 (EXO V3)

You must install the Microsoft Exchange Online PowerShell V3 module to support the new API’s. For more information on EXO V3 module, see About the Exchange Online PowerShell V3 module (https://learn.microsoft.com/en-us/powershell/exchange/exchange-online-powershell-v2?view=exchange-ps#updates-for-the-exo-v3-module).

- For prerequisites to install the EXO V3 module, see Prerequisites for EXO V3 module (https://learn.microsoft.com/en-us/powershell/exchange/connect-to-exchange-online-powershell?view=exchange-ps&source=recommendations).
- For installing the EXO V3 module, see Install the EXO V3 module (https://learn.microsoft.com/en-us/powershell/exchange/connect-to-exchange-online-powershell?view=exchange-ps&source=recommendations).

Configuring Azure AD Proxy Application for Modern Authentication Methods

You must enable the permission in the Azure portal to access Microsoft Office 365 with modern authentication.

NOTE: Before upgrading to 5.1.7 ensure to run this command on the server running the Identity Manager Exchange Service using PowerShell.

The procedure to set the permission is shown below:

1. Login to the Azure AD Portal.
2. Select **Azure Active Directory**.

3 Navigate to **App Registration >** find and select your application in the list (for example: *<MySample_Azure_Appln>*) **> Authentication > Advanced Settings**.

4 Set **Enable the following mobile and desktop flows** under **Allow public client flows** permission to **Yes**.

IMPORTANT: The multi-factor authentication (MFA) must be disabled for the Azure account which is used with the driver.

Installing the Driver and the Identity Manager Exchange Service

You can install the Azure AD driver on the Identity Manager server or with the Remote Loader. The driver installation program guides you through the driver and the Identity Manager Exchange Service installation.

NOTE:

1. IDM Exchange service must be run on the same machine as the driver and configured to listen only on local host.
2. IDM Exchange service must be run with least privilege required for the configured `PowerShell cmdlets` to execute.
3. Only system administrator must be provided access to the IDM exchange service machine

Perform the following actions to install and configure the Exchange Service:

1 Copy Exchange Service from `[ISO]:\products\IDM\windows\setup\drivers\azuread\ExchangeService` to any local drive on the server you intend to run this service.

2 Navigate to the directory where you copied the ExchServerHost.exe in the first step.

For example, `C:\ExchangeService`

3 Run the following command to install the Exchange Service:

```
<location of InstallUtil>\InstallUtil.exe ExchServerHost.exe
```

For example:

```
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\InstallUtil.exe
ExchServerHost.exe
```

where, `C:\Windows\Microsoft.NET\Framework64\v4.0.30319\InstallUtil.exe` is the location where `InstallUtil.exe` is located.

4 Ensure the server certificate is available in iManager. To create the server certificate, see "Securing Communication with Identity Manager Exchange Service" on page 76

5 Open cmd prompt, and navigate to the local drive location where the `ExchangeService` is saved, as mentioned in Step 1 on page 76 (`\products\IDM\windows\setup\drivers\azuread\ExchangeService\`), and execute the `configureExchService` batch file as explained below.

NOTE: When upgrading the driver from 5.1.6 to 5.1.7 in the Identity Manager Exchange Server, you need to replace ExchServerHost.exe,IDMExchServer.dll files and configureExchService.bat files located in `\ExchangeService` folder.

While installing the Identity Manager Exchange Service for the first time using the port and certificate, you need to add an additional argument **AuthenticationType**. This argument refers to whether you want to use the Basic Authentication or Certificate Based Authentication. The certificate friendly name provided in the command should be the nickname (exchcba) of the certificate to be created for the Identity Manager Exchange Service.

Example for Basic Authentication Command:

```
configureExchService.bat 9001 exchcba 0
```

Example for Certificate Based Authentication Command:

```
configureExchService.bat 9001 exchcba 5
```

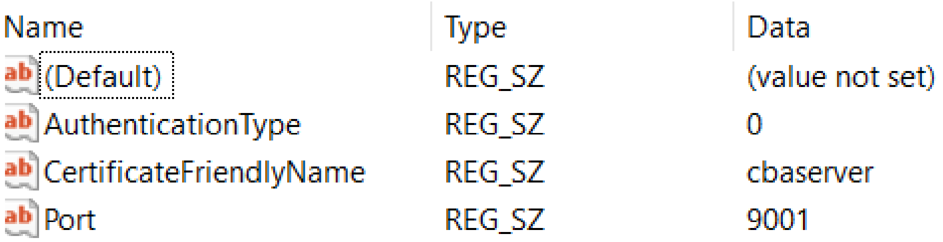
There is another way of choosing the authentication type via Registry Editor as explained below.

Goto Registry Editor and navigate to **HKEY_LOCAL_MACHINE** -> **Software** -> **Novell** -> **ExchServer**

You can edit the value of the key **AuthenticationType** as required.

Below is the snapshot of the Registry Editor where only 3 keys are present after upgrading to 5.1.6.

Figure 2-1

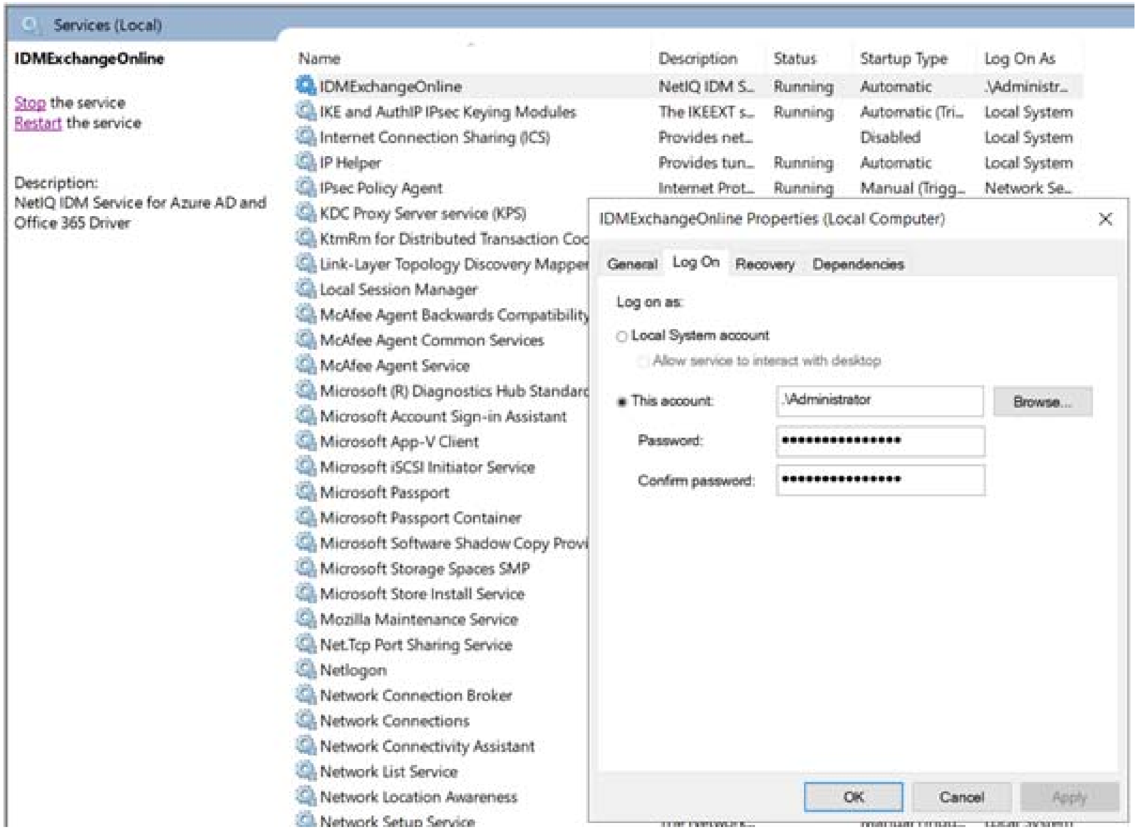


Name	Type	Data
ab (Default)	REG_SZ	(value not set)
ab AuthenticationType	REG_SZ	0
ab CertificateFriendlyName	REG_SZ	cbaserver
ab Port	REG_SZ	9001

NOTE: When you choose Basic Authentication in **iManager**, go to the **Identity Manager Exchange Server** and edit the value of **AuthenticationType** in the registry to 0. Similarly when you choose Certificate Based Authentication in **iManager**, then go to the **Identity Manager Exchange Server** and edit the value of **AuthenticationType** in the registry to 5.

Apart from this, you need to run the **IDMExchangeOnline** service as an Administrator. Run **Services.msc** and go to the **IDMExchangeOnline**, right click **IDMExchangeOnline** and select **Properties** and under **Log On**, Select "**This account**" and provide the credentials of Local administrator account as shown below.

Figure 2-2



Services (Local)

IDMExchangeOnline

Stop the service
Restart the service

Description:
NetIQ IDM Service for Azure AD and Office 365 Driver

Name	Description	Status	Startup Type	Log On As
IDMExchangeOnline	NetIQ IDM S…	Running	Automatic	.\Administr…
IKE and AuthIP IPsec Keying Modules	The IKEEXT s…	Running	Automatic (Tri…	Local System
Internet Connection Sharing (ICS)	Provides net…		Disabled	Local System
IP Helper	Provides tun…	Running	Automatic	Local System
IPsec Policy Agent	Internet Prot…	Running	Manual (Trigg…	Network Se…
KDC Proxy Server service (KPS)				
KtmRm for Distributed Transaction Coc				
Link-Layer Topology Discovery Mapper				
Local Session Manager				
McAfee Agent Backwards Compatibility				
McAfee Agent Common Services				
McAfee Agent Service				
Microsoft (R) Diagnostics Hub Standard				
Microsoft Account Sign-in Assistant				
Microsoft App-V Client				
Microsoft iSCSI Initiator Service				
Microsoft Passport				
Microsoft Passport Container				
Microsoft Software Shadow Copy Provi				
Microsoft Storage Spaces SMP				
Microsoft Store Install Service				
Mozilla Maintenance Service				
Net.Tcp Port Sharing Service				
Netlogon				
Network Connection Broker				
Network Connections				
Network Connectivity Assistant				
Network List Service				
Network Location Awareness				
Network Setup Service				

IDMExchangeOnline Properties (Local Computer)

Log on as:

○ Local System account

□ Allow service to interact with desktop

● This account: .\Administrator

Password: ••••••••••••••••

Confirm password: ••••••••••••••••

NOTE: Windows Local Administrator account is the one that gets created when you install a Windows Server and this account is not part of domain.

6 To start the service, navigate to **Control Panel** > **Administrative Tools** > **Services**.

7 Right-click the **IDMExchangeOnline** service and select **Start**.

NOTE: To uninstall the service, open a .NET command prompt and issue the `InstallUtil /u` ExchServerHost.exe command.

NetIQ recommends you to use TLS 1.1 and TLS 1.2 protocols with the Identity Manager Exchange Service. If you are using ciphers and protocols such as RC4 and Triple DES, or SSLv2/v3 on a server running Identity Manager Exchange Service, you must disable them using the `disableWeakCiphers.reg` file provided in the Exchange Service installation directory. You can either execute the registry file or import the file into Windows Registry. After the changes are made, restart the server. For more information about restricting the use of certain cryptographic algorithms and protocols on Windows servers, see Microsoft Support Site.

Securing Driver Communication

The driver communicates over SSL with Azure AD and Identity Manager Exchange Service.

IMPORTANT: The connection accepts certificates only from a Java keystore. Make sure that the keystore for the certificates is a Java keystore.

The following sections provide instructions for creating a secure connection:

- "Securing Communication with Identity Manager Exchange Service" on page 34
- "Secured Communication with Microsoft Graph API" on page 35
- "Certificate Based Authentication in Azure AD Driver 5.1.7" on page 36

Securing Communication with Identity Manager Exchange Service

To set up SSL between the driver and Identity Manager Exchange Service, you need to create and import a server certificate into the root certificate store of the Windows server where the service is deployed. The following procedure assumes eDirectory as the Certificate Authority (CA).

1 Login to **Identity Console** as an Administrator -> **Certificate Management** -> **Server Certificate Management** -> Click "**+**" icon

   1a Provide the **Nickname** and for creation method select **Custom**.

   For example: Nickname could be "exchcba".

   1b Select **Organizational Certificate Authority**.

   1c Click **Next**.

   1d Under Certificate Parameters, click + for Subject Alternative Names, select **Type** (Choose **IP Address** from the drop-down and **Name**: Enter the IP Address of the Server running Identity Manager Exchange Service). Click **Next**.

   1e Select Your organization's certificate and click **Next**.

   1f Click **OK** until the certificate is successfully created.

2 Export the server certificate created for Identity Manager Exchange Service in step 1a, from the connected eDirectory server and save it to a file in the `pfx` format.

   2a Login to Identity Console, log in to the connected eDirectory server with administrator rights.

   2b Click Certificate Management > Server Certificate Management , then select the Server Certificate that has been created (ex: exchcba) for exchange service in step1.

   2c Click 📤 to export the certificate.

   2d Ensure that by default the Identity Manager Exchange Server Certificate (as created in Step1) has been selected and select **Export Private Key**.

   2e Enter the password and click **OK**.

   2f To save the certificate to a file, click **Save the Exported File**.

3 Import the certificate to the trusted store of the Windows server on which you will run Identity Manager Exchange Service.

   3a Copy the .`pfx` file to the Windows server.

   3b Click **Start** > **Run**> **mmc**.

   3c Click **File** > **Add/Remove Snap-in**.

3d Select **Certificates** and click **Add** to import this snap-in by choosing *Computer account*.

3e Click **Finish**.

3f Navigate to **Certificates** > **Trusted Root Certification Authorities**.

3g Right-click and then select **All Tasks** > **Import**.

3h On the **Welcome to the Certificate Import Wizard** page, click **Next**.

3i Click **Browse** and select the eDirectory certificate you exported in "Export the server certificate from the connected eDirectory server and save it to a file in the `pfx format.`" `on page 76`

3j Specify the password and click **Next**.

3k Click **Finish** to import the certificate into the trust store.

4 Start Identity Manager Exchange Service. For more information, see "Verifying and Starting the Identity Manager Exchange Service" on page 87.

5 Open the following Exchange service URL from your browser:

```
https://<Exchange_Service>:Port/ExchServer
```

6 Obtain the public certificate and import it into the same keystore which was created and placed in IDM Server as mentioned in (for example, the keystore `azuread`).

For example, perform the following steps to obtain a public certificate on Google Chrome:

6a Click 🔒 from the address bar and then click **Details**.

6b In the **Security** tab, click **View Certificate**.

6c In the **Details** tab, click **Copy to File**.

6d In the **Certificate Import Wizard**, click **Next**.

6e Select **DER encoded binary** and click **Next**.

6f Click **Browse** and navigate to the directory where you want to save the certificate.

6g Specify a name for the certificate and click **Next**.

6h Click **Finish** to complete the export.

6i Add the exported key to the driver keystore by using the following Java keytool command:

```
keytool -import -file <path to the exchange cert
file>\<certname.cer> -keystore <mykeystore> -alias <aliasname>
```

NOTE: Ensure that certificates inside the keystore have different alias names for all the imported certificates. Use the existing driver keystore if you are upgrading the driver from 5.1.5 to 5.1.6.

Secured Communication with Microsoft Graph API

To set up SSL between the driver and Azure AD graph REST endpoints, perform the following steps:

1 Open the following URL from your browser:

- `https://login.microsoftonline.com/`
- `https://graph.microsoft.com/`
- `https://azure.microsoft.com/`

2 Obtain the public certificate and import it into the keystore.

For example (Suppose you are accessing https://graph.microsoft.com/), if you are using Google Chrome, perform the following steps:

2a In the address bar, click 🔒 and then click › next to browser address bar (for example:**graph.microsoft.com**).

2b Select **Certificate (Valid)**. The certificate is displayed.

2c Click **Certification Path**. The Certification Path displays the hierarchical structure of the structure of all the certificates.

2d Select the root certificate (the top most parent certificate), and click **View Certificate**. The root certificate is displayed.

2e To save the certificate to your system, click **Details > Copy to File > Next > Next**.

2f Enter a filename for the certificate and save it to a location as required.

2g Add the exported key to the driver keystore using the following Java keytool command:

You might have to create a new keystore(`.jks` file), if one such file doesn't exist already. This keystore file will contain the public certificate of the Azure graph endpoint and the exchange service certificate.

```
keytool -import -file <path to the graph cert file>\<certname.crt> -
keystore <mykeystore> -alias <aliasname>
```

For example:
```
keytool -import -file msgraph.cer -keystore azuread.jks -
alias msgraph
```

NOTE:
- Ensure to place the new keystore in IDM Server. In case of Remote Loader place the keystore file in the system where the Azure AD driver is running.
- Ensure that you follow the above steps to import all the certificates into the keystore.

Certificate Based Authentication in Azure AD Driver 5.1.7

1 Generate a password protected private key in PEM format by executing the below commands. You can use an existing password protected private key in PEM format or create a password protected private key using the below command.

```
openssl genrsa -aes256 -out private_key.pem 2048
```

Note: this command will ask you to provide a password for this private key.

2 Generate a certificate signing request using the private key.

```
openssl req -new -key private_key.pem -out cert.csr
```

Note: This command will ask you to input the password for the private key.

Also, this command will ask for a variety of extra information, like company name, country, and a password. None of this is used by the sample, so you can set these values as nothing/anything you want.

3 Generate a x509-certificate using the csr file and the private key

```
openssl x509 -req -days 365 -in cert.csr -signkey private_key.pem -out
cert.crt
```

Note: This command will ask you to input the password for the private key

4 Generate the pfx certificate using the crt file and the private key

```
openssl pkcs12 -export -out certcba.pfx -inkey private_key.pem -in
cert.crt -name "mypfxfile"
```

Note: This command will first ask you to input the password for the private key.

Then it asks you to provide the password for the pfx certificate which will be used as the keystore password in configuring Azure CBA. In the above command the "-name" parameter refers to alias which will be used as an input in configuring Azure CBA.

5 Using Designer, perform the following steps.

5a Right Click Driver Object and click Properties.

5b Goto Driver Configuration -> Driver Parameter

5c Under Driver Options, choose Certificate Based Authentication as authentication type and provide following details.

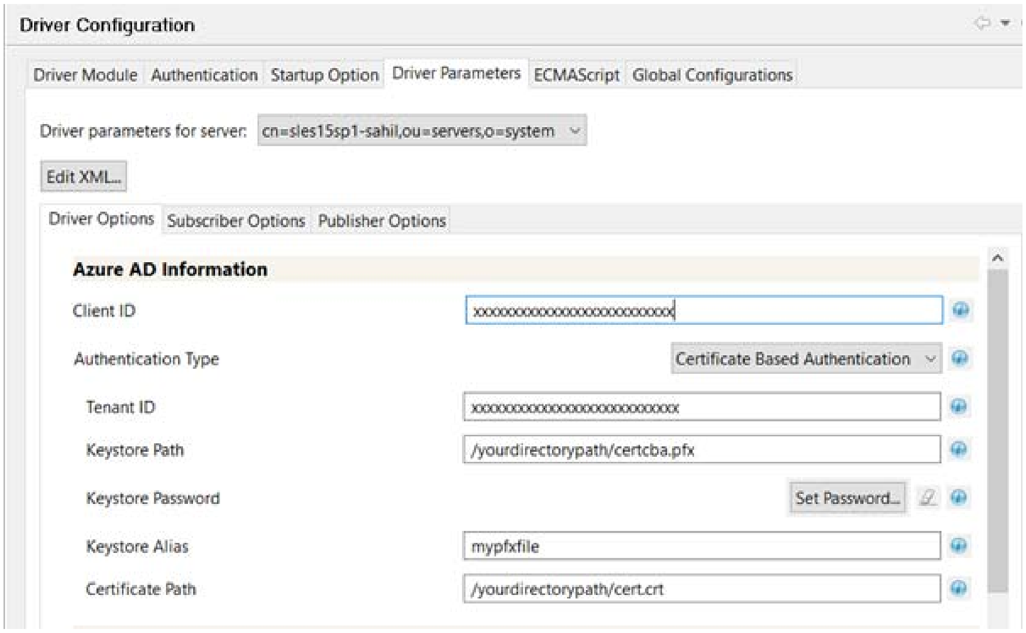
5c1 Tenant ID

5c2 Keystore Path (Ex: /yourdirectorypath/certcba.pfx) - is the absolute location to keystore file created in Step 4.

5c3 Keystore Password: the password that you provided in step 4 for generating the pfx certificate.

5c4 Certificate Path (Ex: /yourdirectorypath/cert.crt) - is the absolute location to the certificate created in Step 3.

5c5 Alias: the value of the parameter "-name" that you provided in step 4.



Driver parameters for server: cn=sles15sp1-sahil,ou=servers,o=system

Azure AD Information

Client ID: xxxxxxxxxxxxxxxxxxxxxxxxxx

Authentication Type: Certificate Based Authentication

Tenant ID: xxxxxxxxxxxxxxxxxxxxxxxxxx

Keystore Path: /yourdirectorypath/certcba.pfx

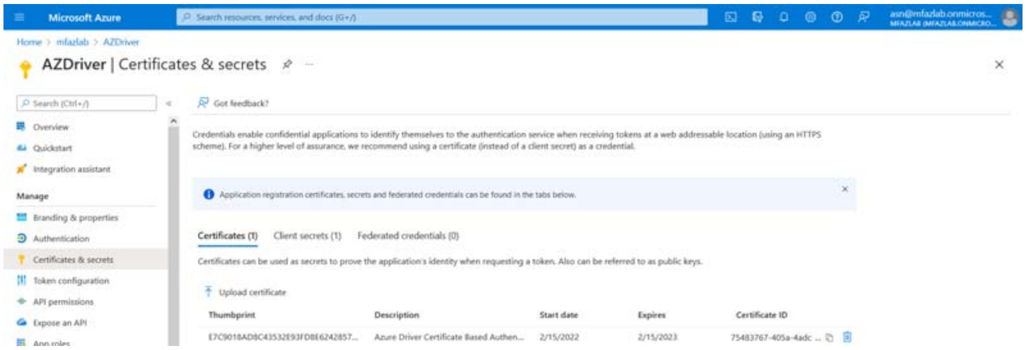
Keystore Password

Keystore Alias: mypfxfile

Certificate Path: /yourdirectorypath/cert.crt

5c6 Uploading certificate to Azure AD and configure the driver parameters.

Login to the Azure portal. In the Application menu blade, click on the Certificates & secrets, in the Certificates section, upload certificate generated in Step 3 (the crt file).



Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Application registration certificates, secrets and federated credentials can be found in the tabs below.

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

Thumbprint	Description	Start date	Expires	Certificate ID
E7C9018AD8C43532E93FDBE6242857...	Azure Driver Certificate Based Authen...	2/15/2022	2/15/2023	75483767-405a-4adc ...

Creating a New Driver Object

You install the Azure AD driver files on the server where you want to run the driver, and then proceed to create the driver in the Identity Vault. You create the Azure AD driver by installing the driver packages or importing the basic driver configuration file and then modifying the driver configuration to suit your environment.

Creating the Driver Object in Designer

An administrator must use Designer to create the Azure AD driver.

The Azure AD driver is package-enabled and you must install and deploy the driver using Designer. You can use iManager to view and edit the deployed configuration. NetIQ recommends that you use Designer to perform any changes.

Importing the Driver Packages

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

1. Open Designer.
2. In the toolbar, click **Help** > **Check for Package Updates**.
3. Click **OK** to update the packages or click **OK** if the packages are up-to-date.
4. Right-click **Package Catalog** and then select **Import Package**.
5. Select any Azure AD driver packages.

   or

   Click **Select All** to import all of the packages displayed.
6. Click **OK** to import the selected packages, then click **OK** in the successfully imported package message.

After the packages are imported, continue with Installing the Driver Packages.

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

   Alternatively, you can drag and drop the Azure AD driver icon from the Cloud section of the Designer palette.
3. In the **Driver Configuration** wizard, select the `Azure AD Base` package from the list of base packages, then click **Next**.
4. Select the optional features to install for the Azure AD driver. All options are selected by default. The options are:
   - **Default Configuration:** This package contains the default configuration information for the driver. Always leave this option selected.

     NOTE: The **Azure AD Default** package and **Azure AD Exchange Default** package are included in **Default Configuration** package. By default, the **Azure AD Exchange Default** package is not selected. Select this package if you plan to use the Identity Manager Exchange Service.
   - **Entitlements and License Support:** This package contains configuration information and policies for synchronizing user accounts, group membership, roles, exchange roles, licenses, SKU, Teams and Channels. If you want to enable account creation and auditing through entitlements, verify that this option is selected.

     To enable the hybrid mode, select the **Azure AD Hybrid Entitlements** package. In this mode, the driver supports only Roles and License entitlements.
   - **Exchange Role Support:** This package contains configuration information and policies for synchronizing exchange roles. Ensure that this package is selected.
   - **Password Synchronization:** This package contains the policies that enables the driver to synchronize passwords. If you want to synchronize passwords, ensure that this package is selected.
   - **Account Tracking:** This package contains the policies that enable you to track accounts for reports. If you are using Identity Reporting, ensure that this package is selected.
   - **Data Collection:** This package contains the policies that enables the driver to collect data for reports. If you are using the Identity Manager Reporting Module, ensure that this package is selected.
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. Click **OK** to install any additional package dependencies.

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

9 On the **Driver Configuration** page, fill in the following fields to configure the driver:

Authentication ID: Provide the authentication information while configuring the domain connections for the driver. This is a qualified `userprincipalname` on Azure AD with login permissions. For example, `admin@domain.onmicrosoft.com`

Password: Specify the password for the driver to authenticate to Azure AD.

Driver Options: To view the driver options, select **Show**.

Client ID: Specify the account name which the driver will use to access the Azure AD applications.

Client Secret: Specify the Client Secret for the given Client ID.

Tenant ID: Specify the Tenant ID for the hosted application.

NOTE: (Optional) If you require National cloud deployment service:

National Cloud Deployment: Set to Yes.

MS Graph API Base URL: Specify Base URL for the driver using Microsoft Graph API.

MS Graph API Token URL: Specify Token URL for the driver using Microsoft Graph API.

Azure AD B2C Tenant: Set to No.

NOTE: (Optional) If you require B2C service:

National Cloud Deployment: Set to No.

Azure AD B2C Tenant: Set to Yes.

NOTE: (Optional) If you require standard service:

National Cloud Deployment: Set to No.

Azure AD B2C Tenant: Set to No.

Authentication Type: Select any one of the following options of the Authentication mechanism from the drop-down menu.

- Certificate based Authentication - Refer to Step 5c of "Certificate Based Authentication in Azure AD Driver 5.1.6" on page 61.
- Client Secret Authentication

NOTE: You created Client ID and Client Secret while creating a proxy application in Azure AD. For more information, see "Creating a Proxy Application on Azure AD" on page 82.

Show Schema Extensions Configuration: To show the schema extensions configuration options for the application (Azure AD), select **Show**.

Enable Hybrid Operation Mode: In hybrid mode, the driver provisions only roles, licenses, teams and channels while the users and groups are provisioned by the AD driver. By default, the parameter is set to **Yes**. If you want to run the driver in normal mode, set the option to **No**.

Activate Azure Directory Roles: By default, the driver obtains the roles that have been pre-activated in Azure Directory. If you want the driver to activate all Azure Directory roles, set this option to **Yes**. This fetches all the activated roles in Identity Applications. These roles are also available at the driver startup. Roles activation is one time activity and need not be performed again.

To obtain only pre-activated roles, leave the setting unchanged.

Existing Schema Extensions: To retain the previously-loaded configuration from Azure AD, select **Preserve**. To remove existing configuration, specify **Remove**.

Add a schema extension: Specify appropriate configuration details while adding a schema extension. You can add multiple schema extensions if required.

- **Name of extension:** Specify the name of the schema extension. For example, `Title`.

  If you create multiple schema extensions with the same name, the driver uses the first extension in the list and ignores the remaining extensions that have the same name.
- **Type of extension:** Specify the data type for the configured schema extension. Ensure that the data type is a supported schema extension type in Azure AD.
- **Target objects of extension:** Lists the target objects for the schema extension. A schema extension can be extended to multiple target object classes. For example, if you have a schema extension called `Title`, it can be extended to a `User` and `Group` object classes.

NOTE: After adding the schema extension attribute, add the application attribute name to the driver filter in the following format:

```
extension_<client_id>_<attribute_name>
```

where `<client_id>` indicates the client ID that is used by the driver to connect to Azure AD.

For example, `extension_4691ac9cbee390e6e8e_Title`.

Subscriber Options: To view the Subscriber options, select **Show**.

Truststore file: Specify the name and path of the truststore file containing the trusted certificates used when a remote server is configured to provide server authentication. This file will contain certificates for Azure Graph and Exchange Service. For example, `c:\security\truststore`.

Proxy Host and Port: When an HTTP proxy is used, specify the host address and the host port. For example, 192.10.1.3:18180. Otherwise, leave the field blank.

Exchange and Powershell Service: When Identity Manager Exchange Service is enabled, the driver synchronizes Exchange users and groups using this service.

Exchange Authentication Type:

- Certificate Based Authentication - Refer to "Creating Client Certificate for IDM Exchange Service" on page 63
- Client Based Authentication

Exchange Service URL: Specify the URL of the Identity Manager Exchange Service.

For example, `https://<ip-addr>:<port>/ExchServer`.

Office 365 Exchange Online: To initiate a connection with Exchange Online and synchronize Office 365 exchange users and groups, select **Yes**.

Queue Operations: To enable queuing of objects when synchronizing between Azure AD and Identity Manager Exchange Service, select **True**.

Page Size: Set a value for the number of results displayed per page during Exchange Publisher poll.

Trace location: Specify the custom path where you want to save the Identity Manager Exchange Service logs. By default, the logs will be saved in this component's installation directory.

Trace Level: Set the trace level for the Identity Manager Exchange Service.

The driver supports five trace levels: NOTIFY, INFO, ERROR, MORE INFO, and DEBUG. The default trace level is NOTIFY. The next trace level, that is, INFO provides basic trace messages. ERROR provides some additional information than the previous level. Detailed messages are logged if you select INFO. DEBUG logs information on debugging data along with detailed messages.

Trace File Size Limit: Specify the trace file size limit in MB. The minimum value is 10 MB.

Database Password: Specify the database password. The driver uses this password to encrypt and connect to the Publisher cache. Ensure that the same password is used to reconnect to the cache at a later time.

Group cache clear interval: Specify the value as required to clear the exchange group related cache data. You must specify a numerical value (for example, -1, 0, 1, 2, etc.) in the Group cache clear interval field. The numerical value specified here corresponds to hours.For more information, see "Group Cache Clear Interval" on page 20.

Publisher Options: To view the Publisher options, select **Show**.

Enable Publisher: Allows you to enable or disable the Publisher connection for the driver.

Publisher Polling Interval: Specify a time period after which the driver should query Azure AD for new changes. The time is specified in minutes.

Heart Beat Interval: Allows the driver to send a periodic status message on the Publisher channel when there is no traffic for a specific duration. This indicates the time period at which the heart beat document is issued by the driver shim. The time is specified in minutes.

10 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 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 running. 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) 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.

11 On the **Azure AD Base** page, fill in the following fields, then click **Next**:

- **Domain Name:** Specify the Azure AD domain site context. For example, <*domain name*>.onmicrosoft.com or <*domain name*>.com format.
- **Identities to be synchronized:** Specify whether the driver should synchronize identities from AD or configure the Identity Vault to act as the identity provider.

  If you choose to configure the Identity Vault as an identity provider, association to any other directory is not required.

  When you choose to synchronize identity from AD, you can synchronize only users that have an association with AD. If you are using the driver in hybrid mode, select only **AD** option. This enables the driver to synchronize the identities from the Identity Vault to AD from where the identities will be synchronized to Azure AD cloud through Azure AD Connect.
- **Usage Location:** Specify the two letter country code for the user availing the Office 365 services.

12 (Conditional) On the **Install Azure AD Managed System Information** page, fill in the following fields to define the ownership of Azure AD, then click **Next**:

General Information

- **Name**: Specify a descriptive name for the managed system.
- **Description**: Specify a brief description of the managed system.
- **Location**: Specify the physical location of the managed system.
- **Vendor**: Select the vendor of the managed system.
- **Version**: Specify the version of the managed system.

System Ownership

- **Business Owner** - Select a user object in the Identity Vault that is the business owner of Azure AD. 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 Azure AD. This can only be a user object, not a role, group, or container.

  This page is only displayed if you selected to install the Data Collection packages and the Account Tracking packages.

System Classification

- **Classification**: Select the classification of Azure AD. This information is displayed in the reports. The options are as follows:
  - Mission-Critical
  - Vital
  - Not-Critical

    - Other

      If you select Other, you must specify a custom classification for Azure AD.

  - **Environment**: Select the type of environment Azure AD provides. The options are as follows:
    - Development
    - Test
    - Staging
    - Production
    - Other

      If you select Other, you must specify a custom environment for Azure AD.

13. On the Azure AD Password Synchronization page, fill in the following fields, then click Next:
    - **Set Password Never Expires:** If you set this option to True on the newly created users, the password does not expire for those users.
    - **Disable Force Change Password at First Login:** If you set this option to True, a user is not prompted to change the password when the user logs in to Azure AD for the first time.
    - **Set Strong Password Required:** If you set this option to True, the user needs to set a strong password.
14. On the Account Tracking page, specify the name of the realm, security domain, or namespace in which the account name is unique. You must set the Realm to the Azure AD domain name.
15. On the Confirm Installation Tasks page, review the summary of tasks and click Finish.

The driver is now created. You can modify the configuration settings by Configuring the Driver.

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.

To edit the properties, perform the following steps:

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

   NOTE: These settings are only displayed if you installed the Entitlements package. If you selected the Azure AD Hybrid Entitlements package, only Roles, License, Teams and Channel Entitlements are supported with this 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 **False**, user and group membership synchronization is managed using the non-entitlement configuration method.

  NOTE: Using Group Membership Entitlement you can add or remove members from SharePoint Online.

- **Use License Entitlement:** Ensure the value of this parameter is set to true to enable the driver to manage licenses using the License entitlement. By default, the value is set to **True**.
- **SKU Entitlement:** Ensure the value of this parameter is set to true to enable the driver to manage SKU Subscription assignments based on the entitlement. By default, the value is set to **True**.
- **Use Roles Entitlement:** Ensure the value of this parameter is set to true to enable the driver to manage roles using the Roles entitlement. By default, the value is set to **True**.
- **Teams Entitlement:** Ensure the value of this parameter is set to true to enable the driver to manage roles using the Teams entitlement. By default, the value is set to **True**.
  - Add User as Owner to Team - Select “Yes” to add the User as “Owner” to Team.
- **Channels Entitlement:** Ensure the value of this parameter is set to true to enable the driver to manage roles using the Channels entitlement. By default, the value is set to **True**.
  - Add User as Owner to Channel - Select “Yes” to add the User as “Owner” to Channel.
  - Team and Channel Name Separator - Specify the character to separate Team and Channel name.

    For example: Team::Channel

5. Click **Apply**.
6. Modify any other settings as necessary.

   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 Azure AD, your synchronization requirements for the driver might differ from the default policies. If this is the case, you require customization.
7. Click **OK** when finished.
8. Continue with “Deploying the Driver” on page 94.

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, 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 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, then select **Live** > **Start Driver**.

Creating the Driver Object in Designer

An administrator must use Designer to create the Azure AD driver.

The Azure AD driver is package-enabled and you must install and deploy the driver using Designer. You can use iManager to view and edit the deployed configuration. NetIQ recommends that you use Designer to perform any changes.

Activating the Driver

The Identity Manager driver for Office 365 and Azure AD is part of the Identity Manager Integration Module for Microsoft Enterprise.

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

Updating the Driver Files

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

To update the driver files perform the following steps:

1. Stop the driver instance by using iManager, Designer, or dxcmd by performing one of the following actions:
   - If the driver is running locally, stop the driver instance and the Identity Vault.
   - If the driver is running with a Remote Loader instance, stop the driver and the Remote Loader instance.

   For example, go to a command prompt on Linux and run `ndsmanage stopall`
2. Download the driver patch file to a temporary folder on your server.
3. Extract the contents of the driver patch file.
4. For Linux, open a command prompt and run the following command to upgrade the existing RPM:

   ```
   rpm -Uvh <Driver Patch File Temporary Location>/linux/netiq-
   DXMLRESTAzure.rpm
   ```
5. For Windows, perform the following actions:

   5a Navigate to the <*Extracted Driver Patch File Temporary Location*>`\windows` folder.

   5b Copy the following jars to the folder `<IdentityManager installation>\NDS\lib` or `<IdentityManager installation>\RemoteLoader\<architecture>\lib`.
   - `AZDriverShim.jar`
   - `RestLib.jar`
   - `OData.jar`
6. (Conditional) If the driver is running locally, start the Identity Vault and the driver instance.

For example, open a command prompt on Linux and run `ndsmanage startall`

7 (Conditional) If the driver is running with a Remote Loader, start the Remote Loader and the driver instance.

3 Installing Azure AD 5.1.6

The Identity Manager Driver for Office 365 and Azure Active Directory (Azure AD driver) allows you to seamlessly provision and deprovision users, group memberships, exchange mailboxes, roles, and licenses to Azure AD cloud. You can also configure the driver to integrate with Identity Manager Service for Exchange Online (Identity Manager Exchange Service) for synchronizing Office 365 attributes.

NOTE: Microsoft has announced the retirement of Azure AD and Msol PowerShell modules. MS has recommended using Microsoft graph based PowerShell cmdlets instead. For more information, see: Upgrade from Azure AD PowerShell to Microsoft Graph PowerShell (https://learn.microsoft.com/en-us/powershell/microsoftgraph/migration-steps?view=graph-powershell-1.0).

- "Preparing for Installation" on page 49
- "Certificate Based Authentication support for Identity Manager Exchange Service" on page 63
- "Creating a New Driver Object" on page 65
- "Updating the Driver Files" on page 73

Preparing for Installation

Prerequisites

This section provides the prerequisites, considerations, and system setup needed to install the driver:

- "Prerequisites for the Driver" on page 49
- "Prerequisites for the User Account to be Configured in the Driver" on page 50
- "Prerequisites for Identity Manager Exchange Service" on page 50
- "Prerequisites for OAuth 2.0" on page 51
- "Assigning the Rights to the Application" on page 54
- "Prerequisites for Support of Modern Authentication" on page 55

Prerequisites for the Driver

The driver requires the following applications:

- Identity Manager 4.8.4 or later
- Identity Manager Designer 4.8.4 or later
- Identity Manager REST driver 1.1.2.0400 or later

Prerequisites for the User Account to be Configured in the Driver

You must ensure that the user account you are configuring in the driver has the following Roles or Permissions in the Azure application:

- At a minimum, the user account must have the following roles:
  - User Administrator
  - Exchange Administrator
  - Privileged Role Administrator
  - Teams Administrator
- The multi-factor authentication for the user account is disabled.
- A user account has conditional access. For more information on providing conditional access, see What is Conditional Access.

NOTE: You may need to provide additional roles based on their requirement.

Prerequisites for Identity Manager Exchange Service

- Microsoft Windows Server 2016, or Microsoft Windows Server 2019.
- Install .NET Framework 4.7.2 or later.
- Microsoft Visual C++ 2017 Redistributable packages for Visual Studio.

  Download the packages from the Microsoft Downloads website.
- Upgrade to PowerShell 5.1 or later.
- Update PowerShellGet to the latest version using Install-Module PowerShellGet -Force-5
- Install the ExchangeOnlineManagement and Microsoft Graph modules by executing the below commands in PowerShell:
  - `Install-Module -Name ExchangeOnlineManagement`
  - `Install-Module Microsoft.Graph -Scope AllUsers`

  NOTE: If you get “Unable to resolve package source” error, then run `[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12` command before running the Install-Module commands.
- Windows Azure AD Module for Windows Powershell on the computer where you will install Windows Powershell service.

  Perform the following steps to upgrade PowerShell to the latest version:

  1. Open a Windows PowerShell console.
  2. Run the following Install-Module cmdlet or Install-Script cmdlet:
     - If it is a module: `Install-Module -Name <moduleName> -RequiredVersion <version>`

For example, `Install-Module -Name MSOnline`. Refer to www.powershellgallery.com (https://www.powershellgallery.com/packages/MSOnline/1.1.183.66)

- If it is a script: `Install-Script -Name <scriptName> -RequiredVersion <version>`

Identity Manager Exchange Service can be run on a user configured port. However, the service cannot be used with any other REST client tools.

Prerequisites for OAuth 2.0

The driver uses OAuth 2.0 protocol to authenticate to Azure AD. To support this protocol for authentication, you need to have a proxy application for the Azure AD driver on Azure AD. The Client ID and Client Secret allotted to the application will be later used in the Azure AD driver configuration. For more information about Azure Active Directory Application Proxy, see Microsoft Azure documentation.

Creating a Proxy Application on Azure AD

A proxy application is created in the Azure Portal. Creating a proxy application involves the following steps:

1. Registering an application and obtaining a client ID. For more information see, Registering an Application.
2. Generating an application password or the client secret. For more information see, Certificates and Secrets (https://learn.microsoft.com/en-us/entra/identity-platform/quickstart-register-app#add-credentials).
3. Configuring API permissions (Delegated and Application permissions). Set the delegated and application permissions as shown in the following table. For more information see, Add permissions to access web APIs.

Table 3-1 *List of Application type of APIs*

API	Type	Description	Admin Consent
Application.Read.All	Application	Read all applications	Grant Admin Consent
Application.ReadWrite.All	Application	Read and write all applications	Grant Admin Consent
AuditLog.Read.All	Application	Read all audit log data	Grant Admin Consent
Channel.Create	Application	Create channels	Grant Admin Consent
Channel.Delete.All	Application	Delete channels	Grant Admin Consent
Channel.ReadBasic.All	Application	Read the names and descriptions of all channels	Grant Admin Consent
ChannelMember.Read.All	Application	Read the members of all channels	Grant Admin Consent

API	Type	Description	Admin Consent
ChannelMember.ReadWrite.All	Application	Add and remove members from all channels	Grant Admin Consent
ChannelSettings.Read.All	Application	Read the names, descriptions, and settings of all channels	Grant Admin Consent
ChannelSettings.ReadWrite.All	Application	Read and write the names, descriptions, and settings of all channels	Grant Admin Consent
Device.Read.All	Application	Read all devices	Grant Admin Consent
Device.ReadWrite.All	Application	Read and write devices	Grant Admin Consent
Directory.Read.All	Application	Read directory data	Grant Admin Consent
Directory.ReadWrite.All	Application	Read and write directory data	Grant Admin Consent
Domain.ReadWrite.All	Application	Read and write domains	Grant Admin Consent
Group.Create	Application	Create groups	Grant Admin Consent
Group.Read.All	Application	Read all groups	Grant Admin Consent
Group.ReadWrite.All	Application	Read and write all groups	Grant Admin Consent
GroupMember.Read.All	Application	Read all group memberships	Grant Admin Consent
GroupMember.ReadWrite.All	Application	Read and write all group memberships	Grant Admin Consent
Team.Create	Application	Create teams	Grant Admin Consent
Team.ReadBasic.All	Application	Get a list of all teams	Grant Admin Consent
TeamMember.Read.All	Application	Read the members of all teams	Grant Admin Consent
TeamMember.ReadWrite.All	Application	Add and remove members from all teams	Grant Admin Consent
TeamMember.ReadWrite NonOwnerRole.All	Application	Add and remove members with non-owner role for all teams	Grant Admin Consent
TeamSettings.Read.All	Application	Read all teams' settings	Grant Admin Consent
TeamSettings.ReadWrite.All	Application	Read and change all teams' settings	Grant Admin Consent
User.Read.All	Application	Read all users	Grant Admin Consent
User.ReadWrite.All	Application	Read and write all users' full profiles	Grant Admin Consent

API	Type	Description	Admin Consent
UserAuthenticationMethod.Read.All	Application	Read all user authentication methods	Grant Admin Consent
UserAuthenticationMethod.ReadWrite.All	Application	Read and write all users authentication methods	Grant Admin Consent

Table 3-2 List of Delegated type of APIs

API	Type	Description	Admin Consent
AuditLog.Read.All	Delegated	Read audit log data	Grant Admin Consent
Directory.AccessAsUser.All	Delegated	Access directory as the signed in user	Grant Admin Consent
Directory.Read.All	Delegated	Read directory data	Grant Admin Consent
Directory.ReadWrite.All	Delegated	Read and write directory data	Grant Admin Consent
Group.Read.All	Delegated	Read all groups	Grant Admin Consent
Group.ReadWrite.All	Delegated	Read and write all groups	Grant Admin Consent
GroupMember.Read.All	Delegated	Read group memberships	Grant Admin Consent
GroupMember.ReadWrite.All	Delegated	Read and write group memberships	Grant Admin Consent
RoleManagement.Read.All	Delegated	Read role management data for all RBAC providers	Grant Admin Consent
RoleManagement.Read.Directory	Delegated	Read directory RBAC settings	Grant Admin Consent
UserAuthenticationMethod.Read	Delegated	Read user authentication methods	Grant Admin Consent
UserAuthenticationMethod.Read.All	Delegated	Read all user’s authentication methods	Grant Admin Consent
UserAuthenticationMethod.ReadWrite	Delegated	Read and write user authentication methods	Grant Admin Consent
UserAuthenticationMethod.ReadWrite.All	Delegated	Read and write all user authentication methods	Grant Admin Consent
User.Read	Delegated	Sign in and read user profile	Grant Admin Consent
User.Read.All	Delegated	Read all users' full profiles	Grant Admin Consent
User.ReadBasic.All	Delegated	Read all users' basic profiles	Grant Admin Consent
User.ReadWrite	Delegated	Read and write access to user profile	Grant Admin Consent

API	Type	Description	Admin Consent
User.ReadWrite.All	Delegated	Read and write all users' full profiles	Grant Admin Consent

NOTE: You may need to provide additional permissions based on their requirement.

The Client ID and Client Secret can now be used for driver configurations or any other REST clients.

Assigning the Rights to the Application

1. In the server where you have installed the exchange service, login to PowerShell and connect to the Office 365 Exchange Online service, using the following command:

   ```
   Connect-MSolService
   ```

2. To obtain the Client ID for your application, replace <AppPrincipalId> with the Client ID that you obtained from Creating a Proxy Application on Azure AD and run the following commands in PowerShell.

   ```
   Get-MsolServicePrincipal | ft DisplayName, <AppPrincipalId> -AutoSize
   ```

   ```
   $ClientIdWebApp = '<AppPrincipalId>'
   ```

   ```
   $webApp = Get-MsolServicePrincipal –AppPrincipalId $ClientIdWebApp
   ```

3. Assign the Company Administrator rights to your application using the Client ID obtained in Step 2 by running the following command:

   ```
   Add-MsolRoleMember -RoleName "Company Administrator" -RoleMemberType
   ServicePrincipal -RoleMemberObjectId $webApp.ObjectID
   ```

   The Company Administrator role will give you rights to delete the directory objects.

   IMPORTANT: The Company Administrator role in Identity Manager is mapped to the Global Administrator role in Azure AD.

Ensure that the account used by the driver to connect to the Exchange Online service has the correct roles to load and execute the following cmdlets:

- New-Mailbox
- Set-Mailbox
- Get-Mailbox
- Remove-Mailbox
- New-MailUser
- Set-MailUser
- Get-MailUser
- Remove-MailUser
- Set-User
- Get-User

- New-DistributionGroup
- Set-DistributionGroup
- Set-Group
- Get-DistributionGroup
- Get-Group
- Remove-DistributionGroup
- Add-DistributionGroupMember
- Remove-DistributionGroupMember
- Get-DistributionGroupMember
- Add-RoleGroupMember
- Remove-RoleGroupMember
- Get-RoleGroupMember
- New-UnifiedGroup
- Get-UnifiedGroup
- Set-UnifiedGroup
- Remove-UnifiedGroup
- Add-UnifiedGroupLinks
- Remove-UnifiedGroupLinks
- Get-UnifiedGroupLinks
- Get-MsolUser
- Set-MsolUser

Absence of the required roles prevents the driver from executing the cmdlets that require those roles.

Prerequisites for Support of Modern Authentication

As Microsoft Office 365 is deprecating the **Basic** authentication, you must now configure the driver with modern authentication method. You must also ensure to have the earlier mentioned prerequisites ("Prerequisites for the Driver" on page 80, "Prerequisites for Identity Manager Exchange Service" on page 81, and "Prerequisites for OAuth 2.0" on page 81) met, and then proceed with the following prerequisites.

The following prerequisites are specific to modern authentication. It is highly recommended to upgrade the driver version 5.1.x to 5.1.3 or later to support modern authentication.

Installing the Microsoft Exchange Online PowerShell V2 (EXO V2)

You must install the Microsoft Exchange Online PowerShell V2 module to support the new API's. For more information on EXO V2 module, see About the Exchange Online PowerShell V2 module.

- For prerequisites to install the EXO V2 module, see Prerequisites for EXO V2 module.
- For installing the EXO V2 module, see Install the EXO V2 module.

Configuring Azure AD Proxy Application for Modern Authentication Methods

You must enable the permission in the Azure portal to access Microsoft Office 365 with modern authentication.

NOTE: Before upgrading to 5.1.6 ensure to run this command on the server running the Identity Manager Exchange Service using PowerShell.

The procedure to set the permission is shown below:

1 Login to the Azure AD Portal.

2 Select **Azure Active Directory**.

3 Navigate to **App Registration >** find and select your application in the list (for example: *<MySample_Azure_Appln>*) **> Authentication > Advanced Settings**.

4 Set **Enable the following mobile and desktop flows** under **Allow public client flows** permission to **Yes**.

> **IMPORTANT:** The multi-factor authentication (MFA) must be disabled for the Azure account which is used with the driver.

Installing the Driver and the Identity Manager Exchange Service

You can install the Azure AD driver on the Identity Manager server or with the Remote Loader. The driver installation program guides you through the driver and the Identity Manager Exchange Service installation.

NOTE:

1. IDM Exchange service must be run on the same machine as the driver and configured to listen only on local host.
2. IDM Exchange service must be run with least privilege required for the configured `PowerShell cmdlets` to execute.
3. Only system administrator must be provided access to the IDM exchange service machine

Perform the following actions to install and configure the Exchange Service:

1 Copy Exchange Service from `[ISO]:\products\IDM\windows\setup\drivers\azuread\ExchangeService` to any local drive on the server you intend to run this service.

2 Navigate to the directory where you copied the ExchServerHost.exe in the first step.

For example, `C:\ExchangeService`

3 Run the following command to install the Exchange Service:

```
<location of InstallUtil>\InstallUtil.exe ExchServerHost.exe
```

For example:

```
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\InstallUtil.exe
ExchServerHost.exe
```

where, `C:\Windows\Microsoft.NET\Framework64\v4.0.30319\InstallUtil.exe` is the location where `InstallUtil.exe` is located.

4. Ensure the server certificate is available in iManager. To create the server certificate, see "Securing Communication with Identity Manager Exchange Service" on page 76

5. Open cmd prompt, and navigate to the local drive location where the `ExchangeService` is saved, as mentioned in Step 1 on page 76 (`\products\IDM\windows\setup\drivers\azuread\ExchangeService\`), and execute the `configureExchService` batch file as explained below.

NOTE: When upgrading the driver from 5.1.5 to 5.1.6 in the Identity Manager Exchange Server, you need to replace ExchServerHost.exe,IDMExchServer.dll files and configureExchService.bat files located in `\ExchangeService` folder.

While installing the Identity Manager Exchange Service for the first time using the port and certificate, you need to add an additional argument **AuthenticationType**. This argument refers to whether you want to use the Basic Authentication or Certificate Based Authentication. The certificate friendly name provided in the command should be the nickname (exchcba) of the certificate to be created for the Identity Manager Exchange Service.

Example for Basic Authentication Command:

```
configureExchService.bat 9001 exchcba 0
```

Example for Certificate Based Authentication Command:

```
configureExchService.bat 9001 exchcba 5
```

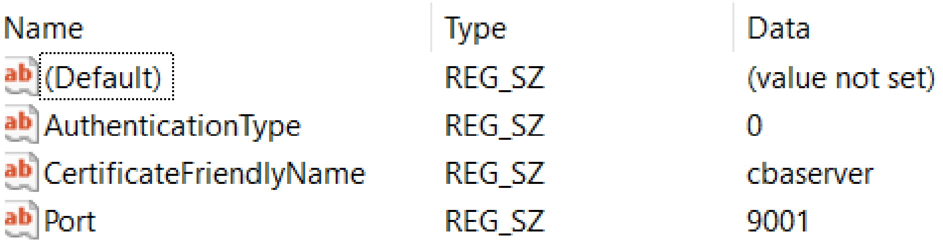
There is another way of choosing the authentication type via Registry Editor as explained below.

Goto Registry Editor and navigate to **HKEY_LOCAL_MACHINE** -> **Software** -> **Novell** -> **ExchServer**

You can edit the value of the key **AuthenticationType** as required.

Below is the snapshot of the Registry Editor where only 3 keys are present after upgrading to 5.1.6.

Figure 3-1

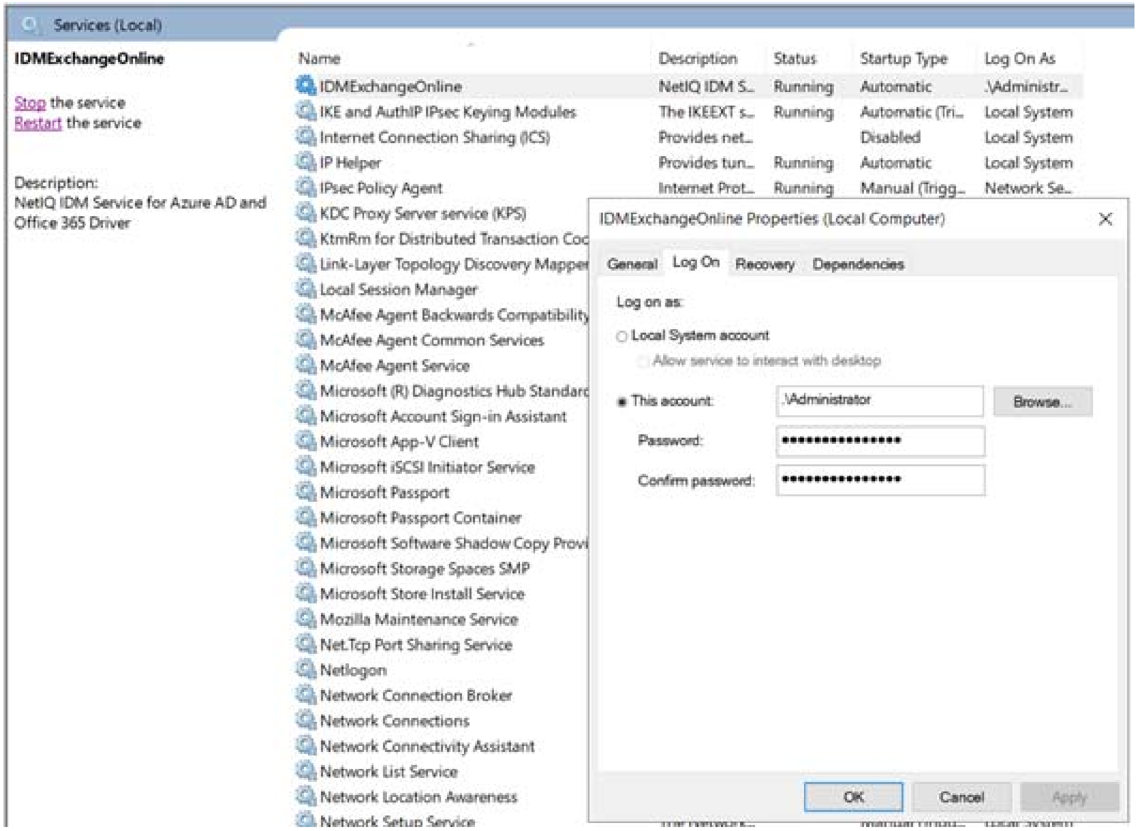


Name	Type	Data
(Default)	REG_SZ	(value not set)
AuthenticationType	REG_SZ	0
CertificateFriendlyName	REG_SZ	cbaserver
Port	REG_SZ	9001

NOTE: When you choose Basic Authentication in **iManager**, go to the **Identity Manager Exchange Server** and edit the value of **AuthenticationType** in the registry to 0. Similarly when you choose Certificate Based Authentication in **iManager**, then go to the **Identity Manager Exchange Server** and edit the value of **AuthenticationType** in the registry to 5.

Apart from this, you need to run the **IDMExchangeOnline** service as an Administrator. Run **Services.msc** and go to the **IDMExchangeOnline**, right click **IDMExchangeOnline** and select **Properties** and under **Log On**, Select "**This account**" and provide the credentials of Local administrator account as shown below.

Figure 3-2



Services (Local)

IDMExchangeOnline

Stop the service
Restart the service

Description:
NetIQ IDM Service for Azure AD and Office 365 Driver

Name	Description	Status	Startup Type	Log On As
IDMExchangeOnline	NetIQ IDM S...	Running	Automatic	.\Administr...
IKE and AuthIP IPsec Keying Modules	The IKEEXT s...	Running	Automatic (Tri...	Local System
Internet Connection Sharing (ICS)	Provides net...		Disabled	Local System
IP Helper	Provides tun...	Running	Automatic	Local System
IPsec Policy Agent	Internet Prot...	Running	Manual (Trigg...	Network Se...
KDC Proxy Server service (KPS)				
KtmRm for Distributed Transaction Coc				
Link-Layer Topology Discovery Mapper				
Local Session Manager				
McAfee Agent Backwards Compatibility				
McAfee Agent Common Services				
McAfee Agent Service				
Microsoft (R) Diagnostics Hub Standarc				
Microsoft Account Sign-in Assistant				
Microsoft App-V Client				
Microsoft iSCSI Initiator Service				
Microsoft Passport				
Microsoft Passport Container				
Microsoft Software Shadow Copy Provi				
Microsoft Storage Spaces SMP				
Microsoft Store Install Service				
Mozilla Maintenance Service				
Net.Tcp Port Sharing Service				
Netlogon				
Network Connection Broker				
Network Connections				
Network Connectivity Assistant				
Network List Service				
Network Location Awareness				
Network Setup Service				

IDMExchangeOnline Properties (Local Computer)

Log on as:

○ Local System account

☐ Allow service to interact with desktop

● This account: .\Administrator

Password: ••••••••••••••••

Confirm password: ••••••••••••••••

NOTE: Windows Local Administrator account is the one that gets created when you install a Windows Server and this account is not part of domain.

6 To start the service, navigate to **Control Panel** > **Administrative Tools** > **Services**.

7 Right-click the **IDMExchangeOnline** service and select **Start**.

NOTE: To uninstall the service, open a .NET command prompt and issue the `InstallUtil /u` ExchServerHost.exe command.

NetIQ recommends you to use TLS 1.1 and TLS 1.2 protocols with the Identity Manager Exchange Service. If you are using ciphers and protocols such as RC4 and Triple DES, or SSLv2/v3 on a server running Identity Manager Exchange Service, you must disable them using the `disableWeakCiphers.reg` file provided in the Exchange Service installation directory. You can either execute the registry file or import the file into Windows Registry. After the changes are made, restart the server. For more information about restricting the use of certain cryptographic algorithms and protocols on Windows servers, see Microsoft Support Site.

Securing Driver Communication

The driver communicates over SSL with Azure AD and Identity Manager Exchange Service.

IMPORTANT: The connection accepts certificates only from a Java keystore. Make sure that the keystore for the certificates is a Java keystore.

The following sections provide instructions for creating a secure connection:

- "Securing Communication with Identity Manager Exchange Service" on page 59
- "Secured Communication with Microsoft Graph API" on page 60
- "Certificate Based Authentication in Azure AD Driver 5.1.6" on page 61

Securing Communication with Identity Manager Exchange Service

To set up SSL between the driver and Identity Manager Exchange Service, you need to create and import a server certificate into the root certificate store of the Windows server where the service is deployed. The following procedure assumes eDirectory as the Certificate Authority (CA).

1 Login to **Identity Console** as an Administrator -> **Certificate Management** -> **Server Certificate Management** -> Click "**+**" icon

   1a Provide the **Nickname** and for creation method select **Custom**.

   For example: Nickname could be "exchcba".

   1b Select **Organizational Certificate Authority**.

   1c Click **Next**.

   1d Under Certificate Parameters, click + for Subject Alternative Names, select **Type** (Choose **IP Address** from the drop-down and **Name**: Enter the IP Address of the Server running Identity Manager Exchange Service). Click **Next**.

   1e Select Your organization's certificate and click **Next**.

   1f Click **OK** until the certificate is successfully created.

2 Export the server certificate created for Identity Manager Exchange Service in step 1a, from the connected eDirectory server and save it to a file in the `pfx` format.

   2a Login to Identity Console, log in to the connected eDirectory server with administrator rights.

   2b Click Certificate Management > Server Certificate Management , then select the Server Certificate that has been created (ex: exchcba) for exchange service in step1.

   2c Click 📤 to export the certificate.

   2d Ensure that by default the Identity Manager Exchange Server Certificate (as created in Step1) has been selected and select **Export Private Key**.

   2e Enter the password and click **OK**.

   2f To save the certificate to a file, click **Save the Exported File**.

3 Import the certificate to the trusted store of the Windows server on which you will run Identity Manager Exchange Service.

   3a Copy the `.pfx` file to the Windows server.

   3b Click **Start** > **Run**> **mmc**.

   3c Click **File** > **Add/Remove Snap-in**.

3d Select **Certificates** and click **Add** to import this snap-in by choosing *Computer account*.

3e Click **Finish**.

3f Navigate to **Certificates** > **Trusted Root Certification Authorities**.

3g Right-click and then select **All Tasks** > **Import**.

3h On the **Welcome to the Certificate Import Wizard** page, click **Next**.

3i Click **Browse** and select the eDirectory certificate you exported in "Export the server certificate from the connected eDirectory server and save it to a file in the `pfx format.`" `on page 76`

3j Specify the password and click **Next**.

3k Click **Finish** to import the certificate into the trust store.

4 Start Identity Manager Exchange Service. For more information, see "Verifying and Starting the Identity Manager Exchange Service" on page 87.

5 Open the following Exchange service URL from your browser:

```
https://<Exchange_Service>:Port/ExchServer
```

6 Obtain the public certificate and import it into the same keystore which was created and placed in IDM Server as mentioned in (for example, the keystore `azuread`).

For example, perform the following steps to obtain a public certificate on Google Chrome:

6a Click 🔒 from the address bar and then click **Details**.

6b In the **Security** tab, click **View Certificate**.

6c In the **Details** tab, click **Copy to File**.

6d In the **Certificate Import Wizard**, click **Next**.

6e Select **DER encoded binary** and click **Next**.

6f Click **Browse** and navigate to the directory where you want to save the certificate.

6g Specify a name for the certificate and click **Next**.

6h Click **Finish** to complete the export.

6i Add the exported key to the driver keystore by using the following Java keytool command:

```
keytool -import -file <path to the exchange cert
file>\<certname.cer> -keystore <mykeystore> -alias <aliasname>
```

NOTE: Ensure that certificates inside the keystore have different alias names for all the imported certificates. Use the existing driver keystore if you are upgrading the driver from 5.1.5 to 5.1.6.

Secured Communication with Microsoft Graph API

To set up SSL between the driver and Azure AD graph REST endpoints, perform the following steps:

1 Open the following URL from your browser:

- `https://login.microsoftonline.com/`
- `https://graph.microsoft.com/`
- `https://azure.microsoft.com/`

2 Obtain the public certificate and import it into the keystore.

For example (Suppose you are accessing https://graph.microsoft.com/), if you are using Google Chrome, perform the following steps:

2a In the address bar, click 🔒 and then click › next to browser address bar (for example:**graph.microsoft.com**).

2b Select **Certificate (Valid)**. The certificate is displayed.

2c Click **Certification Path**. The Certification Path displays the hierarchical structure of the structure of all the certificates.

2d Select the root certificate (the top most parent certificate), and click **View Certificate**. The root certificate is displayed.

2e To save the certificate to your system, click **Details > Copy to File > Next > Next**.

2f Enter a filename for the certificate and save it to a location as required.

2g Add the exported key to the driver keystore using the following Java keytool command:

You might have to create a new keystore(`.jks` file), if one such file doesn't exist already. This keystore file will contain the public certificate of the Azure graph endpoint and the exchange service certificate.

```
keytool -import -file <path to the graph cert file>\<certname.crt> -
keystore <mykeystore> -alias <aliasname>
```

For example: `keytool -import -file msgraph.cer -keystore azuread.jks -alias msgraph`

NOTE:

- Ensure to place the new keystore in IDM Server. In case of Remote Loader place the keystore file in the system where the Azure AD driver is running.
- Ensure that you follow the above steps to import all the certificates into the keystore.

Certificate Based Authentication in Azure AD Driver 5.1.6

1 Generate a password protected private key in PEM format by executing the below commands. You can use an existing password protected private key in PEM format or create a password protected private key using the below command.

```
openssl genrsa -aes256 -out private_key.pem 2048
```

Note: this command will ask you to provide a password for this private key.

2 Generate a certificate signing request using the private key.

```
openssl req -new -key private_key.pem -out cert.csr
```

Note: This command will ask you to input the password for the private key.

Also, this command will ask for a variety of extra information, like company name, country, and a password. None of this is used by the sample, so you can set these values as nothing/anything you want.

3 Generate a x509-certificate using the csr file and the private key

```
openssl x509 -req -days 365 -in cert.csr -signkey private_key.pem -out
cert.crt
```

Note: This command will ask you to input the password for the private key

4 Generate the pfx certificate using the crt file and the private key

```
openssl pkcs12 -export -out certcba.pfx -inkey private_key.pem -in
cert.crt -name "mypfxfile"
```

Note: This command will first ask you to input the password for the private key.

Then it asks you to provide the password for the pfx certificate which will be used as the keystore password in configuring Azure CBA. In the above command the “-name” parameter refers to alias which will be used as an input in configuring Azure CBA.

5 Using Designer, perform the following steps.

5a Right Click Driver Object and click Properties.

5b Goto Driver Configuration -> Driver Parameter

5c Under Driver Options, choose Certificate Based Authentication as authentication type and provide following details.

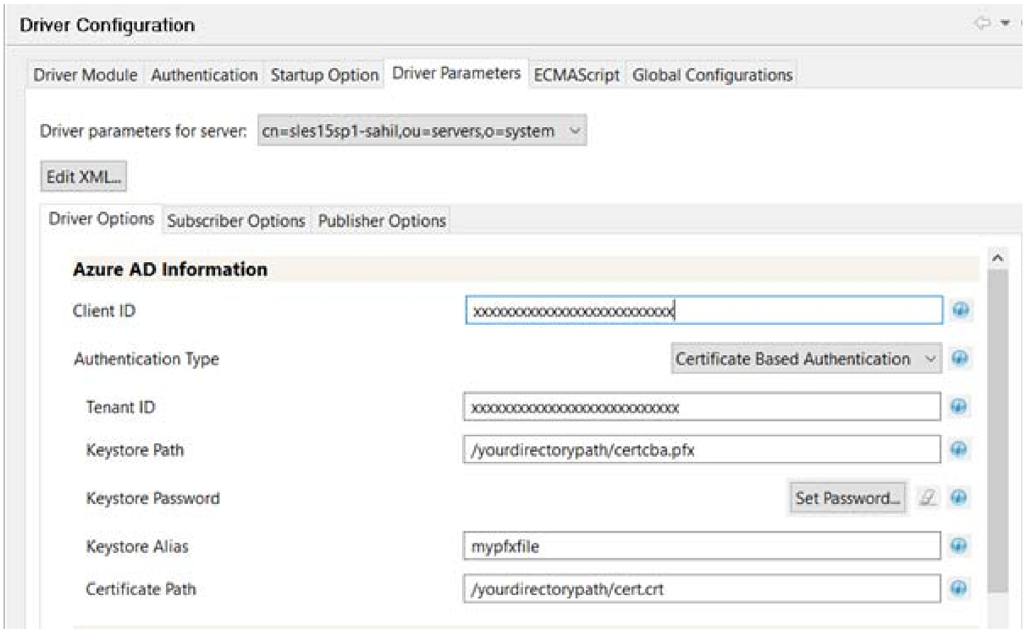
5c1 Tenant ID

5c2 Keystore Path (Ex: /yourdirectorypath/certcba.pfx) - is the absolute location to keystore file created in Step 4.

5c3 Keystore Password: the password that you provided in step 4 for generating the pfx certificate.

5c4 Certificate Path (Ex: /yourdirectorypath/cert.crt) - is the absolute location to the certificate created in Step 3.

5c5 Alias: the value of the parameter “-name” that you provided in step 4.



Driver parameters for server: cn=sles15sp1-sahil,ou=servers,o=system

Azure AD Information

Client ID: xxxxxxxxxxxxxxxxxxxxxxxxxx

Authentication Type: Certificate Based Authentication

Tenant ID: xxxxxxxxxxxxxxxxxxxxxxxxxx

Keystore Path: /yourdirectorypath/certcba.pfx

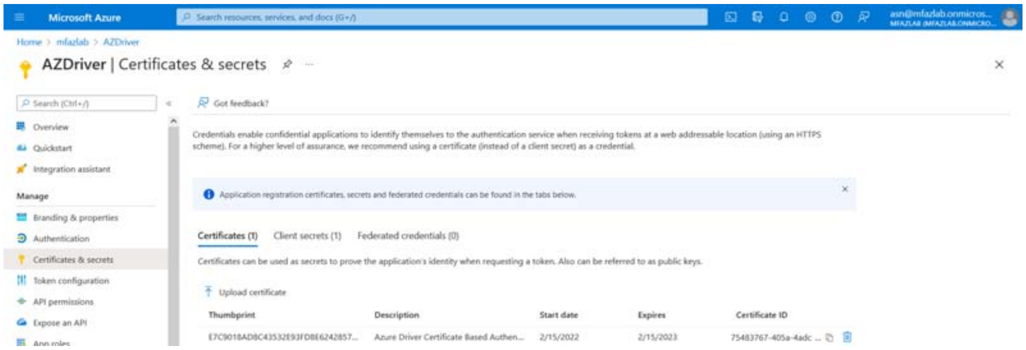
Keystore Password

Keystore Alias: mypfxfile

Certificate Path: /yourdirectorypath/cert.crt

5c6 Uploading certificate to Azure AD and configure the driver parameters.

Login to the Azure portal. In the Application menu blade, click on the Certificates & secrets, in the Certificates section, upload certificate generated in Step 3 (the crt file).



Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Application registration certificates, secrets and federated credentials can be found in the tabs below.

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

Thumbprint	Description	Start date	Expires	Certificate ID
E7C9018AD8C43532E93FD8E6242857...	Azure Driver Certificate Based Authen...	2/15/2022	2/15/2023	75483767-405a-4adc ...

Certificate Based Authentication support for Identity Manager Exchange Service

Creating Client Certificate for IDM Exchange Service

1 Create the client key using the below command.

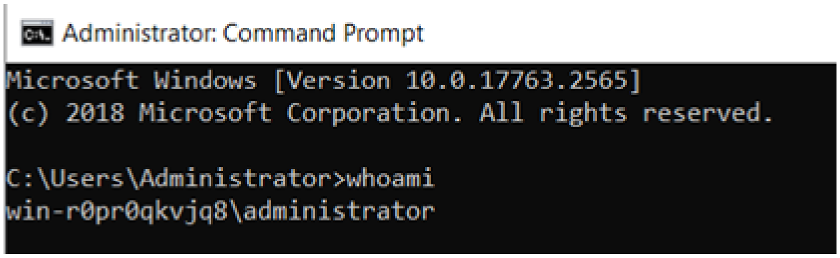
```
openssl genrsa -out client.key.pem 2048
```

2 Create the CSR executing the below command.

```
openssl req -new -key client.key.pem -out client.csr
```

Once the user executes the above command, it is mandatory to input “Common Name” as parameter. The value for the Common Name can be obtained from the Server running Identity Manager Exchange Service by running whoami command in the command prompt as shown below:

Figure 3-3

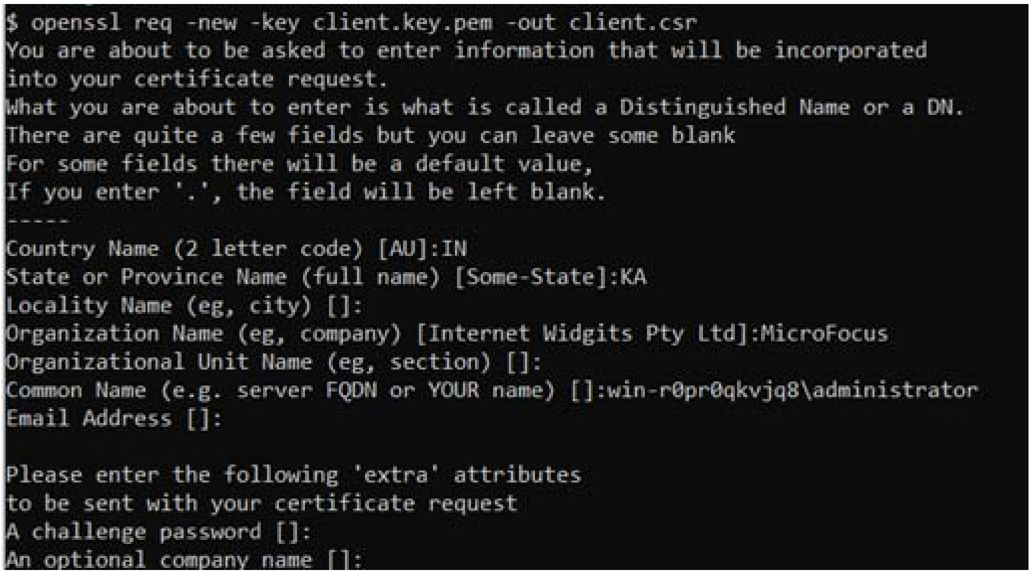


```
Microsoft Windows [Version 10.0.17763.2565]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>whoami
win-r0pr0qkvjq8\administrator
```

Use the whoami value to enter into the Certificate CSR “Common Name” parameter as shown below:

Figure 3-4



```
$ openssl req -new -key client.key.pem -out client.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:KA
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:MicroFocus
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:win-r0pr0qkvjq8\administrator
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Once done, import this CSR in eDirectory under **NETIQ Certificate Server** -> **Issue Certificate** and then click **NEXT** until you get an option to choose `base64` format and download the `base64` certificate.

Generating the Client pfx file

After downloading the Client Certificate in `base64` format, execute the below command to generate the Client Certificate in `pfx` format.

```
openssl pkcs12 -export -out client.pfx -inkey client.key.pem -in client.b64
```

NOTE: Once you execute the above command, you will be prompted to provide the password. This password is required during the driver configuration.

Create a directory in IDM and add the above generated Client Certificate in the IDM directory. Use this directory path in the iManager as given below -

Driver Parameters - > **Subscriber Settings** -> **Exchange and Powershell Service** -> **Exchange Authentication Type** (Certificate Based Authentication) -> **Keytore Path**

Verifying and Starting the Identity Manager Exchange Service

After finishing the installation of Identity Manager Exchange Service, verify that the service is properly installed.

NOTE: Ensure that SSL is configured for the Identity Manager Exchange Service before starting the service. This is a mandatory step before running the service. For more information, see Securing Communication with Identity Manager Exchange Service.

1 From the **Start** menu, type **regedit**.

2 On the **Registry Editor** page, locate the service at **HKEY_LOCAL_MACHINE** > **Software** > **Novell** > **ExchServer** and verify that the **Port** and **CertificateFriendlyName** have the correct values.

   The **CertificateFriendlyName** must be the same as **Certificate Alias** that you specified in Step 1 of the "Securing Communication with Identity Manager Exchange Service" on page 76.

3 Navigate to the services that are running on your server and start the `IDMExchangeOnline` service.

Verifying the Provisioning of Exchange Mailbox

To verify the provisioned mailboxes for users, follow the procedure provided in the Microsoft Exchange Admin Center.

Creating a New Driver Object

You install the Azure AD driver files on the server where you want to run the driver, and then proceed to create the driver in the Identity Vault. You create the Azure AD driver by installing the driver packages or importing the basic driver configuration file and then modifying the driver configuration to suit your environment.

Creating the Driver Object in Designer

An administrator must use Designer to create the Azure AD driver.

The Azure AD driver is package-enabled and you must install and deploy the driver using Designer. You can use iManager to view and edit the deployed configuration. NetIQ recommends that you use Designer to perform any changes.

Importing the Driver Packages

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

1 Open Designer.

2 In the toolbar, click **Help** > **Check for Package Updates**.

3 Click **OK** to update the packages or click **OK** if the packages are up-to-date.

4 Right-click **Package Catalog** and then select **Import Package**.

5 Select any Azure AD driver packages.

   or

   Click **Select All** to import all of the packages displayed.

6 Click **OK** to import the selected packages, then click **OK** in the successfully imported package message.

After the packages are imported, continue with Installing the Driver Packages.

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

   Alternatively, you can drag and drop the Azure AD driver icon from the Cloud section of the Designer palette.
3. In the Driver Configuration wizard, select the `Azure AD Base` package from the list of base packages, then click Next.
4. Select the optional features to install for the Azure AD driver. All options are selected by default. The options are:
   - **Default Configuration:** This package contains the default configuration information for the driver. Always leave this option selected.

     NOTE: The Azure AD Default package and Azure AD Exchange Default package are included in Default Configuration package. By default, the Azure AD Exchange Default package is not selected. Select this package if you plan to use the Identity Manager Exchange Service.
   - **Entitlements and License Support:** This package contains configuration information and policies for synchronizing user accounts, group membership, roles, exchange roles, licenses, SKU, Teams and Channels. If you want to enable account creation and auditing through entitlements, verify that this option is selected.

     To enable the hybrid mode, select the Azure AD Hybrid Entitlements package. In this mode, the driver supports only Roles and License entitlements.
   - **Exchange Role Support:** This package contains configuration information and policies for synchronizing exchange roles. Ensure that this package is selected.
   - **Password Synchronization:** This package contains the policies that enables the driver to synchronize passwords. If you want to synchronize passwords, ensure that this package is selected.
   - **Account Tracking:** This package contains the policies that enable you to track accounts for reports. If you are using Identity Reporting, ensure that this package is selected.
   - **Data Collection:** This package contains the policies that enables the driver to collect data for reports. If you are using the Identity Manager Reporting Module, ensure that this package is selected.
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. Click OK to install any additional package dependencies.
8. On the Driver Information page, specify a name for the driver, then click Next.
9. On the Driver Configuration page, fill in the following fields to configure the driver:

Authentication ID: Provide the authentication information while configuring the domain connections for the driver. This is a qualified `userprincipalname` on Azure AD with login permissions. For example, `admin@domain.onmicrosoft.com`

Password: Specify the password for the driver to authenticate to Azure AD.

Driver Options: To view the driver options, select **Show**.

Client ID: Specify the account name which the driver will use to access the Azure AD applications.

Client Secret: Specify the Client Secret for the given Client ID.

Authentication Type: Select any one of the following options of the Authentication mechanism from the drop-down menu.

- Certificate based Authentication - Refer to Step 5c of "Certificate Based Authentication in Azure AD Driver 5.1.6" on page 61.
- Client Secret Authentication

NOTE: You created Client ID and Client Secret while creating a proxy application in Azure AD. For more information, see "Creating a Proxy Application on Azure AD" on page 82.

Show Schema Extensions Configuration: To show the schema extensions configuration options for the application (Azure AD), select **Show**.

Enable Hybrid Operation Mode: In hybrid mode, the driver provisions only roles, licenses, teams and channels while the users and groups are provisioned by the AD driver. By default, the parameter is set to **Yes**. If you want to run the driver in normal mode, set the option to **No**.

Activate Azure Directory Roles: By default, the driver obtains the roles that have been pre-activated in Azure Directory. If you want the driver to activate all Azure Directory roles, set this option to **Yes**. This fetches all the activated roles in Identity Applications. These roles are also available at the driver startup. Roles activation is one time activity and need not be performed again.

To obtain only pre-activated roles, leave the setting unchanged.

Existing Schema Extensions: To retain the previously-loaded configuration from Azure AD, select **Preserve**. To remove existing configuration, specify **Remove**.

Add a schema extension: Specify appropriate configuration details while adding a schema extension. You can add multiple schema extensions if required.

- **Name of extension:** Specify the name of the schema extension. For example, `Title`.

  If you create multiple schema extensions with the same name, the driver uses the first extension in the list and ignores the remaining extensions that have the same name.
- **Type of extension:** Specify the data type for the configured schema extension. Ensure that the data type is a supported schema extension type in Azure AD.
- **Target objects of extension:** Lists the target objects for the schema extension. A schema extension can be extended to multiple target object classes. For example, if you have a schema extension called `Title`, it can be extended to a `User` and `Group` object classes.

NOTE: After adding the schema extension attribute, add the application attribute name to the driver filter in the following format:

```
extension_<client_id>_<attribute_name>
```

where `<client_id>` indicates the client ID that is used by the driver to connect to Azure AD.

For example, `extension_4691ac9cbee390e6e8e_Title`.

Subscriber Options: To view the Subscriber options, select **Show**.

Truststore file: Specify the name and path of the truststore file containing the trusted certificates used when a remote server is configured to provide server authentication. This file will contain certificates for Azure Graph and Exchange Service. For example, `c:\security\truststore`.

Proxy Host and Port: When an HTTP proxy is used, specify the host address and the host port. For example, 192.10.1.3:18180. Otherwise, leave the field blank.

Exchange and Powershell Service: When Identity Manager Exchange Service is enabled, the driver synchronizes Exchange users and groups using this service.

Exchange Authentication Type:

- Certificate Based Authentication - Refer to "Creating Client Certificate for IDM Exchange Service" on page 63
- Client Based Authentication

Exchange Service URL: Specify the URL of the Identity Manager Exchange Service.

For example, `https://<ip-addr>:<port>/ExchServer`.

Office 365 Exchange Online: To initiate a connection with Exchange Online and synchronize Office 365 exchange users and groups, select **Yes**.

Queue Operations: To enable queuing of objects when synchronizing between Azure AD and Identity Manager Exchange Service, select **True**.

Page Size: Set a value for the number of results displayed per page during Exchange Publisher poll.

Trace location: Specify the custom path where you want to save the Identity Manager Exchange Service logs. By default, the logs will be saved in this component's installation directory.

Trace Level: Set the trace level for the Identity Manager Exchange Service.

The driver supports five trace levels: NOTIFY, INFO, ERROR, MORE INFO, and DEBUG. The default trace level is NOTIFY. The next trace level, that is, INFO provides basic trace messages. ERROR provides some additional information than the previous level. Detailed messages are logged if you select INFO. DEBUG logs information on debugging data along with detailed messages.

Trace File Size Limit: Specify the trace file size limit in MB. The minimum value is 10 MB.

Database Password: Specify the database password. The driver uses this password to encrypt and connect to the Publisher cache. Ensure that the same password is used to reconnect to the cache at a later time.

Group cache clear interval: Specify the value as required to clear the exchange group related cache data. You must specify a numerical value (for example, -1, 0, 1, 2, etc.) in the Group cache clear interval field. The numerical value specified here corresponds to hours.For more information, see "Group Cache Clear Interval" on page 20.

Publisher Options: To view the Publisher options, select **Show**.

Enable Publisher: Allows you to enable or disable the Publisher connection for the driver.

Publisher Polling Interval: Specify a time period after which the driver should query Azure AD for new changes. The time is specified in minutes.

Heart Beat Interval: Allows the driver to send a periodic status message on the Publisher channel when there is no traffic for a specific duration. This indicates the time period at which the heart beat document is issued by the driver shim. The time is specified in minutes.

10 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 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 running. 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) 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.

11 On the **Azure AD Base** page, fill in the following fields, then click **Next**:

- **Domain Name:** Specify the Azure AD domain site context. For example, `<`*`domain name`*`>.onmicrosoft.com` or `<`*`domain name`*`>.com` format.
- **Identities to be synchronized:** Specify whether the driver should synchronize identities from AD or configure the Identity Vault to act as the identity provider.

  If you choose to configure the Identity Vault as an identity provider, association to any other directory is not required.

  When you choose to synchronize identity from AD, you can synchronize only users that have an association with AD. If you are using the driver in hybrid mode, select only **AD** option. This enables the driver to synchronize the identities from the Identity Vault to AD from where the identities will be synchronized to Azure AD cloud through Azure AD Connect.
- **Usage Location:** Specify the two letter country code for the user availing the Office 365 services.

12 (Conditional) On the **Install Azure AD Managed System Information** page, fill in the following fields to define the ownership of Azure AD, then click **Next**:

General Information

- **Name**: Specify a descriptive name for the managed system.
- **Description**: Specify a brief description of the managed system.

- **Location**: Specify the physical location of the managed system.
- **Vendor**: Select the vendor of the managed system.
- **Version**: Specify the version of the managed system.

System Ownership

- **Business Owner** - Select a user object in the Identity Vault that is the business owner of Azure AD. 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 Azure AD. This can only be a user object, not a role, group, or container.

  This page is only displayed if you selected to install the Data Collection packages and the Account Tracking packages.

System Classification

- **Classification**: Select the classification of Azure AD. This information is displayed in the reports. The options are as follows:
  - Mission-Critical
  - Vital
  - Not-Critical
  - Other

    If you select **Other**, you must specify a custom classification for Azure AD.
- **Environment**: Select the type of environment Azure AD provides. The options are as follows:
  - Development
  - Test
  - Staging
  - Production
  - Other

    If you select **Other**, you must specify a custom environment for Azure AD.

13. On the **Azure AD Password Synchronization** page, fill in the following fields, then click **Next**:
    - **Set Password Never Expires:** If you set this option to **True** on the newly created users, the password does not expire for those users.
    - **Disable Force Change Password at First Login:** If you set this option to **True**, a user is not prompted to change the password when the user logs in to Azure AD for the first time.
    - **Set Strong Password Required:** If you set this option to **True**, the user needs to set a strong password.
14. On the **Account Tracking** page, specify the name of the realm, security domain, or namespace in which the account name is unique. You must set the Realm to the Azure AD domain name.
15. On the **Confirm Installation Tasks** page, review the summary of tasks and click **Finish**.

The driver is now created. You can modify the configuration settings by Configuring the Driver.

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.

To edit the properties, perform the following steps:

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

   NOTE: These settings are only displayed if you installed the Entitlements package. If you selected the **Azure AD Hybrid Entitlements** package, only Roles, License, Teams and Channel Entitlements are supported with this 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 **False**, user and group membership synchronization is managed using the non-entitlement configuration method.

     NOTE: Using Group Membership Entitlement you can add or remove members from SharePoint Online.
   - **Use License Entitlement:** Ensure the value of this parameter is set to true to enable the driver to manage licenses using the License entitlement. By default, the value is set to **True**.
   - **SKU Entitlement:** Ensure the value of this parameter is set to true to enable the driver to manage SKU Subscription assignments based on the entitlement. By default, the value is set to **True**.
   - **Use Roles Entitlement:** Ensure the value of this parameter is set to true to enable the driver to manage roles using the Roles entitlement. By default, the value is set to **True**.
   - **Teams Entitlement:** Ensure the value of this parameter is set to true to enable the driver to manage roles using the Teams entitlement. By default, the value is set to **True**.
     - Add User as Owner to Team - Select “Yes” to add the User as “Owner” to Team.
   - **Channels Entitlement:** Ensure the value of this parameter is set to true to enable the driver to manage roles using the Channels entitlement. By default, the value is set to **True**.
     - Add User as Owner to Channel - Select “Yes” to add the User as “Owner” to Channel.

- Team and Channel Name Separator - Specify the character to separate Team and Channel name.

  For example: Team::Channel

5 Click **Apply**.

6 Modify any other settings as necessary.

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 Azure AD, your synchronization requirements for the driver might differ from the default policies. If this is the case, you require customization.

7 Click **OK** when finished.

8 Continue with "Deploying the Driver" on page 94.

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, 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 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, then select **Live** > **Start Driver**.

Creating the Driver Object in Designer

An administrator must use Designer to create the Azure AD driver.

The Azure AD driver is package-enabled and you must install and deploy the driver using Designer. You can use iManager to view and edit the deployed configuration. NetIQ recommends that you use Designer to perform any changes.

Activating the Driver

The Identity Manager driver for Office 365 and Azure AD is part of the Identity Manager Integration Module for Microsoft Enterprise.

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

Updating the Driver Files

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

To update the driver files perform the following steps:

1. Stop the driver instance by using iManager, Designer, or dxcmd by performing one of the following actions:
   - If the driver is running locally, stop the driver instance and the Identity Vault.
   - If the driver is running with a Remote Loader instance, stop the driver and the Remote Loader instance.

   For example, go to a command prompt on Linux and run `ndsmanage stopall`

2 Download the driver patch file to a temporary folder on your server.

3 Extract the contents of the driver patch file.

4 For Linux, open a command prompt and run the following command to upgrade the existing RPM:

4a rpm -Uvh <Driver Patch File Temporary Location>/linux/netiq-DXMLRESTAzure.rpm

4b Copy the following jars to the folder `/opt/novell/eDirectory/lib/dirxml/classes/`

- `IDM_AzureAD_5.1_SP6/common/asm-1.0.2.jar`
- `IDM_AzureAD_5.1_SP6/common/content-type-2.2.jar`
- `IDM_AzureAD_5.1_SP6/common/nimbus-jose-jwt-9.23.jar`
- `IDM_AzureAD_5.1_SP6/common/oauth2-oidc-sdk-9.39.jar`
- `IDM_AzureAD_5.1_SP6/common/msal4j-1.12.0.jar`
- `IDM_AzureAD_5.1_SP6/common/slf4j-log4j12-1.7.33.jar`
- `IDM_AzureAD_5.1_SP6/common/common-2.49.jar`
- `IDM_AzureAD_5.1_SP6/common/json-smart-2.4.8.jar`

5 For Windows, perform the following actions:

5a Navigate to the <*`Extracted Driver Patch File Temporary Location`*>`\windows` folder.

5b Copy the following jars to the folder `<IdentityManager installation>\NDS\lib or <IdentityManager installation>\RemoteLoader\<architecture>\lib`.

- `AZDriverShim.jar`
- `RestLib.jar`
- `OData.jar`
- `IDM_AzureAD_5.1_SP6/common/asm-1.0.2.jar`
- `IDM_AzureAD_5.1_SP6/common/content-type-2.2.jar`
- `IDM_AzureAD_5.1_SP6/common/nimbus-jose-jwt-9.23.jar`
- `IDM_AzureAD_5.1_SP6/common/oauth2-oidc-sdk-9.39.jar`
- `IDM_AzureAD_5.1_SP6/common/msal4j-1.12.0.jar`
- `IDM_AzureAD_5.1_SP6/common/slf4j-log4j12-1.7.33.jar`
- `IDM_AzureAD_5.1_SP6/common/common-2.49.jar`
- `IDM_AzureAD_5.1_SP6/common/json-smart-2.4.8.jar`

6 (Conditional) If the driver is running locally, start the Identity Vault and the driver instance.

For example, open a command prompt on Linux and run `ndsmanage startall`

7 (Conditional) If the driver is running with a Remote Loader, start the Remote Loader and the driver instance.

4 Installing Azure AD 5.1.5

The Identity Manager Driver for Office 365 and Azure Active Directory (Azure AD driver) allows you to seamlessly provision and deprovision users, group memberships, exchange mailboxes, roles, and licenses to Azure AD cloud. You can also configure the driver to integrate with Identity Manager Service for Exchange Online (Identity Manager Exchange Service) for synchronizing Office 365 attributes.

- "Preparing for Installation" on page 75
- "Identity Manager Exchange Service" on page 86
- "Creating a New Driver Object" on page 88
- "Updating the Driver Files" on page 96

Preparing for Installation

Securing Driver Communication

The driver communicates over SSL with Azure AD and Identity Manager Exchange Service.

IMPORTANT: The connection accepts certificates only from a Java keystore. Make sure that the keystore for the certificates is a Java keystore.

The following sections provide instructions for creating a secure connection:

- "Secured Communication with Microsoft Graph API" on page 75
- "Securing Communication with Identity Manager Exchange Service" on page 76
- "Certificate Based Authentication support in Azure AD Driver 5.1.5." on page 78

Secured Communication with Microsoft Graph API

To set up SSL between the driver and Azure AD graph REST endpoints, perform the following steps:

1. Open the following URL from your browser:
   - `https://login.microsoftonline.com/`
   - `https://graph.microsoft.com/`
   - `https://azure.microsoft.com/`
2. Obtain the public certificate and import it into the keystore.

For example (Suppose you are accessing https://graph.microsoft.com/), if you are using Google Chrome, perform the following steps:

- **2a** In the address bar, click 🔒 and then click › next to browser address bar(for example:**graph.microsoft.com**).
- **2b** Select **Certificate (Valid)**. The certificate is displayed.
- **2c** Click **Certification Path**. The Certification Path displays the hierarchical structure of the structure of all the certificates.
- **2d** Select the root certificate (the top most parent certificate), and click **View Certificate**. The root certificate is displayed.
- **2e** To save the certificate to your system, click **Details > Copy to File > Next > Next**.
- **2f** Enter a filename for the certificate and save it to a location as required.
- **2g** Add the exported key to the driver keystore using the following Java keytool command:

  You might have to create a new keystore(`.jks` file), if one such file doesn’t exist already. This keystore file will contain the public certificate of the Azure graph endpoint and the exchange service certificate.

  ```
  keytool -import -file <path to the graph cert file>\<certname.crt> -
  keystore <mykeystore> -alias <aliasname>
  ```

  For example: `keytool -import -file msgraph.cer -keystore azuread.jks -alias msgraph`

  NOTE:
  - Ensure to place the new keystore in IDM Server. In case of Remote Loader place the keystore file in the system where the Azure AD driver is running.
  - Ensure that you follow the above steps to import all the certificates into the keystore.

Securing Communication with Identity Manager Exchange Service

To set up SSL between the driver and Identity Manager Exchange Service, you need to create and import a server certificate into the root certificate store of the Windows server where the service is deployed. The following procedure assumes eDirectory as the Certificate Authority (CA).

1 Create a server certificate.

- **1a** In iManager, log in to the connected eDirectory server with administrator rights.
- **1b** Click **Roles and Tasks** > **NetIQ Certificate Server** > **Create Server Certificate**.
- **1c** Select the server and provide a **nickname** for the certificate.

  The nickname is same that you specified for **Certificate Alias** (example `azuread` as shown in previous section) while installing Identity Manager Exchange Service.
- **1d** Click **Next**, then click **Finish** to complete the certificate creation.

2 Export the server certificate from the connected eDirectory server and save it to a file in the `pfx` format.

- **2a** In iManager, log in to the connected eDirectory server with administrator rights.
- **2b** Click **Roles and Tasks** > **NetIQ Certificate Access** > **Server Certificates**, then select any server certificate.
- **2c** Click **Export**.

2d Select the certificate by nickname and select **Export Private Key**.

2e Enter the password and click **Next**.

2f To save the certificate to a file, click **Save the exported certificate**.

3 Import the certificate to the trusted store of the Windows server on which you will run Identity Manager Exchange Service.

3a Copy the `.pfx` file to the Windows server.

3b Click **Start** > **Run**> **mmc**.

3c Click **File** > **Add/Remove Snap-in**.

3d Select **Certificates** and click **Add** to import this snap-in by choosing *Computer account*.

3e Click **Finish**.

3f Navigate to **Certificates** > **Trusted Root Certification Authorities**.

3g Right-click and then select **All Tasks** > **Import**.

3h On the **Welcome to the Certificate Import Wizard** page, click **Next**.

3i Click **Browse** and select the eDirectory certificate you exported in "Export the server certificate from the connected eDirectory server and save it to a file in the `pfx format." on page 76`

3j Specify the password and click **Next**.

3k Click **Finish** to import the certificate into the trust store.

4 Start Identity Manager Exchange Service. For more information, see "Verifying and Starting the Identity Manager Exchange Service" on page 87.

5 Open the following Exchange service URL from your browser:

```
https://<Exchange_Service>:Port/ExchServer
```

6 Obtain the public certificate and import it into the same keystore which was created and placed in IDM Server as mentioned in (for example, the keystore `azuread`).

For example, perform the following steps to obtain a public certificate on Google Chrome:

6a Click ⚠ from the address bar and then click **Details**.

6b In the **Security** tab, click **View Certificate**.

6c In the **Details** tab, click **Copy to File**.

6d In the **Certificate Import Wizard**, click **Next**.

6e Select **DER encoded binary** and click **Next**.

6f Click **Browse** and navigate to the directory where you want to save the certificate.

6g Specify a name for the certificate and click **Next**.

6h Click **Finish** to complete the export.

6i Add the exported key to the driver keystore by using the following Java keytool command:

```
keytool -import -file <path to the exchange cert
file>\<certname.cer> -keystore <mykeystore> -alias <aliasname>
```

NOTE: Ensure that certificates inside the keystore have different alias names for all the imported certificates.

Certificate Based Authentication support in Azure AD Driver 5.1.5.

The driver now supports two kinds of authentications.

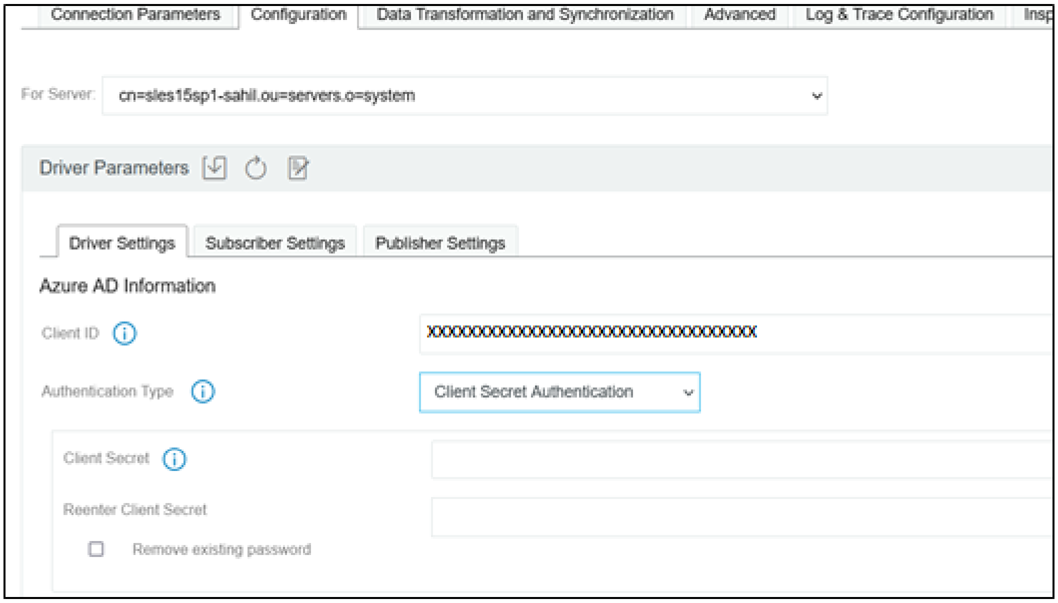
1. Client Secret Authentication: This is the existing authentication mechanism.If this option is chosen, specify the Client Secret Value.
2. Certificate based Authentication: This is the new authentication type supported by the driver.

Based on selected authentication types, perform the following steps to configure authentication types:

1. Client Secret Authentication - This is the existing authentication mechanism.

   1a Provide ClientSecret

   Figure 4-1



Connection Parameters | Configuration | Data Transformation and Synchronization | Advanced | Log & Trace Configuration | Insp

For Server: cn=sles15sp1-sahil.ou=servers.o=system

Driver Parameters

Driver Settings | Subscriber Settings | Publisher Settings

Azure AD Information

Client ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Authentication Type: Client Secret Authentication

Client Secret

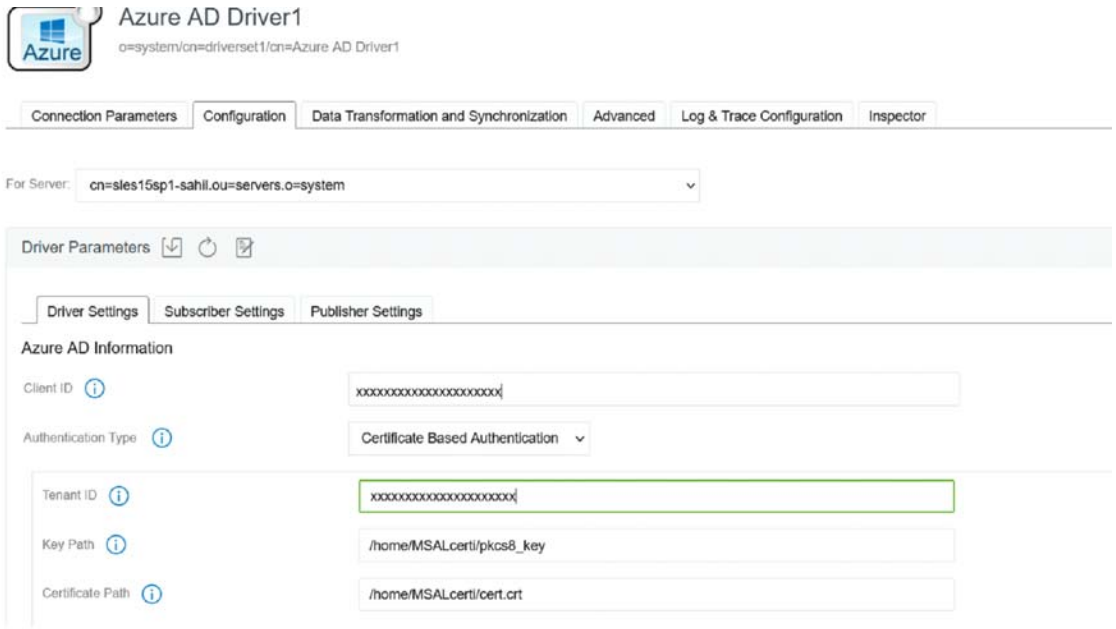
Reenter Client Secret

□ Remove existing password

2. Certificate Based Authentication - This is the new authentication type supported by the driver.

   A pair of private key and public certificate is required for this type of authentication as shown in the below image.

Figure 4-2



Azure AD Driver1

o=system/cn=driverset1/cn=Azure AD Driver1

For Server: cn=sles15sp1-sahil.ou=servers.o=system

Driver Parameters

Azure AD Information

Client ID: xxxxxxxxxxxxxxxxxxxxxx

Authentication Type: Certificate Based Authentication

Tenant ID: xxxxxxxxxxxxxxxxxxxxxx

Key Path: /home/MSALcerti/pkcs8_key

Certificate Path: /home/MSALcerti/cert.crt

2a Generate a private key in PKCS8 format by executing the below commands.

2a1 You can use an existing private key in PEM format or create a private key using the below command.

```
openssl genrsa -out private_key.pem 2048
```

2a2 Convert the private key into the PKCS8 format using the below command.

openssl pkcs8 -topk8 -inform PEM -outform DER -in private_key.pem -nocrypt > pkcs8_key

2b Generate a certificate signing request using the private key.

2b1 openssl req -new -key private_key.pem -out cert.csr

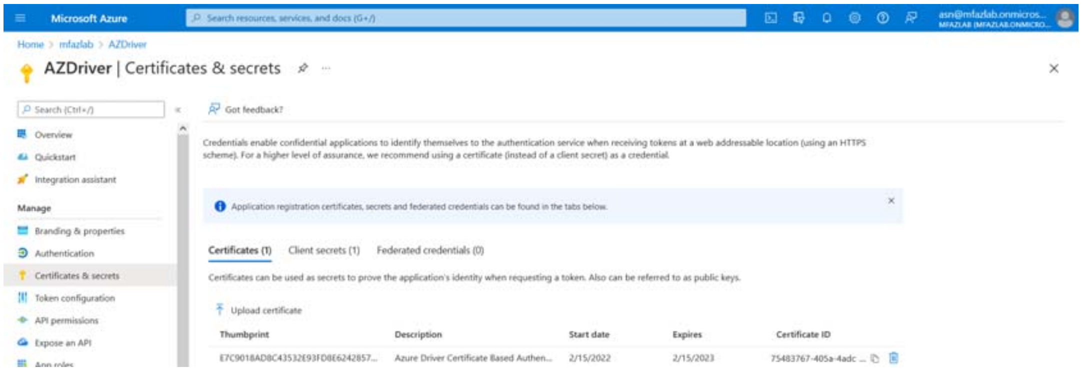
NOTE: This command will ask for a variety of extra information, like company name, country, and a password. None of this is used by the sample, so you can set these values as nothing/anything you want.

2b2
```
openssl x509 -req -days 365 -in cert.csr -signkey
private_key.pem -out cert.crt
```

3 Uploading certificate to Azure AD and configure the driver parameters.

3a Login to the Azure portal. In the Application menu blade, click on the Certificates & secrets, in the Certificates section, upload certificate generated in Step 2b2.

Figure 4-3



Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Application registration certificates, secrets and federated credentials can be found in the tabs below.

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

Thumbprint	Description	Start date	Expires	Certificate ID
E7C9018AD8C43532E93FD8E6242857...	Azure Driver Certificate Based Authen...	2/15/2022	2/15/2023	75483767-405a-4adc ...

3b Using Designer, perform the following steps.

3b1 Right Click Driver Object and click **Properties**

3b2 Goto **Driver Configuration** -> **Driver Parameter**

3b3 Under Driver Options, choose `Certificate Based Authentication` as authentication type and provide following details.

- Tenant ID
- Key Path (Ex: /azcerts/515/cba/pkcs8_key) - is the absolute location to key file created in Step 2a2.
- Certificate Path (Ex: /azcerts/515/cba/cba-cert.crt) -s the absolute location to the certificate created in Step 2b2.

Prerequisites

This section provides the prerequisites, considerations, and system setup needed to install the driver:

- "Prerequisites for the Driver" on page 80
- "Prerequisites for the User Account Configured in the Driver" on page 81
- "Prerequisites for Identity Manager Exchange Service" on page 81
- "Prerequisites for OAuth 2.0" on page 81
- "Assigning the Rights to the Application" on page 84
- "Prerequisites for Support of Modern Authentication" on page 85

Prerequisites for the Driver

The driver requires the following applications:

- Identity Manager 4.8.4 or later
- Identity Manager Designer 4.8.4 or later
- Identity Manager REST driver 1.1.2.0300 or later

Prerequisites for the User Account Configured in the Driver

You must ensure that the user account you are configuring in the driver has the following Roles or Permissions in the Azure application:

- At a minimum, the user account must have the following roles:
  - User Administrator
  - Exchange Service Administrator
  - Privileged Role Administrator
- The multi-factor authentication for the user account is disabled.
- A user account has conditional access. For more information on providing conditional access, see What is Conditional Access.

Prerequisites for Identity Manager Exchange Service

- Microsoft Windows Server 2016, or Microsoft Windows Server 2019
- Microsoft Visual C++ 2017 Redistributable packages for Visual Studio

  Download the packages from the Microsoft Downloads website.
- Install the ExchangeOnlineManagement module by executing the below command in PowerShell:

  ```
  Install-Module -Name ExchangeOnlineManagement
  ```
- Windows Azure AD Module for Windows Powershell on the computer where you will install Windows Powershell service.

  Perform the following steps to upgrade PowerShell to the latest version:

  1. Open a Windows PowerShell console.
  2. Run the following Install-Module cmdlet or Install-Script cmdlet:
     - If it is a module: `Install-Module -Name <moduleName> -RequiredVersion <version>`

       For example, `Install-Module -Name MSOnline -RequiredVersion 1.1.166.0`
     - If it is a script: `Install-Script -Name <scriptName> -RequiredVersion <version>`

Identity Manager Exchange Service can be run on a user configured port. However, the service cannot be used with any other REST client tools.

Prerequisites for OAuth 2.0

The driver uses OAuth 2.0 protocol to authenticate to Azure AD. To support this protocol for authentication, you need to have a proxy application for the Azure AD driver on Azure AD. The Client ID and Client Secret allotted to the application will be later used in the Azure AD driver configuration. For more information about Azure Active Directory Application Proxy, see Microsoft Azure documentation.

Creating a Proxy Application on Azure AD

A proxy application is created in the Azure Portal. Creating a proxy application involves the following steps:

1. Registering an application and obtaining a client ID. For more information see, Registering an Application.
2. Generating an application password or the client secret. For more information see, Certificates and Secrets (https://learn.microsoft.com/en-us/entra/identity-platform/quickstart-register-app#add-credentials).
3. Configuring API permissions (Delegated and Application permissions). Set the delegated and application permissions as shown in the following table. For more information see, Add permissions to access web APIs.

Table 4-1 *List of Application type of APIs*

API	Type	Description	Admin Consent
Application.Read.All	Application	Read all applications	Grant Admin Consent
Application.ReadWrite.All	Application	Read and write all applications	Grant Admin Consent
AuditLog.Read.All	Application	Read all audit log data	Grant Admin Consent
Device.Read.All	Application	Read all devices	Grant Admin Consent
Device.ReadWrite.All	Application	Read and write devices	Grant Admin Consent
Directory.Read.All	Application	Read directory data	Grant Admin Consent
Directory.ReadWrite.All	Application	Read and write directory data	Grant Admin Consent
Domain.ReadWrite.All	Application	Read and write domains	Grant Admin Consent
Group.Create	Application	Create groups	Grant Admin Consent
Group.Read.All	Application	Read all groups	Grant Admin Consent
Group.ReadWrite.All	Application	Read and write all groups	Grant Admin Consent
GroupMember.Read.All	Application	Read all group memberships	Grant Admin Consent
GroupMember.ReadWrite.All	Application	Read and write all group memberships	Grant Admin Consent
UserAuthenticationMethod.Read.All	Application	Read all user authentication methods	Grant Admin Consent
UserAuthenticationMethod.ReadWrite.All	Application	Read and write all users authentication methods	Grant Admin Consent
User.Read.All	Application	Read all users' full profiles	Grant Admin Consent
User.ReadWrite.All	Application	Read and write all users' full profiles	Grant Admin Consent

Table 4-2 List of Delegated type of APIs

API	Type	Description	Admin Consent
Directory.AccessAsUser.All	Delegated	Access directory as the signed in user	Grant Admin Consent
Directory.Read.All	Delegated	Read directory data	Grant Admin Consent
Directory.ReadWrite.All	Delegated	Read and write directory data	Grant Admin Consent
Group.Read.All	Delegated	Read all groups	Grant Admin Consent
Group.ReadWrite.All	Delegated	Read and write all groups	Grant Admin Consent
GroupMember.Read.All	Delegated	Read group memberships	Grant Admin Consent
GroupMember.ReadWrite.All	Delegated	Read and write group memberships	Grant Admin Consent
RoleManagement.Read.All	Delegated	Read role management data for all RBAC providers	Grant Admin Consent
RoleManagement.Read.Directory	Delegated	Read directory RBAC settings	Grant Admin Consent
UserAuthenticationMethod.Read	Delegated	Read user authentication methods	Grant Admin Consent
UserAuthenticationMethod.Read.All	Delegated	Read all user's authentication methods	Grant Admin Consent
UserAuthenticationMethod.ReadWrite	Delegated	Read and write user authentication methods	Grant Admin Consent
UserAuthenticationMethod.ReadWrite.All	Delegated	Read and write all user authentication methods	Grant Admin Consent
User.Read	Delegated	Sign in and read user profile	Grant Admin Consent
User.Read.All	Delegated	Read all users' full profiles	Grant Admin Consent
User.ReadBasic.All	Delegated	Read all users' basic profiles	Grant Admin Consent
User.ReadWrite	Delegated	Read and write access to user profile	Grant Admin Consent
User.ReadWrite.All	Delegated	Read and write all users' full profiles	Grant Admin Consent

NOTE: You may need to provide additional permissions based on their requirement.

As a minimal requirement, the user account used in Azure AD Driver configuration should have the following Roles:

- User Administrator

- Exchange Administrator

NOTE: You may need to provide additional roles based on their requirement.

The Client ID and Client Secret can now be used for driver configurations or any other REST clients.

Assigning the Rights to the Application

1. In the server where you have installed the exchange service, login to PowerShell and connect to the Office 365 Exchange Online service, using the following command:

   ```
   Connect-MSolService
   ```

2. To obtain the Client ID for your application, replace <AppPrincipalId> with the Client ID that you obtained from Creating a Proxy Application on Azure AD and run the following commands in PowerShell.

   ```
   Get-MsolServicePrincipal | ft DisplayName, <AppPrincipalId> -AutoSize
   ```

   ```
   $ClientIdWebApp = '<AppPrincipalId>'
   ```

   ```
   $webApp = Get-MsolServicePrincipal –AppPrincipalId $ClientIdWebApp
   ```

3. Assign the Company Administrator rights to your application using the Client ID obtained in Step 2 by running the following command:

   ```
   Add-MsolRoleMember -RoleName "Company Administrator" -RoleMemberType
   ServicePrincipal -RoleMemberObjectId $webApp.ObjectID
   ```

   The Company Administrator role will give you rights to delete the directory objects.

IMPORTANT: The Company Administrator role in Identity Manager is mapped to the Global Administrator role in Azure AD.

Ensure that the account used by the driver to connect to the Exchange Online service has the correct roles to load and execute the following cmdlets:

- New-Mailbox
- Set-Mailbox
- Get-Mailbox
- Remove-Mailbox
- New-MailUser
- Set-MailUser
- Get-MailUser
- Remove-MailUser
- Set-User
- Get-User
- New-DistributionGroup
- Set-DistributionGroup

- Set-Group
- Get-DistributionGroup
- Get-Group
- Remove-DistributionGroup
- Add-DistributionGroupMember
- Remove-DistributionGroupMember
- Get-DistributionGroupMember
- Add-RoleGroupMember
- Remove-RoleGroupMember
- Get-RoleGroupMember
- New-UnifiedGroup
- Get-UnifiedGroup
- Set-UnifiedGroup
- Remove-UnifiedGroup
- Add-UnifiedGroupLinks
- Remove-UnifiedGroupLinks
- Get-UnifiedGroupLinks

Absence of the required roles prevents the driver from executing the cmdlets that require those roles.

Prerequisites for Support of Modern Authentication

As Microsoft Office 365 is deprecating the **Basic** authentication, you must now configure the driver with modern authentication method. You must also ensure to have the earlier mentioned prerequisites (“Prerequisites for the Driver” on page 80, “Prerequisites for Identity Manager Exchange Service” on page 81, and “Prerequisites for OAuth 2.0” on page 81) met, and then proceed with the following prerequisites.

The following prerequisites are specific to modern authentication. It is highly recommended to upgrade the driver version 5.1.x to 5.1.3 or later to support modern authentication.

Installing the Microsoft Exchange Online PowerShell V2 (EXO V2)

You must install the Microsoft Exchange Online PowerShell V2 module to support the new API’s. For more information on EXO V2 module, see About the Exchange Online PowerShell V2 module.

- For prerequisites to install the EXO V2 module, see Prerequisites for EXO V2 module.
- For installing the EXO V2 module, see Install the EXO V2 module.

Configuring Azure AD Proxy Application for Modern Authentication Methods

You must enable the permission in the Azure portal to access Microsoft Office 365 with modern authentication.

The procedure to set the permission is shown below:

1. Login to the Azure AD Portal.
2. Select **Azure Active Directory**.
3. Navigate to **App Registration >** find and select your application in the list (for example: *<MySample_Azure_Appln>*) **> Authentication > Advanced Settings**.
4. Set **Treat Application as a Public Client** permission to **Yes**.

IMPORTANT: The multi-factor authentication (MFA) must be disabled for the Azure account which is used with the driver.

Identity Manager Exchange Service

You can install the Azure AD driver on the Identity Manager server or with the Remote Loader.

Installing the Driver and the Identity Manager Exchange Service

The driver installation program guides you through the driver and the Identity Manager Exchange Service installation.

NOTE:

1. IDM Exchange service must be run on the same machine as the driver and configured to listen only on local host.
2. IDM Exchange service must be run with least privilege required for the configured `PowerShell cmdlets` to execute.
3. Only system administrator must be provided access to the IDM exchange service machine

Perform the following actions to install and configure the Exchange Service:

1. Copy Exchange Service from `[ISO]:\products\IDM\windows\setup\drivers\azuread\ExchangeService` to any local drive on the server you intend to run this service.
2. Navigate to the directory where you copied the ExchServerHost.exe in the first step.

   For example, `C:\ExchangeService`
3. Run the following command to install the Exchange Service:

   ```
   <location of InstallUtil>\InstallUtil.exe ExchServerHost.exe
   ```

   For example:

   ```
   C:\Windows\Microsoft.NET\Framework64\v4.0.30319\InstallUtil.exe
   ExchServerHost.exe
   ```

where, `C:\Windows\Microsoft.NET\Framework64\v4.0.30319\InstallUtil.exe` is the location where `InstallUtil.exe` is located.

4 Ensure the server certificate is available in iManager. To create the server certificate, see "Securing Communication with Identity Manager Exchange Service" on page 76

5 Open cmd prompt, and navigate to the local drive location where the ExchangeService is saved, as mentioned in Step 1 on page 76 (`\products\IDM\windows\setup\drivers\azuread\ExchangeService\`), and execute the command `configureExchService.bat <port> <certificate_name>`.

   For example: `configureExchService.bat 9001 azuread`. Where `9001` is the port number and `azuread` is the nickname of the certificate that was created in iManager.

6 To start the service, navigate to **Control Panel** > **Administrative Tools** > **Services**.

7 Right-click the IDMExchangeOnline service and select **Start**.

NOTE: To uninstall the service, open a .NET command prompt and issue the `InstallUtil /u` ExchServerHost.exe command.

NetIQ recommends you to use TLS 1.1 and TLS 1.2 protocols with the Identity Manager Exchange Service. If you are using ciphers and protocols such as RC4 and Triple DES, or SSLv2/v3 on a server running Identity Manager Exchange Service, you must disable them using the `disableWeakCiphers.reg` file provided in the Exchange Service installation directory. You can either execute the registry file or import the file into Windows Registry. After the changes are made, restart the server. For more information about restricting the use of certain cryptographic algorithms and protocols on Windows servers, see Microsoft Support Site.

Verifying and Starting the Identity Manager Exchange Service

After finishing the installation of Identity Manager Exchange Service, verify that the service is properly installed.

NOTE: Ensure that SSL is configured for the Identity Manager Exchange Service before starting the service. This is a mandatory step before running the service. For more information, see Securing Communication with Identity Manager Exchange Service.

1 From the **Start** menu, type **regedit**.

2 On the **Registry Editor** page, locate the service at **HKEY_LOCAL_MACHINE** > **Software** > **Novell** > **ExchServer** and verify that the **Port** and **CertificateFriendlyName** have the correct values.

   The **CertificateFriendlyName** must be the same as **Certificate Alias** that you specified in Step 1 of the "Securing Communication with Identity Manager Exchange Service" on page 76.

3 Navigate to the services that are running on your server and start the `IDMExchangeOnline` service.

Verifying the Provisioning of Exchange Mailbox

To verify the provisioned mailboxes for users, follow the procedure provided in the Microsoft Exchange Admin Center.

Creating a New Driver Object

You install the Azure AD driver files on the server where you want to run the driver, and then proceed to create the driver in the Identity Vault. You create the Azure AD driver by installing the driver packages or importing the basic driver configuration file and then modifying the driver configuration to suit your environment.

Creating the Driver Object in Designer

An administrator must use Designer to create the Azure AD driver.

The Azure AD driver is package-enabled and you must install and deploy the driver using Designer. You can use iManager to view and edit the deployed configuration. NetIQ recommends that you use Designer to perform any changes.

Importing the Driver Packages

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

1. Open Designer.
2. In the toolbar, click **Help** > **Check for Package Updates**.
3. Click **OK** to update the packages or click **OK** if the packages are up-to-date.
4. Right-click **Package Catalog** and then select **Import Package**.
5. Select any Azure AD driver packages.

   or

   Click **Select All** to import all of the packages displayed.
6. Click **OK** to import the selected packages, then click **OK** in the successfully imported package message.

After the packages are imported, continue with Installing the Driver Packages.

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

   Alternatively, you can drag and drop the Azure AD driver icon from the Cloud section of the Designer palette.
3. In the **Driver Configuration** wizard, select the `Azure AD Base` package from the list of base packages, then click **Next**.

4 Select the optional features to install for the Azure AD driver. All options are selected by default. The options are:

- **Default Configuration:** This package contains the default configuration information for the driver. Always leave this option selected.

  NOTE: The **Azure AD Default** package and **Azure AD Exchange Default** package are included in **Default Configuration** package. By default, the **Azure AD Exchange Default** package is not selected. Select this package if you plan to use the Identity Manager Exchange Service.

- **Entitlements and License Support:** This package contains configuration information and policies for synchronizing user accounts, group membership, roles, exchange roles, licenses and SKU. If you want to enable account creation and auditing through entitlements, verify that this option is selected.

  To enable the hybrid mode, select the **Azure AD Hybrid Entitlements** package. In this mode, the driver supports only Roles and License entitlements.

- **Exchange Role Support:** This package contains configuration information and policies for synchronizing exchange roles. Ensure that this package is selected.
- **Password Synchronization:** This package contains the policies that enables the driver to synchronize passwords. If you want to synchronize passwords, ensure that this package is selected.
- **Account Tracking:** This package contains the policies that enable you to track accounts for reports. If you are using Identity Reporting, ensure that this package is selected.
- **Data Collection:** This package contains the policies that enables the driver to collect data for reports. If you are using the Identity Manager Reporting Module, ensure that this package is selected.

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. Click **OK** to install any additional package dependencies.

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

9 On the **Driver Configuration** page, fill in the following fields to configure the driver:

Authentication ID: Provide the authentication information while configuring the domain connections for the driver. This is a qualified `userprincipalname` on Azure AD with login permissions. For example, `admin@domain.onmicrosoft.com`

Password: Specify the password for the driver to authenticate to Azure AD.

Driver Options: To view the driver options, select **Show**.

Client ID: Specify the account name which the driver will use to access the Azure AD applications.

Authentication Type: Select any one of the following options of the Authentication mechanism from the drop-down menu.

- Certificate based Authentication - Specify the below parameters when Certificate based Authentication type is chosen.
  - Tenant ID - Specify the Tenant ID of customer where the driver will connect.
  - Key Path - Specify the certificate path for private key generated as part of Certificate based Authentication.
  - Certificate Path - Specify the path and the file name for the certificate created as part of configuring the Certificate based Authentication.
- Client Secret Authentication - Specify the Client Secret password for the Client ID to access the Azure AD applications.

NOTE: You created Client ID and Client Secret while creating a proxy application in Azure AD. For more information, see "Creating a Proxy Application on Azure AD" on page 82.

Show Schema Extensions Configuration: To show the schema extensions configuration options for the application (Azure AD), select **Show**.

Enable Hybrid Operation Mode: In hybrid mode, the driver provisions only roles and licenses while the users and groups are provisioned by the AD driver. By default, the parameter is set to **Yes**. If you want to run the driver in normal mode, set the option to **No**.

Activate Azure Directory Roles: By default, the driver obtains the roles that have been pre-activated in Azure Directory. If you want the driver to activate all Azure Directory roles, set this option to **Yes**. This fetches all the activated roles in Identity Applications. These roles are also available at the driver startup. Roles activation is one time activity and need not be performed again.

To obtain only pre-activated roles, leave the setting unchanged.

Existing Schema Extensions: To retain the previously-loaded configuration from Azure AD, select **Preserve**. To remove existing configuration, specify **Remove**.

Add a schema extension: Specify appropriate configuration details while adding a schema extension. You can add multiple schema extensions if required.

- **Name of extension:** Specify the name of the schema extension. For example, `Title`.

  If you create multiple schema extensions with the same name, the driver uses the first extension in the list and ignores the remaining extensions that have the same name.
- **Type of extension:** Specify the data type for the configured schema extension. Ensure that the data type is a supported schema extension type in Azure AD.
- **Target objects of extension:** Lists the target objects for the schema extension. A schema extension can be extended to multiple target object classes. For example, if you have a schema extension called `Title`, it can be extended to a `User` and `Group` object classes.

NOTE: After adding the schema extension attribute, add the application attribute name to the driver filter in the following format:

`extension_<client_id>_<attribute_name>`

where `<client_id>` indicates the client ID that is used by the driver to connect to Azure AD.

For example, `extension_4691ac9cbee390e6e8e_Title`.

Subscriber Options: To view the Subscriber options, select **Show**.

Truststore file: Specify the name and path of the truststore file containing the trusted certificates used when a remote server is configured to provide server authentication. This file will contain certificates for Azure Graph and Exchange Service. For example, `c:\security\truststore`.

Proxy Host and Port: When an HTTP proxy is used, specify the host address and the host port. For example, 192.10.1.3:18180. Otherwise, leave the field blank.

Exchange and Powershell Service: When Identity Manager Exchange Service is enabled, the driver synchronizes Exchange users and groups using this service.

Exchange Service URL: Specify the URL of the Identity Manager Exchange Service.

For example, `https://<ip-addr>:<port>/ExchServer`.

Office 365 Exchange Online: To initiate a connection with Exchange Online and synchronize Office 365 exchange users and groups, select **Yes**.

Queue Operations: To enable queuing of objects when synchronizing between Azure AD and Identity Manager Exchange Service, select **True**.

Page Size: Set a value for the number of results displayed per page during Exchange Publisher poll.

Trace location: Specify the custom path where you want to save the Identity Manager Exchange Service logs. By default, the logs will be saved in this component's installation directory.

Trace Level: Set the trace level for the Identity Manager Exchange Service.

The driver supports five trace levels: NOTIFY, INFO, ERROR, MORE INFO, and DEBUG. The default trace level is NOTIFY. The next trace level, that is, INFO provides basic trace messages. ERROR provides some additional information than the previous level. Detailed messages are logged if you select INFO. DEBUG logs information on debugging data along with detailed messages.

Trace File Size Limit: Specify the trace file size limit in MB. The minimum value is 10 MB.

Database Password: Specify the database password. The driver uses this password to encrypt and connect to the Publisher cache. Ensure that the same password is used to reconnect to the cache at a later time.

Group cache clear interval: Specify the value as required to clear the exchange group related cache data. You must specify a numerical value (for example, -1, 0, 1, 2, etc.) in the Group cache clear interval field. The numerical value specified here corresponds to hours.For more information, see "Group Cache Clear Interval" on page 20.

Publisher Options: To view the Publisher options, select **Show**.

Enable Publisher: Allows you to enable or disable the Publisher connection for the driver.

Publisher Polling Interval: Specify a time period after which the driver should query Azure AD for new changes. The time is specified in minutes.

Heart Beat Interval: Allows the driver to send a periodic status message on the Publisher channel when there is no traffic for a specific duration. This indicates the time period at which the heart beat document is issued by the driver shim. The time is specified in minutes.

10. 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 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 running. 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) 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.

11 On the **Azure AD Base** page, fill in the following fields, then click **Next**:

- **Domain Name:** Specify the Azure AD domain site context. For example, <*domain name*>`.onmicrosoft.com` or <*domain name*>`.com` format.
- **Identities to be synchronized:** Specify whether the driver should synchronize identities from AD or configure the Identity Vault to act as the identity provider.

  If you choose to configure the Identity Vault as an identity provider, association to any other directory is not required.

  When you choose to synchronize identity from AD, you can synchronize only users that have an association with AD. If you are using the driver in hybrid mode, select only **AD** option. This enables the driver to synchronize the identities from the Identity Vault to AD from where the identities will be synchronized to Azure AD cloud through Azure AD Connect.
- **Usage Location:** Specify the two letter country code for the user availing the Office 365 services.

12 (Conditional) On the **Install Azure AD Managed System Information** page, fill in the following fields to define the ownership of Azure AD, then click **Next**:

General Information

- **Name**: Specify a descriptive name for the managed system.
- **Description**: Specify a brief description of the managed system.
- **Location**: Specify the physical location of the managed system.
- **Vendor**: Select the vendor of the managed system.
- **Version**: Specify the version of the managed system.

System Ownership

- **Business Owner** - Select a user object in the Identity Vault that is the business owner of Azure AD. 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 Azure AD. This can only be a user object, not a role, group, or container.

  This page is only displayed if you selected to install the Data Collection packages and the Account Tracking packages.

System Classification

- **Classification**: Select the classification of Azure AD. This information is displayed in the reports. The options are as follows:
  - Mission-Critical
  - Vital
  - Not-Critical
  - Other

    If you select **Other**, you must specify a custom classification for Azure AD.
- **Environment**: Select the type of environment Azure AD provides. The options are as follows:
  - Development
  - Test
  - Staging
  - Production
  - Other

    If you select **Other**, you must specify a custom environment for Azure AD.

13 On the **Azure AD Password Synchronization** page, fill in the following fields, then click **Next**:

- **Set Password Never Expires:** If you set this option to **True** on the newly created users, the password does not expire for those users.
- **Disable Force Change Password at First Login:** If you set this option to **True**, a user is not prompted to change the password when the user logs in to Azure AD for the first time.
- **Set Strong Password Required:** If you set this option to **True**, the user needs to set a strong password.

14 On the **Account Tracking** page, specify the name of the realm, security domain, or namespace in which the account name is unique. You must set the Realm to the Azure AD domain name.

15 On the **Confirm Installation Tasks** page, review the summary of tasks and click **Finish**.

The driver is now created. You can modify the configuration settings by Configuring the Driver.

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.

To edit the properties, perform the following steps:

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

   NOTE: These settings are only displayed if you installed the Entitlements package. If you selected the **Azure AD Hybrid Entitlements** package, only Roles and License entitlements are supported with this 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 **False**, user and group membership synchronization is managed using the non-entitlement configuration method.
   - **Use License Entitlement:** Ensure the value of this parameter is set to true to enable the driver to manage licenses using the License entitlement. By default, the value is set to **True**.
   - **SKU Entitlement:** Ensure the value of this parameter is set to true to enable the driver to manage SKU Subscription assignments based on the entitlement. By default, the value is set to **True**.
   - **Use Roles Entitlement:** Ensure the value of this parameter is set to true to enable the driver to manage roles using the Roles entitlement. By default, the value is set to **True**.
5. Click **Apply**.
6. Modify any other settings as necessary.

   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 Azure AD, your synchronization requirements for the driver might differ from the default policies. If this is the case, you require customization.
7. Click **OK** when finished.
8. Continue with “Deploying the Driver” on page 94.

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, 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 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, then select **Live** > **Start Driver**.

Creating the Driver Object in Designer

An administrator must use Designer to create the Azure AD driver.

The Azure AD driver is package-enabled and you must install and deploy the driver using Designer. You can use iManager to view and edit the deployed configuration. NetIQ recommends that you use Designer to perform any changes.

Activating the Driver

The Identity Manager driver for Office 365 and Azure AD is part of the Identity Manager Integration Module for Microsoft Enterprise.

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

Updating the Driver Files

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

To update the driver files perform the following steps:

1. Stop the driver instance by using iManager, Designer, or dxcmd by performing one of the following actions:
   - If the driver is running locally, stop the driver instance and the Identity Vault.
   - If the driver is running with a Remote Loader instance, stop the driver and the Remote Loader instance.

   For example, go to a command prompt on Linux and run `ndsmanage stopall`
2. Download the driver patch file to a temporary folder on your server.
3. Extract the contents of the driver patch file.
4. For Linux, open a command prompt and run the following command to upgrade the existing RPM:

   4a rpm -Uvh <Driver Patch File Temporary Location>/linux/netiq-DXMLRESTAzure.rpm

   4b Copy the following jars to the folder `/opt/novell/eDirectory/lib/dirxml/classes/`
   - `IDM_AzureAD_5.1_SP5/common/asm-1.0.2.jar`
   - `IDM_AzureAD_5.1_SP5/common/common-2.45.4.jar`
   - `IDM_AzureAD_5.1_SP5/common/content-type-2.2.jar`
   - `IDM_AzureAD_5.1_SP5/common/json-smart-2.4.7.jar`
   - `IDM_AzureAD_5.1_SP5/common/msal4j-1.11.0.jar`
   - `IDM_AzureAD_5.1_SP5/common/nimbus-jose-jwt-9.15.2.jar`
   - `IDM_AzureAD_5.1_SP5/common/oauth2-oidc-sdk-9.20.1.jar`
   - `DM_AzureAD_5.1_SP5/common/slf4j-log4j12-1.7.32.jar`

5 For Windows, perform the following actions:

5a Navigate to the <*Extracted Driver Patch File Temporary Location*>\windows folder.

5b Copy the following jars to the folder <IdentityManager installation>\NDS\lib or <IdentityManager installation>\RemoteLoader\<architecture>\lib .

- AZDriverShim.jar
- RestLib.jar
- OData.jar
- IDM_AzureAD_5.1_SP5/common/asm-1.0.2.jar
- IDM_AzureAD_5.1_SP5/common/common-2.45.4.jar
- IDM_AzureAD_5.1_SP5/common/content-type-2.2.jar
- IDM_AzureAD_5.1_SP5/common/json-smart-2.4.7.jar
- IDM_AzureAD_5.1_SP5/common/msal4j-1.11.0.jar
- IDM_AzureAD_5.1_SP5/common/nimbus-jose-jwt-9.15.2.jar
- IDM_AzureAD_5.1_SP5/common/oauth2-oidc-sdk-9.20.1.jar
- DM_AzureAD_5.1_SP5/common/slf4j-log4j12-1.7.32.jar

6 (Conditional) If the driver is running locally, start the Identity Vault and the driver instance.

For example, open a command prompt on Linux and run ndsmanage startall

7 (Conditional) If the driver is running with a Remote Loader, start the Remote Loader and the driver instance.

5 Installing Azure AD Driver prior to 5.1.5

The Identity Manager Driver for Office 365 and Azure Active Directory (Azure AD driver) allows you to seamlessly provision and deprovision users, group memberships, exchange mailboxes, roles, and licenses to Azure AD cloud. You can also configure the driver to integrate with Identity Manager Service for Exchange Online (Identity Manager Exchange Service) for synchronizing Office 365 attributes.

- "Securing Driver Communication" on page 99
- "Installing the Driver Files" on page 101
- "Creating a New Driver Object" on page 107
- "Upgrading an existing Driver" on page 115

Securing Driver Communication

The driver communicates over SSL with Azure AD and Identity Manager Exchange Service.

IMPORTANT: The connection accepts certificates only from a Java keystore. Make sure that the keystore for the certificates is a Java keystore.

The following sections provide instructions for creating a secure connection:

- "Securing Communication with Azure AD Graph" on page 99
- "Securing Communication with Identity Manager Exchange Service" on page 100

Securing Communication with Azure AD Graph

To set up SSL between the driver and Azure AD graph REST endpoints, perform the following steps:

1 Open the following URL from your browser:

```
https://graph.windows.net/
```

2 Obtain the public certificate and import it into the keystore.

For example, if you are using Mozilla Firefox, perform the following steps:

2a In the address bar, click 🔒 and then click › next to **graph.windows.net**.

2b Select **Certificate (Valid)**. The certificate is displayed.

2c Click **Certification Path**. The Certification Path displays the hierarchical structure of the structure of all the certificates.

2d Select the root certificate (the top most parent certificate), and click **View Certificate**. The root certificate is displayed.

2e To save the certificate to your system, click **Details > Copy to File > Next > Next**.

2f Enter a filename for the certificate and save it to a location as required.

2g Add the exported key to the driver keystore using the following Java keytool command:

You might have to create a new keystore(`.jks` file), if one such file doesn't exist already. This keystore file will contain the public certificate of the Azure graph endpoint and the exchange service certificate.

```
keytool -import -file <path to the graph cert file>\<certname.crt> -
keystore <mykeystore> -alias <aliasname>
```

For example: `keytool -import -file azuread.crt -keystore azuread.jks -alias azuread`.

NOTE:
- Ensure to place the new keystore in IDM Server. In case of Remote Loader place the keystore file in the system where the Azure AD driver is running.
- Ensure that you follow the above steps to import all the certificates into the keystore.

Securing Communication with Identity Manager Exchange Service

To set up SSL between the driver and Identity Manager Exchange Service, you need to create and import a server certificate into the root certificate store of the Windows server where the service is deployed. The following procedure assumes eDirectory as the Certificate Authority (CA).

1 Create a server certificate.

1a In iManager, log in to the connected eDirectory server with administrator rights.

1b Click **Roles and Tasks** > **NetIQ Certificate Server** > **Create Server Certificate**.

1c Select the server and provide a **nickname** for the certificate.

The nickname is same that you specified for **Certificate Alias** (example `azuread` as shown in previous section) while installing Identity Manager Exchange Service.

1d Click **Next**, then click **Finish** to complete the certificate creation.

2 Export the server certificate from the connected eDirectory server and save it to a file in the `pfx` format.

2a In iManager, log in to the connected eDirectory server with administrator rights.

2b Click **Roles and Tasks** > **NetIQ Certificate Access** > **Server Certificates**, then select any server certificate.

2c Click **Export**.

2d Select the certificate by nickname and select **Export Private Key**.

2e Enter the password and click **Next**.

2f To save the certificate to a file, click **Save the exported certificate**.

3 Import the certificate to the trusted store of the Windows server on which you will run Identity Manager Exchange Service.

3a Copy the `.pfx` file to the Windows server.

3b Click **Start** > **Run**> **mmc**.

3c Click **File** > **Add/Remove Snap-in**.

3d Select **Certificates** and click **Add** to import this snap-in by choosing *Computer account*.

3e Click **Finish**.

3f Navigate to **Certificates** > **Trusted Root Certification Authorities**.

3g Right-click and then select **All Tasks** > **Import**.

3h On the **Welcome to the Certificate Import Wizard** page, click **Next**.

3i Click **Browse** and select the eDirectory certificate you exported in Step 2 on page 100.

3j Specify the password and click **Next**.

3k Click **Finish** to import the certificate into the trust store.

4 Start Identity Manager Exchange Service. For more information, see "Verifying and Starting the Identity Manager Exchange Service" on page 107.

5 Open the following Exchange service URL from your browser:

```
https://<Exchange_Service>:Port/ExchServer
```

6 Obtain the public certificate and import it into the same keystore which was created and placed in IDM Server as mentioned in Step 2g on page 100 (for example, the keystore `azuread` as shown in the example for the Step 2g on page 100).

For example, perform the following steps to obtain a public certificate on Google Chrome:

6a Click ⚠ from the address bar and then click **Details**.

6b In the **Security** tab, click **View Certificate**.

6c In the **Details** tab, click **Copy to File**.

6d In the **Certificate Import Wizard**, click **Next**.

6e Select **DER encoded binary** and click **Next**.

6f Click **Browse** and navigate to the directory where you want to save the certificate.

6g Specify a name for the certificate and click **Next**.

6h Click **Finish** to complete the export.

6i Add the exported key to the driver keystore by using the following Java keytool command:

```
keytool -import -file <path to the exchange cert
file>\<certname.cer> -keystore <mykeystore> -alias <aliasname>
```

NOTE: Ensure the keystore alias names are different for Azure AD Graph and the Exchange Service.

Installing the Driver Files

You can install the Azure AD driver on the Identity Manager server or with the Remote Loader.

Preparing for Installation

This section provides the prerequisites, considerations, and system setup needed to install the driver:

- "Prerequisites for the Driver" on page 102
- "Prerequisites for the User Account Configured in the Driver" on page 102

- "Prerequisites for Identity Manager Exchange Service" on page 102
- "Prerequisites for OAuth 2.0" on page 103
- "Assigning the Rights to the Application" on page 104
- "Prerequisites for Support of Modern Authentication" on page 105

Prerequisites for the Driver

The driver requires the following applications:

- Identity Manager 4.7 or later
- Identity Manager Designer 4.7 or later
- Identity Manager REST driver 1.0.0.1 or later

Prerequisites for the User Account Configured in the Driver

You must ensure that the user account you are configuring in the driver has the following Roles or Permissions in the Azure application:

- At a minimum, the user account must have the following roles:
  - User Administrator
  - Exchange Service Administrator
  - Privileged Role Administrator
- The multi-factor authentication for the user account is disabled.
- A user account has conditional access. For more information on providing conditional access, see What is Conditional Access.

Prerequisites for Identity Manager Exchange Service

- Microsoft Windows Server 2016, or Microsoft Windows Server 2019
- Microsoft Windows Management Framework 4.0
- Microsoft Visual C++ 2017 Redistributable packages for Visual Studio

  Download the packages from the Microsoft Downloads website.
- Windows Azure AD Module for Windows Powershell on the computer where you will install Windows Powershell service.

  Perform the following steps to upgrade PowerShell to the latest version:

  1. Open a Windows PowerShell console.
  2. Run the following Install-Module cmdlet or Install-Script cmdlet:
     - If it is a module: `Install-Module -Name <moduleName> -RequiredVersion <version>`

       For example, `Install-Module -Name MSOnline -RequiredVersion 1.1.166.0`
     - If it is a script: `Install-Script -Name <scriptName> -RequiredVersion <version>`

3. Install the Exchange Online Management tool and execute the below command:

   ```
   Install-Module -Name ExchangeOnlineManagement
   ```

Identity Manager Exchange Service can be run on a user configured port. However, the service cannot be used with any other REST client tools.

Prerequisites for OAuth 2.0

The driver uses OAuth 2.0 protocol to authenticate to Azure AD. To support this protocol for authentication, you need to have a proxy application for the Azure AD driver on Azure AD. The Client ID and Client Secret allotted to the application will be later used in the Azure AD driver configuration. For more information about Azure Active Directory Application Proxy, see Microsoft Azure documentation.

Creating a Proxy Application on Azure AD

A proxy application is created in the Azure Portal. Creating a proxy application involves the following steps:

1. Registering an application and obtaining a client ID. For more information see, Registering an Application.
2. Generating an application password or the client secret. For more information see, Certificates and Secrets (https://learn.microsoft.com/en-us/entra/identity-platform/quickstart-register-app#add-credentials).
3. Configuring API permissions (Delegated and Application permissions). Set the delegated and application permissions as shown in the following table. For more information see, Add permissions to access web APIs.

API/Permission Name	Type	Description	Admin consent required
Azure Active Directory Graph (5)			
Device.ReadWrite.All	Application	Read and write devices	Yes
Directory.Read.All	Application	Read directory data	Yes
Directory.ReadWrite.All	Application	Read and write directory data	Yes
Domain.ReadWrite.All	Application	Read and write domains	Yes
Microsoft Graph (1)			
User.Read	Delegated	Sign in and read user profile	No

NOTE: You may need to provide additional permissions based on their requirement.

As a minimal requirement, the user account used in Azure AD Driver configuration should have the following Roles:

- User Administrator

- Exchange Administrator

NOTE: You may need to provide additional roles based on their requirement.

The Client ID and Client Secret can now be used for driver configurations or any other REST clients.

Assigning the Rights to the Application

1. In the server where you have installed the exchange service, login to PowerShell and connect to the Office 365 Exchange Online service, using the following command:

   ```
   Connect-MSolService
   ```

2. To obtain the Client ID for your application, replace `<AppPrincipalId>` with the Client ID that you obtained from "Creating a Proxy Application on Azure AD" on page 103 and run the following commands in PowerShell.

   ```
   Get-MsolServicePrincipal | ft DisplayName, <AppPrincipalId> -AutoSize
   ```

   ```
   $ClientIdWebApp = '<AppPrincipalId>'
   ```

   ```
   $webApp = Get-MsolServicePrincipal –AppPrincipalId $ClientIdWebApp
   ```

3. Assign the `Company Administrator` rights to your application using the Client ID obtained in Step 2 by running the following command:

   ```
   Add-MsolRoleMember -RoleName "Company Administrator" -RoleMemberType
   ServicePrincipal -RoleMemberObjectId $webApp.ObjectID
   ```

   The `Company Administrator` role will give you rights to delete the directory objects.

   IMPORTANT: The `Company Administrator` role in Identity Manager is mapped to the `Global Administrator` role in Azure AD.

Ensure that the account used by the driver to connect to the Exchange Online service has the correct roles to load and execute the following cmdlets:

- New-Mailbox
- Set-Mailbox
- Get-Mailbox
- Remove-Mailbox
- New-MailUser
- Set-MailUser
- Get-MailUser
- Remove-MailUser
- Set-User
- Get-User
- New-DistributionGroup
- Set-DistributionGroup

- Set-Group
- Get-DistributionGroup
- Get-Group
- Remove-DistributionGroup
- Add-DistributionGroupMember
- Remove-DistributionGroupMember
- Get-DistributionGroupMember
- Add-RoleGroupMember
- Remove-RoleGroupMember
- Get-RoleGroupMember
- New-UnifiedGroup
- Get-UnifiedGroup
- Set-UnifiedGroup
- Remove-UnifiedGroup
- Add-UnifiedGroupLinks
- Remove-UnifiedGroupLinks
- Get-UnifiedGroupLinks

Absence of the required roles prevents the driver from executing the cmdlets that require those roles.

Prerequisites for Support of Modern Authentication

As Microsoft Office 365 is deprecating the **Basic** authentication, you must now configure the driver with modern authentication method. You must also ensure to have the earlier mentioned prerequisites ("Prerequisites for the Driver" on page 102, "Prerequisites for Identity Manager Exchange Service" on page 102, and "Prerequisites for OAuth 2.0" on page 103) met, and then proceed with the following prerequisites.

The following prerequisites are specific to modern authentication. It is highly recommended to upgrade the driver version 5.1.x to 5.1.3 to support modern authentication.

Installing the Microsoft Exchange Online PowerShell V2 (EXO V2)

You must install the Microsoft Exchange Online PowerShell V2 module to support the new API's. For more information on EXO V2 module, see About the Exchange Online PowerShell V2 module.

- For prerequisites to install the EXO V2 module, see Prerequisites for EXO V2 module.
- For installing the EXO V2 module, see Install the EXO V2 module.

Configuring Azure AD Proxy Application for Modern Authentication Methods

You must enable the permission in the Azure portal to access Microsoft Office 365 with modern authentication.

The procedure to set the permission is shown below:

1. Login to the Azure AD Portal.
2. Select **Azure Active Directory**.
3. Navigate to **App Registration >** find and select your application in the list (for example: <*MySample_Azure_Appln*>) **> Authentication > Advanced Settings**.
4. Set **Treat Application as a Public Client** permission to **Yes**.

   IMPORTANT: The multi-factor authentication (MFA) must be disabled for the Azure account which is used with the driver.

Installing the Driver and the Identity Manager Exchange Service

The driver installation program guides you through the driver and the Identity Manager Exchange Service installation.

Perform the following actions to install and configure the Exchange Service:

1. Copy Exchange Service from `[ISO]:\products\IDM\windows\setup\drivers\azuread\ExchangeService` to any local drive on the server you intend to run this service.
2. Navigate to the directory where you copied the ExchServerHost.exe in the first step.

   For example, `C:\ExchangeService`
3. Run the following command to install the Exchange Service:

   `<location of InstallUtil>\InstallUtil.exe ExchServerHost.exe`

   For example:

   ```
   C:\Windows\Microsoft.NET\Framework64\v4.0.30319\InstallUtil.exe
   ExchServerHost.exe
   ```

   where, `C:\Windows\Microsoft.NET\Framework64\v4.0.30319\InstallUtil.exe` is the location where `InstallUtil.exe` is located.
4. Ensure the server certificate is available in iManager. To create the server certificate, see "Securing Communication with Identity Manager Exchange Service" on page 100
5. Open cmd prompt, and navigate to the local drive location where the ExchangeService is saved, as mentioned in Step 1 on page 100 (`\products\IDM\windows\setup\drivers\azuread\ExchangeService\`), and execute the command `configureExchService.bat <port> <certificate_name>`.

   For example: `configureExchService.bat 9001 azuread`. Where `9001` is the port number and `azuread` is the nickname of the certificate that was created in iManager.
6. To start the service, navigate to **Control Panel** > **Administrative Tools** > **Services**.
7. Right-click the IDMExchangeOnline service and select **Start**.

NOTE: To uninstall the service, open a .NET command prompt and issue the `InstallUtil /u` ExchServerHost.exe command.

NetIQ recommends you to use TLS 1.1 and TLS 1.2 protocols with the Identity Manager Exchange Service. If you are using ciphers and protocols such as RC4 and Triple DES, or SSLv2/v3 on a server running Identity Manager Exchange Service, you must disable them using the `disableWeakCiphers.reg` file provided in the Exchange Service installation directory. You can either execute the registry file or import the file into Windows Registry. After the changes are made, restart the server. For more information about restricting the use of certain cryptographic algorithms and protocols on Windows servers, see Microsoft Support Site.

Verifying and Starting the Identity Manager Exchange Service

After finishing the installation of Identity Manager Exchange Service, verify that the service is properly installed.

NOTE: Ensure that SSL is configured for the Identity Manager Exchange Service before starting the service. This is a mandatory step before running the service. For more information, see Securing Communication with Identity Manager Exchange Service.

1. From the **Start** menu, type **regedit**.
2. On the **Registry Editor** page, locate the service at **HKEY_LOCAL_MACHINE** > **Software** > **Novell** > **ExchServer** and verify that the **Port** and **CertificateFriendlyName** have the correct values.

   The **CertificateFriendlyName** must be the same as **Certificate Alias** that you specified in Step 1 of the "Securing Communication with Identity Manager Exchange Service" on page 100.
3. Navigate to the services that are running on your server and start the `IDMExchangeOnline` service.

Verifying the Provisioning of Exchange Mailbox

To verify the provisioned mailboxes for users, follow the procedure provided in the Microsoft Exchange Admin Center.

Creating a New Driver Object

You install the Azure AD driver files on the server where you want to run the driver, and then proceed to create the driver in the Identity Vault. You create the Azure AD driver by installing the driver packages or importing the basic driver configuration file and then modifying the driver configuration to suit your environment.

Creating the Driver Object in Designer

An administrator must use Designer to create the Azure AD driver.

The Azure AD driver is package-enabled and you must install and deploy the driver using Designer. You can use iManager to view and edit the deployed configuration. NetIQ recommends that you use Designer to perform any changes.

Importing the Driver Packages

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

1. Open Designer.
2. In the toolbar, click **Help** > **Check for Package Updates**.
3. Click **OK** to update the packages or click **OK** if the packages are up-to-date.
4. Right-click **Package Catalog** and then select **Import Package**.
5. Select any Azure AD driver packages.

   or

   Click **Select All** to import all of the packages displayed.
6. Click **OK** to import the selected packages, then click **OK** in the successfully imported package message.

After the packages are imported, continue with Installing the Driver Packages.

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

   Alternatively, you can drag and drop the Azure AD driver icon from the Cloud section of the Designer palette.
3. In the **Driver Configuration** wizard, select the `Azure AD Base` package from the list of base packages, then click **Next**.
4. Select the optional features to install for the Azure AD driver. All options are selected by default. The options are:
   - **Default Configuration:** This package contains the default configuration information for the driver. Always leave this option selected.

     NOTE: The **Azure AD Default** package and **Azure AD Exchange Default** package are included in **Default Configuration** package. By default, the **Azure AD Exchange Default** package is not selected. Select this package if you plan to use the Identity Manager Exchange Service.
   - **Entitlements and License Support:** This package contains configuration information and policies for synchronizing user accounts, group membership, roles and licenses. If you want to enable account creation and auditing through entitlements, verify that this option is selected.

     To enable the hybrid mode, select the **Azure AD Hybrid Entitlements** package. In this mode, the driver supports only Roles and License entitlements.

- **Password Synchronization:** This package contains the policies that enables the driver to synchronize passwords. If you want to synchronize passwords, ensure that this package is selected.
- **Account Tracking:** This package contains the policies that enable you to track accounts for reports. If you are using Identity Reporting, ensure that this package is selected.
- **Data Collection:** This package contains the policies that enables the driver to collect data for reports. If you are using the Identity Manager Reporting Module, ensure that this package is selected.

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. Click **OK** to install any additional package dependencies.

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

9 On the **Driver Configuration** page, fill in the following fields to configure the driver:

Authentication ID: Provide the authentication information while configuring the domain connections for the driver. This is a qualified `userprincipalname` on Azure AD with login permissions. For example, `admin@domain.onmicrosoft.com`

Password: Specify the password for the driver to authenticate to Azure AD.

Driver Options: To view the driver options, select **Show**.

Client ID: Specify the account name which the driver will use to access the Azure AD applications.

Client Secret: Specify the password for the Client ID to access the Azure AD applications.

NOTE: You created Client ID and Client Secret while creating a proxy application in Azure AD. For more information, see "Creating a Proxy Application on Azure AD" on page 103.

Show Schema Extensions Configuration: To show the schema extensions configuration options for the application (Azure AD), select **Show**.

Enable Hybrid Operation Mode: In hybrid mode, the driver provisions only roles and licenses while the users and groups are provisioned by the AD driver. By default, the parameter is set to **Yes**. If you want to run the driver in normal mode, set the option to **No**.

Activate Azure Directory Roles: By default, the driver obtains the roles that have been pre-activated in Azure Directory. If you want the driver to activate all Azure Directory roles, set this option to **Yes**. This fetches all the activated roles in Identity Applications. These roles are also available at the driver startup. Roles activation is one time activity and need not be performed again.

To obtain only pre-activated roles, leave the setting unchanged.

Existing Schema Extensions: To retain the previously-loaded configuration from Azure AD, select **Preserve**. To remove existing configuration, specify **Remove**.

Add a schema extension: Specify appropriate configuration details while adding a schema extension. You can add multiple schema extensions if required.

- **Name of extension:** Specify the name of the schema extension. For example, `Title`.

If you create multiple schema extensions with the same name, the driver uses the first extension in the list and ignores the remaining extensions that have the same name.

- **Type of extension:** Specify the data type for the configured schema extension. Ensure that the data type is a supported schema extension type in Azure AD.
- **Target objects of extension:** Lists the target objects for the schema extension. A schema extension can be extended to multiple target object classes. For example, if you have a schema extension called `Title`, it can be extended to a `User` and `Group` object classes.

NOTE: After adding the schema extension attribute, add the application attribute name to the driver filter in the following format:

`extension_<client_id>_<attribute_name>`

where <`client_id`> indicates the client ID that is used by the driver to connect to Azure AD.

For example, `extension_4691ac9cbee390e6e8e_Title`.

Subscriber Options: To view the Subscriber options, select **Show**.

Truststore file: Specify the name and path of the truststore file containing the trusted certificates used when a remote server is configured to provide server authentication. This file will contain certificates for Azure Graph and Exchange Service. For example, `c:\security\truststore`.

Proxy Host and Port: When an HTTP proxy is used, specify the host address and the host port. For example, 192.10.1.3:18180. Otherwise, leave the field blank.

Exchange and Powershell Service: When Identity Manager Exchange Service is enabled, the driver synchronizes Exchange users and groups using this service.

Exchange Service URL: Specify the URL of the Identity Manager Exchange Service.

For example, `https://<ip-addr>:<port>/ExchServer`.

Office 365 Exchange Online: To initiate a connection with Exchange Online and synchronize Office 365 exchange users and groups, select **Yes**.

Queue Operations: To enable queuing of objects when synchronizing between Azure AD and Identity Manager Exchange Service, select **True**.

Page Size: Set a value for the number of results displayed per page during Exchange Publisher poll.

Trace location: Specify the custom path where you want to save the Identity Manager Exchange Service logs. By default, the logs will be saved in this component's installation directory.

Trace Level: Set the trace level for the Identity Manager Exchange Service.

The driver supports five trace levels: NOTIFY, INFO, ERROR, MORE INFO, and DEBUG. The default trace level is NOTIFY. The next trace level, that is, INFO provides basic trace messages. ERROR provides some additional information than the previous level. Detailed messages are logged if you select INFO. DEBUG logs information on debugging data along with detailed messages.

Trace File Size Limit: Specify the trace file size limit in MB. The minimum value is 10 MB.

Database Password: Specify the database password. The driver uses this password to encrypt and connect to the Publisher cache. Ensure that the same password is used to reconnect to the cache at a later time.

Group cache clear interval: Specify the value as required to clear the exchange group related cache data. You must specify a numerical value (for example, -1, 0, 1, 2, etc.) in the Group cache clear interval field. The numerical value specified here corresponds to hours.For more information, see "Group Cache Clear Interval" on page 20.

Publisher Options: To view the Publisher options, select **Show**.

Enable Publisher: Allows you to enable or disable the Publisher connection for the driver.

Publisher Polling Interval: Specify a time period after which the driver should query Azure AD for new changes. The time is specified in minutes.

Heart Beat Interval: Allows the driver to send a periodic status message on the Publisher channel when there is no traffic for a specific duration. This indicates the time period at which the heart beat document is issued by the driver shim. The time is specified in minutes.

10 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 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 running. 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) 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.

11 On the **Azure AD Base** page, fill in the following fields, then click **Next**:

- **Domain Name:** Specify the Azure AD domain site context. For example, `<`*`domain name`*`>.onmicrosoft.com` or `<`*`domain name`*`>.com` format.
- **Identities to be synchronized:** Specify whether the driver should synchronize identities from AD or configure the Identity Vault to act as the identity provider.

  If you choose to configure the Identity Vault as an identity provider, association to any other directory is not required.

  When you choose to synchronize identity from AD, you can synchronize only users that have an association with AD. If you are using the driver in hybrid mode, select only **AD** option. This enables the driver to synchronize the identities from the Identity Vault to AD from where the identities will be synchronized to Azure AD cloud through Azure AD Connect.

- **Usage Location:** Specify the two letter country code for the user availing the Office 365 services.

12 (Conditional) On the **Install Azure AD Managed System Information** page, fill in the following fields to define the ownership of Azure AD, then click **Next**:

General Information

- **Name**: Specify a descriptive name for the managed system.
- **Description**: Specify a brief description of the managed system.
- **Location**: Specify the physical location of the managed system.
- **Vendor**: Select the vendor of the managed system.
- **Version**: Specify the version of the managed system.

System Ownership

- **Business Owner** - Select a user object in the Identity Vault that is the business owner of Azure AD. 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 Azure AD. This can only be a user object, not a role, group, or container.

  This page is only displayed if you selected to install the Data Collection packages and the Account Tracking packages.

System Classification

- **Classification**: Select the classification of Azure AD. This information is displayed in the reports. The options are as follows:
  - Mission-Critical
  - Vital
  - Not-Critical
  - Other

    If you select **Other**, you must specify a custom classification for Azure AD.
- **Environment**: Select the type of environment Azure AD provides. The options are as follows:
  - Development
  - Test
  - Staging
  - Production
  - Other

    If you select **Other**, you must specify a custom environment for Azure AD.

13 On the **Azure AD Password Synchronization** page, fill in the following fields, then click **Next**:

- **Set Password Never Expires:** If you set this option to **True** on the newly created users, the password does not expire for those users.
- **Disable Force Change Password at First Login:** If you set this option to **True**, a user is not prompted to change the password when the user logs in to Azure AD for the first time.
- **Set Strong Password Required:** If you set this option to **True**, the user needs to set a strong password.

14 On the **Account Tracking** page, specify the name of the realm, security domain, or namespace in which the account name is unique. You must set the Realm to the Azure AD domain name.

15 On the **Confirm Installation Tasks** page, review the summary of tasks and click **Finish**.

The driver is now created. You can modify the configuration settings by Configuring the Driver.

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.

To edit the properties, perform the following steps:

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.

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

NOTE: These settings are only displayed if you installed the Entitlements package. If you selected the **Azure AD Hybrid Entitlements** package, only Roles and License entitlements are supported with this 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 **False**, user and group membership synchronization is managed using the non-entitlement configuration method.

- **Use License Entitlement:** Ensure the value of this parameter is set to true to enable the driver to manage licenses using the License entitlement. By default, the value is set to **True**.
- **Use Roles Entitlement:** Ensure the value of this parameter is set to true to enable the driver to manage roles using the Roles entitlement. By default, the value is set to **True**.

5 Click **Apply**.

6 Modify any other settings as necessary.

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 Azure AD, your synchronization requirements for the driver might differ from the default policies. If this is the case, you require customization.

7 Click **OK** when finished.

8 Continue with "Deploying the Driver" on page 114.

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, 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 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, then select **Live** > **Start Driver**.

Creating the Driver Object in Designer

An administrator must use Designer to create the Azure AD driver.

The Azure AD driver is package-enabled and you must install and deploy the driver using Designer. You can use iManager to view and edit the deployed configuration. NetIQ recommends that you use Designer to perform any changes.

Activating the Driver

The Identity Manager driver for Office 365 and Azure AD is part of the Identity Manager Integration Module for Microsoft Enterprise.

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

Upgrading an existing Driver

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

- "What's New in Version 5.1.0.0?" on page 115
- "Working with MapDB 3.0.5" on page 115
- "Upgrading the Driver" on page 116

What's New in Version 5.1.0.0?

Identity Manager 4.7 provides support for MapDB 3.0.5. To ensure that your driver works correctly with Identity Manager 4.7 engine, see Working with MapDB 3.0.5.

Working with MapDB 3.0.5

NetIQ recommends that you review the following sections before upgrading your driver to work with Identity Manager 4.7 engine:

- "Understanding Identity Manager 4.7 Engine Support for Driver Versions" on page 116
- "Manually Removing the MapDB Cache Files" on page 116

Understanding Identity Manager 4.7 Engine Support for Driver Versions

- Drivers shipped with Identity Manager 4.7 are compatible with Identity Manager 4.7 Engine or Remote Loader. You must perform the following actions to complete the driver upgrade:
  1. Upgrade the Identity Manager Engine.
  2. (Conditional) Upgrade the Remote Loader.
  3. Upgrade the driver.
  4. Manually remove the MapDB state cache files from the Identity Vault's DIB directory. For more information, see "Manually Removing the MapDB Cache Files" on page 116.
- Drivers shipped before Identity Manager 4.7 are not compatible with Identity Manager 4.7 Engine or Remote Loader.
- Drivers shipped with Identity Manager 4.7 are not backward compatible with Identity Manager 4.6.x Engine or Remote Loader.
- Drivers shipped with Identity Manager 4.7 are not backward compatible with Identity Manager 4.5.x Engine or Remote Loader.

Manually Removing the MapDB Cache Files

The Identity Manager engine upgrade process removes the existing MapDB driver work cache files (`dx*`) from the Identity Vault's DIB directory (`/var/opt/novell/eDirectory/data/dib` or `C:\Novell\NDS\DIBFiles`). You must manually remove the existing MapDB state cache files for the driver after upgrading Identity Manager and the driver from a version prior to Identity Manager 4.7:

```
<Azure driver name>_obj.db.*
```

where ***** is the name of the state cache file for the driver. For example, `<Azure driver name>_obj.db.t` or `<Azure driver name>_obj.db.p`

This action ensures that your driver works correctly with Identity Manager 4.7 engine.

Upgrading the Driver

The driver upgrade process involves upgrading the installed driver packages and updating the existing driver files. These are independent tasks and can be separately planned for a driver. For example, you can update the driver packages and choose not to update the driver files at the same time. However, you are recommended to complete all the update steps within a short amount of time to ensure that the driver has the latest updates.

- "Upgrading the Installed Packages" on page 116
- "Updating the Driver Files" on page 117

Before starting the upgrade process, ensure that you have taken a back-up of the current driver configuration.

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. However, if you need to add a custom package to the Package Catalog, you can import the package `.jar` file. For more information about creating custom packages, see Developing Packages in 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*.

Updating the Driver Files

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

To update the driver files:

1 Stop the driver instance by using iManager, Designer, or dxcmd by performing one of the following actions:

- If the driver is running locally, stop the driver instance and the Identity Vault.
- If the driver is running with a Remote Loader instance, stop the driver and the Remote Loader instance.

For example, go to a command prompt on Linux and run `ndsmanage stopall`

2 Download the driver patch file to a temporary folder on your server.

3 Extract the contents of the driver patch file.

4. Update the driver files:
   - **Linux:** Open a command prompt and run the following command to upgrade the existing RPM:

     ```
     rpm -Uvh <Driver Patch File Temporary Location>/linux/netiq-DXMLRESTAzure.rpm
     ```

   - **Windows:** Navigate to the <*Extracted Driver Patch File Temporary Location*>\windows folder and perform the following actions:
     - Copy the `DXMLRESTAzureConfig.jar` and `DXMLRESTAzureShim.jar` files to `<IdentityManager installation>\NDS\lib` or `<IdentityManager installation>\RemoteLoader\<architecture>\lib` folder.
     - Copy the `DXMLRESTAzureUtil.jar` file to `<IdentityManager installation>\DirXMLUtilities\restazure\util` folder.
5. (Conditional) If the driver is running locally, start the Identity Vault and the driver instance.

   For example, open a command prompt on Linux and run `ndsmanage startall`
6. (Conditional) If the driver is running with a Remote Loader, start the Remote Loader and the driver instance.

6 Procedure for upgrading from Azure AD Driver 5.1.5 to 5.1.6 or later

NOTE:
- Microsoft has announced the retirement of Azure AD and Msol PowerShell modules. MS has recommended using Microsoft graph based PowerShell cmdlets instead. For more information, see: Upgrade from Azure AD PowerShell to Microsoft Graph PowerShell. (https://learn.microsoft.com/en-us/powershell/microsoftgraph/migration-steps?view=graph-powershell-1.0)
- It is highly recommended to perform the Driver Upgrade by taking downtime so that there no changes in Azure are lost while the driver is getting upgraded. Otherwise, after the upgrade, it will be necessary to migrate the Users and Groups into Identity Vault to ensure that no changes in Azure were lost.

The driver upgrade process involves upgrading the installed driver packages and updating the existing driver files. These are independent tasks and can be separately planned for a driver. For example, you can update the driver packages and choose not to update the driver files at the same time. However, you are recommended to complete all the update steps within a short amount of time to ensure that the driver has the latest updates.

- "How to upgrade to Azure AD Driver 5.1.6" on page 119
- "Upgrading the Installed Packages" on page 121

Before starting the upgrade process, ensure that you have taken a back-up of the current driver configuration.

How to upgrade to Azure AD Driver 5.1.6

Below are the steps listed to upgrade the Azure AD Driver of version 5.1.6.

1 Stop the IDM Server and driver on RL (remote Loader).

2 Update the existing Azure AD Driver Files. For more information, see Step 4 on page 74 for Linux and Step 5 on page 74 for Windows.

3 Upgrade the Azure AD Designer Packages using Designer. For more information see,"Upgrading the Installed Packages" on page 130.

 3a Application Schema changes for MS Graph APIs.

 3b Schema Mapping Policy. For the complete list of schema changes, refer to Property differences between Azure AD Graph and Microsoft Graph (https://docs.microsoft.com/en-us/graph/migrate-azure-ad-graph-property-differences)

 3c Policies using the old schema.

NOTE: If there are any customizations on the above policies then, the new changes coming from package need to be manually merged. Similarly, if there are any new policy created by you using the old schema name then, you need to change them.

4 Configure the authentication type in driver parameters.

5 Configure API permissions in Azure AD Application.

NOTE: In addition to the API permissions configured for Azure AD Driver 5.1.5, you need to add following new permissions.

Table 6-1 *List of Application type of APIs*

API	Type	Description	Admin Consent
Channel.Create	Application	Create channels	Grant Admin Consent
Channel.Delete.All	Application	Delete channels	Grant Admin Consent
Channel.ReadBasic.All	Application	Read the names and descriptions of all channels	Grant Admin Consent
ChannelMember.Read.All	Application	Read the members of all channels	Grant Admin Consent
ChannelMember.ReadWrite.All	Application	Add and remove members from all channels	Grant Admin Consent
ChannelSettings.Read.All	Application	Read the names, descriptions, and settings of all channels	Grant Admin Consent
ChannelSettings.ReadWrite.All	Application	Read and write the names, descriptions, and settings of all channels	Grant Admin Consent
Team.Create	Application	Create teams	Grant Admin Consent
Team.ReadBasic.All	Application	Get a list of all teams	Grant Admin Consent
TeamMember.Read.All	Application	Read the members of all teams	Grant Admin Consent
TeamMember.ReadWrite.All	Application	Add and remove members from all teams	Grant Admin Consent
TeamMember.ReadWrite NonOwnerRole.All	Application	Add and remove members with non-owner role for all teams	Grant Admin Consent
TeamSettings.Read.All	Application	Read all teams' settings	Grant Admin Consent
TeamSettings.ReadWrite.All	Application	Read and change all teams' settings	Grant Admin Consent

6 Configure Secure Communication. For more information see Securing Driver Communication

7 Start the IDM Server and driver on RL.

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. However, if you need to add a custom package to the Package Catalog, you can import the package `.jar` file. For more information about creating custom packages, see Developing Packages in 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*.

7 Procedure for upgrading to Azure AD Driver 5.1.5

NOTE: It is highly recommended to perform the Driver Upgrade by taking downtime so that there no changes in Azure are lost while the driver is getting upgraded. Otherwise, after the upgrade, it will be necessary to migrate the Users and Groups into Identity Vault to ensure that no changes in Azure were lost.

The driver upgrade process involves upgrading the installed driver packages and updating the existing driver files. These are independent tasks and can be separately planned for a driver. For example, you can update the driver packages and choose not to update the driver files at the same time. However, you are recommended to complete all the update steps within a short amount of time to ensure that the driver has the latest updates.

- "How to upgrade to Azure AD Driver 5.1.5" on page 123
- "Upgrading the Installed Packages" on page 125
- "Post Upgrade Tasks" on page 126

Before starting the upgrade process, ensure that you have taken a back-up of the current driver configuration.

How to upgrade to Azure AD Driver 5.1.5

Below are the steps listed to upgrade the Azure AD Driver of version 5.1.5.

1 Stop the IDM Server and driver on RL (remote Loader).

2 Update the existing Azure AD Driver Files. For more information, see Step 4 on page 96 for Linux and Step 5 on page 97 for Windows.

3 Upgrade the Azure AD Designer Packages using Designer. For more information see,"Upgrading the Installed Packages" on page 130.

   3a Application Schema changes for MS Graph APIs

   3b Schema Mapping Policy. For the complete list of schema changes, refer to Property differences between Azure AD Graph and Microsoft Graph (https://docs.microsoft.com/en-us/graph/migrate-azure-ad-graph-property-differences)

   3c Policies using the old schema.

   NOTE: If there are any customizations on the above policies then, the new changes coming from package need to be manually merged. Similarly, if there are any new policy created by you using the old schema name then, you need to change them.

4 Configure the authentication type in driver parameters.

5 Configure API permissions in Azure AD Application.

Table 7-1 List of Application type of APIs

API	Type	Description	Admin Consent
Application.Read.All	Application	Read all applications	Grant Admin Consent
Application.ReadWrite.All	Application	Read and write all applications	Grant Admin Consent
Device.Read.All	Application	Read all devices	Grant Admin Consent
Device.ReadWrite.All	Application	Read and write devices	Grant Admin Consent
Directory.Read.All	Application	Read directory data	Grant Admin Consent
Directory.ReadWrite.All	Application	Read and write directory data	Grant Admin Consent
Domain.ReadWrite.All	Application	Read and write domains	Grant Admin Consent
Group.Create	Application	Create groups	Grant Admin Consent
Group.Read.All	Application	Read all groups	Grant Admin Consent
Group.ReadWrite.All	Application	Read and write all groups	Grant Admin Consent
GroupMember.Read.All	Application	Read all group memberships	Grant Admin Consent
GroupMember.ReadWrite.All	Application	Read and write all group memberships	Grant Admin Consent
User.Read.All	Application	Read all users' full profiles	Grant Admin Consent
User.ReadWrite.All	Application	Read and write all users' full profiles	Grant Admin Consent

Table 7-2 List of Delegated type of APIs

API	Type	Description	Admin Consent
Directory.AccessAsUser.All	Delegated	Access directory as the signed in user	Grant Admin Consent
Directory.Read.All	Delegated	Read directory data	Grant Admin Consent
Directory.ReadWrite.All	Delegated	Read and write directory data	Grant Admin Consent
Group.Read.All	Delegated	Read all groups	Grant Admin Consent
Group.ReadWrite.All	Delegated	Read and write all groups	Grant Admin Consent
GroupMember.Read.All	Delegated	Read group memberships	Grant Admin Consent
GroupMember.ReadWrite.All	Delegated	Read and write group memberships	Grant Admin Consent

API	Type	Description	Admin Consent
RoleManagement.Read.All	Delegated	Read role management data for all RBAC providers	Grant Admin Consent
RoleManagement.Read.Directory	Delegated	Read directory RBAC settings	Grant Admin Consent
User.Read	Delegated	Sign in and read user profile	Grant Admin Consent
User.Read.All	Delegated	Read all users' full profiles	Grant Admin Consent
User.ReadBasic.All	Delegated	Read all users' basic profiles	Grant Admin Consent
User.ReadWrite	Delegated	Read and write access to user profile	Grant Admin Consent
User.ReadWrite.All	Delegated	Read and write all users' full profiles	Grant Admin Consent

6 Configure Secure Communication. For more information see Securing Driver Communication

7 Perform Post Upgrade Tasks. For more information see "Post Upgrade Tasks" on page 131

8 Start the IDM Server and driver on RL.

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. However, if you need to add a custom package to the Package Catalog, you can import the package `.jar` file. For more information about creating custom packages, see Developing Packages in 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*.

Post Upgrade Tasks

1. Clearing the DirXML-DriverStorage Attribute.

   The dirxml-DriverStorage attribute is a "per-replica" attribute. So you must delete it on the replica server associated with the driverset. To accomplish this, while launching the Generic LDAP browser or iManager, instead of providing the treename during authentication, provide the IP address of the server associated with the driverset in the tree name field. This allows you to delete the attribute from the replica server associated with the driverset.

   NOTE: A new marker will be created post upgrade driver starts, since it uses a different API to connect to Azure. This information is stored in the attribute DirXML-DriverStorage. For this reason, the driver won't be able to capture events while the driver is being updated.

2. Start the Driver.
3. If you have not taken downtime as mentioned earlier then, it is mandatory to migrate the Users and Groups into Identity Vault to ensure that no changes in Azure are lost.

8 Procedure for upgrading from Azure AD Driver 5.1.4 or earlier to 5.1.6 or later

NOTE:
- Microsoft has announced the retirement of Azure AD and Msol PowerShell modules. MS has recommended using Microsoft graph based PowerShell cmdlets instead. For more information, see: Upgrade from Azure AD PowerShell to Microsoft Graph PowerShell. (https://learn.microsoft.com/en-us/powershell/microsoftgraph/migration-steps?view=graph-powershell-1.0)
- It is highly recommended to perform the Driver Upgrade by taking downtime so that there no changes in Azure are lost while the driver is getting upgraded. Otherwise, after the upgrade, it will be necessary to migrate the Users and Groups into Identity Vault to ensure that no changes in Azure were lost.

The driver upgrade process involves upgrading the installed driver packages and updating the existing driver files. These are independent tasks and can be separately planned for a driver. For example, you can update the driver packages and choose not to update the driver files at the same time. However, you are recommended to complete all the update steps within a short amount of time to ensure that the driver has the latest updates.

- "How to upgrade to Azure AD Driver 5.1.6" on page 127
- "Upgrading the Installed Packages" on page 130
- "Post Upgrade Tasks" on page 131

Before starting the upgrade process, ensure that you have taken a back-up of the current driver configuration.

How to upgrade to Azure AD Driver 5.1.6

Below are the steps listed to upgrade the Azure AD Driver of version 5.1.6.

1 Stop the IDM Server and driver on RL (remote Loader).

2 Update the existing Azure AD Driver Files. For more information, see Step 4 on page 74 for Linux and Step 5 on page 74 for Windows.

3 Upgrade the Azure AD Designer Packages using Designer. For more information see,"Upgrading the Installed Packages" on page 130.

3a Application Schema changes for MS Graph APIs

3b Schema Mapping Policy. For the complete list of schema changes, refer to Property differences between Azure AD Graph and Microsoft Graph (https://docs.microsoft.com/en-us/graph/migrate-azure-ad-graph-property-differences)

3c Policies using the old schema.

NOTE: If there are any customizations on the above policies then, the new changes coming from package need to be manually merged. Similarly, if there are any new policy created by you using the old schema name then, you need to change them.

4 Configure the authentication type in driver parameters.

5 Configure API permissions in Azure AD Application.

***Table 8-1** List of Application type of APIs*

API	Type	Description	Admin Consent
Application.Read.All	Application	Read all applications	Grant Admin Consent
Application.ReadWrite.All	Application	Read and write all applications	Grant Admin Consent
Channel.Create	Application	Create channels	Grant Admin Consent
Channel.Delete.All	Application	Delete channels	Grant Admin Consent
Channel.ReadBasic.All	Application	Read the names and descriptions of all channels	Grant Admin Consent
ChannelMember.Read.All	Application	Read the members of all channels	Grant Admin Consent
ChannelMember.ReadWrite.All	Application	Add and remove members from all channels	Grant Admin Consent
ChannelSettings.Read.All	Application	Read the names, descriptions, and settings of all channels	Grant Admin Consent
ChannelSettings.ReadWrite.All	Application	Read and write the names, descriptions, and settings of all channels	Grant Admin Consent
Device.Read.All	Application	Read all devices	Grant Admin Consent
Device.ReadWrite.All	Application	Read and write devices	Grant Admin Consent
Directory.Read.All	Application	Read directory data	Grant Admin Consent
Directory.ReadWrite.All	Application	Read and write directory data	Grant Admin Consent
Domain.ReadWrite.All	Application	Read and write domains	Grant Admin Consent
Group.Create	Application	Create groups	Grant Admin Consent
Group.Read.All	Application	Read all groups	Grant Admin Consent
Group.ReadWrite.All	Application	Read and write all groups	Grant Admin Consent
GroupMember.Read.All	Application	Read all group memberships	Grant Admin Consent

API	Type	Description	Admin Consent
GroupMember.ReadWrite.All	Application	Read and write all group memberships	Grant Admin Consent
Team.Create	Application	Create teams	Grant Admin Consent
Team.ReadBasic.All	Application	Get a list of all teams	Grant Admin Consent
TeamMember.Read.All	Application	Read the members of all teams	Grant Admin Consent
TeamMember.ReadWrite.All	Application	Add and remove members from all teams	Grant Admin Consent
TeamMember.ReadWrite NonOwnerRole.All	Application	Add and remove members with non-owner role for all teams	Grant Admin Consent
TeamSettings.Read.All	Application	Read all teams' settings	Grant Admin Consent
TeamSettings.ReadWrite.All	Application	Read and change all teams' settings	Grant Admin Consent
User.Read.All	Application		Grant Admin Consent
User.ReadWrite.All	Application	Read and write all users' full profiles	Grant Admin Consent

Table 8-2 *List of Delegated Type of APIs*

API	Type	Description	Admin Consent
Directory.ReadWrite.All	Delegated	Read and write directory data	Grant Admin Consent
Group.Read.All	Delegated	Read all groups	Grant Admin Consent
Group.ReadWrite.All	Delegated	Read and write all groups	Grant Admin Consent
GroupMember.Read.All	Delegated	Read group memberships	Grant Admin Consent
GroupMember.ReadWrite.All	Delegated	Read and write group memberships	Grant Admin Consent
RoleManagement.Read.All	Delegated	Read role management data for all RBAC providers	Grant Admin Consent
RoleManagement.Read.Directory	Delegated	Read directory RBAC settings	Grant Admin Consent
User.Read	Delegated	Sign in and read user profile	Grant Admin Consent
User.Read.All	Delegated	Read all users' full profiles	Grant Admin Consent
User.ReadBasic.All	Delegated	Read all users' basic profiles	Grant Admin Consent

API	Type	Description	Admin Consent
User.ReadWrite	Delegated	Read and write access to user profile	Grant Admin Consent
User.ReadWrite.All	Delegated	Read and write all users' full profiles	Grant Admin Consent

6 Configure Secure Communication. For more information see Securing Driver Communication

7 Perform Post Upgrade Tasks. For more information see "Post Upgrade Tasks" on page 131

8 Start the IDM Server and driver on RL.

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. However, if you need to add a custom package to the Package Catalog, you can import the package `.jar` file. For more information about creating custom packages, see Developing Packages in 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*.

Post Upgrade Tasks

1 Clearing the DirXML-DriverStorage Attribute.

The dirxml-DriverStorage attribute is a "per-replica" attribute. So you must delete it on the replica server associated with the driverset. To accomplish this, while launching the Generic LDAP browser or iManager, instead of providing the treename during authentication, provide the IP address of the server associated with the driverset in the tree name field. This allows you to delete the attribute from the replica server associated with the driverset.

NOTE: A new marker will be created post upgrade driver starts, since it uses a different API to connect to Azure. This information is stored in the attribute DirXML-DriverStorage. For this reason, the driver won't be able to capture events while the driver is being updated.

2 Start the Driver.

3 If you have not taken downtime as mentioned earlier then, it is mandatory to migrate the Users and Groups into Identity Vault to ensure that no changes in Azure are lost.

9 Transitioning from Existing Office 365 Driver to New Azure AD Driver

The Identity Manager driver for Office 365 and Azure Active Directory introduces significant architectural changes over the existing Office 365 driver. The driver provides the same and more functionality through an improved design that is more efficient and easier to configure. To learn more about the new architecture, see "Understanding How the Driver Works" on page 13.

Given these changes, in order to use the new driver, you need to migrate users and groups from Office 365 to Azure AD. The following sections help you accomplish this:

- "Preparing for Migrating Identities from Azure AD to Identity Vault" on page 133
- "Migrating Identities" on page 133
- "Transitioning Assignments Through User Application" on page 134

Preparing for Migrating Identities from Azure AD to Identity Vault

NetIQ recommends that you perform the migration in a test environment similar to your production environment before upgrading the production systems.

Before you begin, ensure that the following prerequisites are met:

- Turn off Exchange service and entitlements before starting the migration.
- Ensure that there is a valid matching attribute for user and group objects in the Identity Vault. You need to create a matching policy that includes a matching attribute so that you can do a one-to-one mapping. When a match is found, an association is created. For example, the `cn` attribute for a user in the Identity Vault is mapped to `UserPrincipalname` attribute in Azure AD. Similarly, the `cn` attribute for a group in Identity Vault is mapped to `displayName` attribute in Azure AD.

Migrating Identities

This process involves migration of all users and groups from Azure AD into the Identity Vault.

1. In iManager, click **Identity Manager** > **Identity Manager Overview**.
2. Browse to and select the driver set.
3. Click the Azure AD driver icon.
4. In the **Identity Manager Driver Overview** page, select **Migrate** and then **Migrate into Identity Vault**.

NOTE: You cannot perform a wild-card search to query users and groups for migration. You must select the user or group class to migrate.

5 In the **Edit List** window, select **User** class and click **OK**.

Similarly, to migrate Azure AD groups, select **Groups** class and click **OK.**

This process will also update the association for the migrated objects.

Transitioning Assignments Through User Application

As there are significant architectural changes between the existing Office 365 driver and the Azure AD driver, you need to recreate the existing Office 365 resources in the Azure AD driver. The following considerations apply while transitioning the existing Office 365 assignments to the Azure AD driver:

Recreate the Office 365 driver resources for the Azure AD driver

You can use the existing Office 365 resources as a reference to create the resources manually and then map them appropriately to the existing Office 365 roles. For example, you have an existing role in Identity Applications called `IT_Admin_O365_Role` and the role is mapped to `O365_MailboxAdmin`, `O365_SecurityAdmin`, and `O365_SharePointServiceAdmin` resources.

To transition the role assignments from existing Office 365 driver to Azure AD driver, you need to create similar resources for the Azure AD driver and then map them appropriately to existing `IT_Admin_O365_Role` role. For more information about creating roles and resources, see Creating and Managing Roles in the *NetIQ Identity Manager - Administrator’s Guide to the Identity Applications*.

The following procedure explains how to create a new resource in Azure AD, assign an entitlement value to the resource, and map the resource to an existing Office 365 role in Identity Applications.

To create Azure AD resource and assign an entitlement value to the resource:

1. Turn on entitlements for the Azure AD driver.
2. Create a new resource.

   Open a Web browser and log in to Identity Applications. For example: `http://myappserver:8543/idmdash/`
3. Go to **Administration > Resources** and click the **+** icon.
4. Select **With entitlement**.
5. In **Entitlement or Driver** list, select the Azure AD driver.
6. In **Entitlement Association**, select `Mailbox Administrator` from the list.
7. Click **Create Resource**.
8. Specify the required values such as **Resource Name** and **Resource Description** to create a new resource with entitlement for the Azure AD driver. Click **Apply**.

You must also create resources for other roles. For example, `Security Admin`, and `SharePointService Admin`.

To map the newly created resource to an existing Office 365 role:

1. Go to **Administration > Roles**.

2. Select the Office 365 role from the list. For example, `IT_Admin_O365_Role`.
3. Select **Map Resource to Role**.
4. In **Available for Mapping > Resources**, drag and drop the newly created Azure AD resource to **Mapped Resources**.
5. (Conditional) If a resource request form is configured, specify the necessary information and click **Apply**.
6. Specify the **Mapped Description** and click **Apply**.

Manually assign permissions on the newly created resources

If you have resources with direct assignments (resources not mapped to any role), then manually assign the permissions appropriately on the newly created resources for the Azure AD driver.

The procedure to assign permissions manually is shown below:

1. Go to **Administration > Resources**.
2. Select the newly created Azure AD resource.
3. select **Resource Assignments**.
4. Click **+** to assign to the required users in the system.

10 Understanding Identity Manager Exchange Service

Identity Manager Exchange Service is a REST-based Windows service to support Exchange Online. The Azure AD driver leverages this service to provision or deprovision user mailboxes, mail users, create or remove distribution lists and security groups on Office 365 Exchange Online. This service converts the driver REST calls to Exchange Online cmdlets to manage Exchange Online.

When the Azure AD driver starts, it initializes the service by sending information to Office 365 such as exchange domain, user name, and password. The Azure AD driver is properly initialized only if the system time is synchronized between the servers running the driver and the Exchange Online service.

The schema includes the following attributes to support Office 365 Exchange Online:

- **DirXML-AADObjectType:** Contains the type for a user or a group object.

Name	Description
UserMailbox	Creates a mailbox user in Exchange Online
MailUser	Creates a mail user in Exchange Online
Distribution	Creates a distribution group in Exchange Online
Security	Creates a security group in Exchange Online
UnifiedGroup	Creates a Office 365 group in Exchange Online

  For example, to add a mail user, set the **DirXML-AADObjectType** attribute to `MailUser`. To create an Exchange group, set this attribute to `Distribution` or `Security`.
- **DirXML-AADArchiveStatus:** Contains the mailbox archive status for an Exchange Online user.
- **DirXML-AADLitigationHoldEnabled:** Contains the mailbox litigation hold status for an Exchange Online user.
- **DirXML-AADLegacyExchangeDN:** Contains the Exchange server DN for a mailbox.

If you are not using Exchange Online, these attributes are not required.

The service also supports execution of PowerShell cmdlets that are part of XDS as values of `psexecute` attribute.

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 the cmdlets to the Azure AD driver using policies.

IMPORTANT: The PowerShell commands should be wrapped in double quotes to pass a value to `psexecute`. Identity Manager uses double quotes, but PowerShell prefers single quotes.

For example:

```
<modify-attr attr-name="psexecute">
 <add-value>
  <value type="string">Get-Process</value>
 </add-value>
</modify-attr>
```

NOTE: For PowerShell reference, use lowercase format. For example, `psexecute`.

For more information about PowerShell, see the following resources:

- "Getting Started with Windows PowerShell" (http://technet.microsoft.com/en-us/library/aa973757%28VS.85%29.aspx)
- "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)

11 Troubleshooting

Refer to the following sections if you are experiencing a problem with the Azure AD driver.

- "Synchronizing country and usageLocation Attributes" on page 139
- "Azure AD Password Complexity" on page 140
- "Restoring the Driver to Current State" on page 140
- "No Trusted Certificate Found Exception" on page 141
- "Exchange Error During Driver Restart" on page 141
- "Setting the set-executionPolicy to RemoteSigned in the Powershell" on page 141
- "Email Address is Set Incorrectly for Groups that are Provisioned to a Different Valid Domain" on page 141
- "Revoking Roles and Licenses in Hybrid Mode" on page 141
- "Setting Primary SMTP Address With EmailAddress Attribute Displays An Error" on page 142
- "Mapping company Attribute with companyName Attribute Displays An Error" on page 142
- "Issue with the Size of PowerShell Log File" on page 142
- "License Dependency in Developer Pack" on page 142
- "User Name Cannot Contain Some Special Characters" on page 142
- "Restoring a Mailbox or Mail User Displays a Warning Message" on page 143
- "Random Errors While Connecting to the Exchange Portal" on page 143
- "Adding a Graph User to Group Fails" on page 144
- "No Trusted Certificate Found Exception" on page 144
- "Driver Fails to Connect to Microsoft Graph API Due to Invalid Certificate Error" on page 144
- "Driver Fails to Connect to Office 365 due to the MS Exchange License Issues" on page 145
- "Driver Shuts Down if Role Assignment Fails in Azure Application" on page 145
- "Channel Exception on Too Many Requests" on page 146
- "Error While Adding Users as Members to a Group in Azure" on page 146
- "Certificate based authentication fails with "Refresh token and Client Secret must be provided to get the new Access Token exception" Error when driver is configured to use proxy server" on page 146

Synchronizing country and usageLocation Attributes

You can set the following attributes while using eDirectory to select a country for a user:

Attribute	Description
C	Mapped with usageLocation attribute of Azure AD. It contains a two-character country code as defined by the ISO.
co	Mapped with country attribute of Azure AD. It contains a longer name for the country.

Since the ISO-defined character country codes are intended to be used by the Azure AD licenses, the default schema in the Identity Vault includes co and C attributes.

Azure AD Password Complexity

Passwords must adhere to the Azure AD password requirements. A user cannot be added if the specified password does not meet these requirements.

Complexities and requirements in Azure AD 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 Azure AD and eDirectory.

For information about Azure AD password complexity requirements, see "Password must meet complexity requirements".

For information about managing passwords in eDirectory, see the *Password Management Administration Guide*.

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

Restoring the Driver to Current State

If you need to reset the Publisher state of the driver to prevent the driver from picking up interim events and performing unwanted actions on the Identity Vault, perform the following steps after stopping the driver.

1. Delete the **Dirxml-DriverStorage** attribute on the driver object in the Identity Vault.
2. If the driver is remotely loaded, delete the state file from the Remote Loader server.
3. Set the driver to **Manual** or **Automatic** startup.
4. Start the driver.

No Trusted Certificate Found Exception

This issue occurs when the certificate is changed at the Azure Graph endpoint.

To workaround this issue, import the new public certificate into the trust store of the driver. For more information, see "Securing Communication with Azure AD Graph" on page 99.

- Issue: No trusted certificate found
  - Resolution: Refer to Secure communication with MS Graph API and make sure all required certificates are downloaded and imported into the keystore

Exchange Error During Driver Restart

Except the case of invalid credentials, when the driver fails to connect to MSOL, the driver makes three attempts to connect to MSOL with an interval of 30 seconds after each attempt.

If the driver fails to connect to MSOL after three attempts, the driver shuts down.

Setting the set-executionPolicy to RemoteSigned in the Powershell

The `set-ExecutionPolicy` cmdlet enables you to determine the PowerShell scripts that can be run on your computer.

By default, `set-ExecutionPolicy` is set to **Restricted**. To start the driver, change the setting to **RemoteSigned** in PowerShell. Otherwise, the driver fails to start and displays an error message.

Email Address is Set Incorrectly for Groups that are Provisioned to a Different Valid Domain

When you configure the driver to a different valid domain and add an exchange group on the Subscriber channel, the email address is set incorrectly for the group on the exchange portal.

Workaround: Add `PrimarySmtpAddress` to the driver filter and add `PrimarySmtpAddress` for the group on the Subscriber channel.

Revoking Roles and Licenses in Hybrid Mode

When the Azure AD driver is running in a hybrid mode and a user's account permission is revoked using the AD driver, the account is either disabled or deleted in AD and the corresponding association is removed from the Identity Vault. This action also triggers AAD Connect to disable or delete the user from Azure AD. However, this action does not revoke user's Roles and License assignments in the User Application.

Workaround: Manually remove the Roles and License assignments for the user from the User Application.

Setting Primary SMTP Address With EmailAddress Attribute Displays An Error

When you set the primary SMTP address from **EmailAddress** attribute, an error message is displayed.

Workaround: To synchronize the primary SMTP address, create a custom attribute in eDirectory and map it with the **PrimarySmtpAddress** attribute.

Mapping company Attribute with companyName Attribute Displays An Error

The driver does not allow mapping the **company** attribute with the **companyName** attribute and reports an error.

Workaround: Change the application's attribute name from **companyName** to **Company**. The Azure AD driver treats **companyName** as an Azure Graph API attribute and **Company** as an exchange attribute.

Issue with the Size of PowerShell Log File

By default, the PowerShell log file is located at `\AppData\Local\Microsoft\Office365\Powershell`. When you execute `MSOnline` cmdlets in the Exchange service, the log file increases in size.

Workaround: To prevent the logs from getting included in the file, restrict the rights to the directory containing the log file.

License Dependency in Developer Pack

The Office 365 portal does not allow you to revoke an individual license if it has a dependency on other licenses. For example, an individual license is dynamically allocated to a resource in the User Application.

To workaround this issue, remove the dependent licenses before attempting to revoke the individual license. For example, if `Office Online for Developer` license is dependent on `SharePoint Online for Developer` license, you need to revoke `Office Online for Developer` license before you revoke `SharePoint Online for Developer` license.

User Name Cannot Contain Some Special Characters

While adding a user, if the user name includes some special characters that Azure AD does not support in **userprincipalname**, the user addition fails with an error. For details about the characters that are not supported, see the Microsoft web site.

You can customize your policies to remove the unsupported characters from the attributes.

Restoring a Mailbox or Mail User Displays a Warning Message

When you restore a mailbox or mail user, the following message is displayed:

```
Operation Failed: Get-User: The operation couldn't be performed because
object couldn't be found
```

This issue does not cause any functionality loss. You can ignore the message and continue with restoring the mailbox or mail user.

Random Errors While Connecting to the Exchange Portal

You might encounter the following errors while connecting to the Exchange portal:

Driver fails to connect to the Exchange portal

The driver shuts down with the following error message.

```
There was no endpoint listening that could accept the message.
```

This issue is observed when the Exchange portal is down.

Workaround: Start the driver when the Exchange portal is functional.

Driver stops with a fatal error

In iManager, when you modify any driver parameter and try to restart the driver, the driver shuts down with the following error message:

```
IOException: graph.windows.net: Name or service not known
```

Workaround: Start the driver when the Exchange portal is functional.

Azure AD Cloud reports java.net.UnknownHostException: login.windows.net error

This issue is randomly observed.

If you restart the driver, the driver displays the following error, when the **Office 365 Exchange Online** parameter is set to **No** and Publisher channel is disabled.

```
java.net.UnknownHostException: login.windows.net
```

This error is reported from Azure AD. As there is no functionality loss, start the driver after few minutes.

Random Error Messages in Exchange Trace

When you run the driver with Exchange Online enabled, error messages are reported in the Exchange trace. For example:

```
publish: Operation Failed: Starting a command on the remote server failed
with the following error message. The I/O operation has been aborted
because of either a thread exit or an application request
```

Ignore the error message as it does not cause any functionality loss.

- Issue: Errors in sync related to privileges
  - Resolution: Check for API permissions
- Issue: Publisher Polling not fetching delta
  - Resolution: Stop driver, delete the DirXML-DriverStorage attribute and then start the driver.
- Issue: Errors related to Authorization
  - Resolution: Check for credentials, Client ID and Client secret.

Adding a Graph User to Group Fails

This issue is randomly observed.

To workaround this issue, delete the **DirXML-ApplicationSchema** attribute and restart the driver.

No Trusted Certificate Found Exception

A client certificate needs to be added to the driver's Java keystore. For more information, see "Securing Communication with Identity Manager Exchange Service" on page 76.

Driver Fails to Connect to Microsoft Graph API Due to Invalid Certificate Error

Most Azure services get their SSL/TLS certificates from a known set of intermediate certificate authorities (CAs) that Microsoft operates. Microsoft publishes details of these CAs in its Certificate Practice Statement (CPS). The following CA's have been recently introduced:

- Microsoft IT TLS CA 1
- Microsoft IT TLS CA 2
- Microsoft IT TLS CA 4
- Microsoft IT TLS CA 5

You must include the new CAs in the driver's truststore file. Otherwise, the driver reports an invalid certificate exception in the trace.

To workaround this issue, perform the steps described in "Securing Communication with Azure AD Graph" on page 99. You must repeat the procedure for all the certificates generated by all the four CAs. After the certificates are imported into the truststore file, the Azure driver works properly.

The name of a certificate is specified by the 'Issued by' field of the certificate.

Microsoft keeps replacing the CAs that it uses to validate Azure Graph API (https://graph.windows.net/); therefore, you must refresh the browser every time the API is launched. In case a new certificate is available, you must download it.

Driver Fails to Connect to Office 365 due to the MS Exchange License Issues

Explanation: When you are configuring the Azure AD driver with Office 365, fatal connectivity errors are encountered due to issues of MS Exchange Licenses. The error message is shown below:

```
<status level="fatal">Error Connecting to Office365. [ps.outlook.com]
Connecting to remote server ps.outlook.com failed with the following error
message : The WinRM client received an HTTP status code of 403 from the
remote WS-Management service. For more information, see the
about_Remote_Troubleshooting Help topic.</status> <init-params event-
id="write-state">
```

Possible cause: Certain issues with MS Exchange Licenses cause the driver to fail to connect and shuts down automatically.

Solution: You must configure the driver with an Azure account which has the `Global Administrator` role, and the appropriate subscriptions and licenses associated with it.

Driver Shuts Down if Role Assignment Fails in Azure Application

The resource authorization might fail and the driver shuts down, when you try to assign a role to a resource, for example, Global Administrator role in Azure application. This might happen due to permission issues associated with the role assigned.

An example of the error could be like as shown below:

```
<output> <status event-id="SLES15SP1#20200919045633#1#4:b286457d-cfc2-
45af-8487-7d4586b2c2cf" level="fatal" type="driver-
general">com.novell.nds.dirxml.driver.azure.exceptions.ChannelException:
{"odata.error":{"code":"Authorization_RequestDenied","message":{"lang":"en
","value":"Insufficient privileges to complete the
operation."},"requestId":"d14faaab-7aa2-44fb-b22a-
c7fef7575194","date":"2020-09-19T04:56:36"}}<operation-data><entitlement-
impl id="" name="Role" qualified-src-dn="O=data\OU=users\CN=ARastogi"
src="UA" src-dn="\SLES15SP1_TREE\data\users\ARastogi" src-entry-id="34673"
state="1">{"ID":"725e41c7-a785-4633-88b6-d9c6cfcfa8d8"}</entitlement-
impl></operation-data></status></output>
```

NOTE: The error messages differ case to case, and appear based on the specific issue encountered for role assignment.

You must ensure that the entitlement role used for assignment has all the correct permissions and rights available.

Channel Exception on Too Many Requests

Explanation: The user encounters the following error message after restarting the driver while performing transactions via MS GRAPH API.

```
com.novell.nds.dirxml.driver.azure.exceptions.ChannelException:
{"error":{"code":"UnknownError","message":"Too Many
Requests","innerError":{"date":"2023-04-03T07:39:41","request-
id":"030d3573-95b1-458d-b4ef-9b6aca307d5c","client-request-id":"030d3573-
95b1-458d-b4ef-9b6aca307d5c"}}}
```

Possible Cause: This occurs for both versions (5.1.5 and 5.1.6) of Azure AD driver due to the changes made on MS GRAPH API.

Solution: As a work around, add `signInActivity` attribute in "Unsupported User Attributes by MS Graph APIs" in driver configuration.

Error While Adding Users as Members to a Group in Azure

Issue: `"error">com.novell.nds.dirxml.driver.xds.XDSModifyElement cannot be cast to com.novell.nds.dirxml.driver.xds.XDSAddElement`

User may get the above error in the following scenarios:

- After changing merge authority to application, new groups coming from Azure attempt to change within the Identity Vault prior to the add.
- In Azure/Exchange Online, when adding members while creating a group.

(Bug 712005)

Workaround: Perform the following steps to resolve the issue:

1. Create a Group in Azure.
2. Sync it to Identity Vault.
3. Then, add members.

Certificate based authentication fails with "Refresh token and Client Secret must be provided to get the new Access Token exception" Error when driver is configured to use proxy server

Issue: When the driver is configured with proxy server, the Certificate Based Authentication for MS Graph fails with following error:

```
"Refresh token and Client Secret must be provided to get the new Access
Token exception”
```

Workaround: While configuring driver with proxy server, use Client Secret Authentication.

A Driver Properties

This section provides information about the Driver Configuration and Global Configuration Values properties for the Azure AD 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.

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

- "Driver Configuration" on page 147
- "Global Configuration Values" on page 152

Driver Configuration

In iManager:

1 Click to display the Identity Manager Administration page.

2 Open the driver set that contains the driver whose properties you want to edit:

2a In the **Administration** list, click **Identity Manager Overview**.

2b Click the **Driver Sets** tab.

2c If the driver set is not listed on the **Driver Sets** tab, use the **Search In** field to search for and display the driver set.

2d Click the driver set to open the Driver Set Overview page.

3 Locate the driver icon, then click the upper right corner of the driver icon to display the **Actions** menu.

4 Click **Edit properties** to display the driver's properties page.

By default, the Driver Configuration page is displayed.

In Designer:

1 Open a project in the Modeler.

2 Right-click the driver line, then click **Properties**.

3 Click **Driver Configuration**.

The Driver Configuration options are divided into the following sections:

- "Driver Module" on page 148
- "Driver Object Password" on page 148
- "Authentication" on page 148
- "Startup Option" on page 149
- "Driver Parameters" on page 149

Driver Module

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

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. This option is not applicable to this driver.
- **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. This is a qualified `userprincipalname` on Azure AD with login permissions. For example, `admin@domain.onmicrosoft.com`

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

The kmo entry is optional. It is only used when there is an SSL connection between the Remote Loader and the 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 iManager.

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

If the driver is **Disabled** and then changed to **Auto start** or **Manual**, you can select the **Do Not Automatically Synchronize the Driver** check box. This prevents the driver from synchronizing objects automatically when it loads. To synchronize objects manually, use the **Synchronize** button on the **Driver Overview** page.

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:

- "Driver Settings" on page 149
- "Driver Operation Mode" on page 150
- "Schema Extensions Configuration" on page 150
- "Subscriber Settings" on page 150
- "Publisher Settings" on page 151

Driver Settings

Client ID: Specifies the account name which the Azure AD driver will use to access the Azure AD applications. You need to set the level of permissions required by the driver.

Client Secret: Specifies the password for the client ID to access the Azure AD applications.

NOTE: For information on creating the Client ID and Client Secret for your application, see "Creating a Proxy Application on Azure AD" on page 82.

Remove Existing Passwords: Select this option to clear the existing password. You can enter a new password at this point.

Driver Operation Mode

Enable Hybrid Operation Mode: In hybrid mode, the driver supports only Roles and License entitlements. The users and groups are provisioned by the AD driver. By default, the parameter is set to **Yes**. If you want to run the driver in cloud-only mode, set the value to **No** and install the **Azure AD Cloud Only Entitlements** package.

Activate Azure Directory Roles: To activate the Azure AD roles, set this parameter to **Yes**. Azure AD driver will fetch only the roles that are already activated.

Schema Extensions Configuration

Show Schema Extensions Configuration: To show the schema extensions in the configuration wizard, select **Show**.

Existing Schema Extensions: To retain the previously-loaded configuration, select **Preserve**. However, when you select **Preserve** and add a new extension, the extension will be added. Select **Remove** to overwrite an existing configuration.

Add a schema extension: Add a schema extension and specify appropriate configuration details. You can add multiple schema extensions if required.

- **Name of extension:** Specify the name of the schema extension. If you create more than one schema extension with the same name, the first extension in the list will be used. The remaining extensions will be ignored.
- **Type of extension:** Indicates the data type for the configured schema extension. Ensure that the data type is a supported schema extension type in Azure AD.
- **Target objects of extension:** Lists the target objects for the schema extension. A schema extension can be extended to multiple target object classes. For example, if you have a schema extension called `Title`, it can be extended to a `User` and `Group` object class.

NOTE: You can configure a maximum of hundred extensions on Azure AD.

Subscriber Settings

Domain Name: Specify the Azure AD domain site context. For example, `<`*`domain name`*`>.onmicrosoft.com` or `<`*`domain name`*`>.com` format.

Truststore file: Specify the name and path of the truststore file containing the trusted certificates used when a remote server is configured to provide server authentication. This file will contain certificates for Azure Graph and Exchange Service. For example, `c:\security\truststore`.

Proxy Host and Port: When an HTTP proxy is used, specify the host address and the host port. For example, 192.10.1.3:18180. Otherwise, leave the field blank.

Set proxy authentication parameters: To set proxy authentication, select **Show**. and specify the user and password for proxy authentication.

Exchange and Powershell Service: When this service is enabled, the driver will synchronize Exchange users and groups using this service.

Exchange Service URL: Specify the URL of the Identity Manager Exchange Service.

For example, `https://<ip-addr>:<port>/ExchServer`.

Refresh Deleted User Cache: When you set this parameter to **Yes**, the local cache that contains the deleted users is refreshed with the objects present in the Office 365 deleted user container.

Office 365 Exchange Online: To initiate a connection with Exchange Online and synchronize Office 365 exchange users and groups, select **Yes**.

Queue Operations: To enable queuing of objects when synchronizing between Azure AD and Identity Manager Exchange Service, select **True**.

Page Size: Set a value for the number of results per page during Exchange Publisher poll.

Trace location: Specify the custom path where you want to save the Identity Manager Exchange Service logs. By default, the logs will be saved in this component's installation directory.

Trace Level: Set the trace level for the Identity Manager Exchange Service. The driver supports five trace levels: NOTIFY, INFO, ERROR, MORE INFO, and DEBUG. The default trace level is NOTIFY. The next trace level, that is, INFO provides basic trace messages. ERROR provides some additional information than the previous level. Detailed messages are logged if you select INFO. DEBUG logs information on debugging data along with detailed messages.

Trace File Size Limit: Specify the trace file size limit. The value is measured in MB. The minimum value is 10 MB.

Database Password: Specify the database password. This password is used to encrypt and connect to the Publisher cache. Ensure that the same password is used to reconnect to the cache at a later time.

Remove Existing Passwords: Select this option to clear the existing password. You can enter a new password at this point.

Group cache clear interval: Specify the value as required to clear the exchange group related cache data. For more information, see "Group Cache Clear Interval" on page 20.

Publisher Settings

Enable publisher: Allows you to enable or disable the Publisher connection for your Azure AD driver.

Publisher Polling Interval: Specify a time period after which Azure AD will be queried for new changes. The time is indicated in minutes.

Heart Beat Interval: Allows the driver to send a periodic status message on the Publisher channel when there has been no Publisher channel traffic for the given number of minutes. This indicates the time period at which the heart beat document is issued by the driver shim. The time is indicated in minutes.

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 Azure AD 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 iManager:

1. Click to display the Identity Manager Administration page.
2. Open the driver set that contains the driver whose properties you want to edit:

   2a In the **Administration** list, click **Identity Manager Overview**.

   2b If the driver set is not listed on the **Driver Sets** tab, use the **Search In** field to search for and display the driver set.

   2c Click the driver set to open the Driver Set Overview page.
3. Locate the Azure AD driver icon, click the upper right corner of the driver icon to display the **Actions** menu, then click **Edit Properties**.

   or

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

To access the driver's GCVs in Designer:

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

   or

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

The global configuration values are organized as follows:

- "Password Synchronization" on page 152
- "Driver Configuration" on page 153
- "Account Tracking" on page 154
- "Exchange Role Entitlement" on page 154
- "Entitlements" on page 155
- "Managed System Information" on page 158

Password Synchronization

These GCVs enable password synchronization between the Identity Vault and the Azure AD 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 iManager, you should edit the Password Management Options on the **Server Variables** tab rather than under the GCVs. The Server Variables page has a better view of the relationship between the different GCVs.

Connected System or Driver Name: Specify the name of the Azure AD 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.

Driver Configuration

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

Synchronization Settings

Use the following GCVs to control how the driver is configured:

Office 365 settings

- **Domain Name:** Specify the Office 365 site context using the `admincentral.onmicrosoft.com` format.
- **Identities to be synchronized:** Specify whether the driver should synchronize identities from AD or configure the Identity Vault to act as the identity provider.

  If you choose to configure the Identity Vault as an identity provider, association to any other directory is not required.

  When you choose to synchronize identity from AD, you can synchronize only users that have an association with AD. If you are using the driver in hybrid mode, select only **AD** option. This enables the driver to synchronize the identities from the Identity Vault to AD from where the identities will be synchronized to Azure AD cloud through Azure AD Connect.

- **Usage Location:** Specify the two letter country code of the user availing Office 365 services.
- **Enable Hybrid Operation Mode:** If **Yes**, the driver will provision only Roles and License entitlements while the users and groups are provisioned by the AD driver. To run the driver in normal mode, set the option to **No**.

Account Tracking

Account tracking is part of 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.

Realm: Specify the name of the realm, security domain, or namespace in which the account name is unique. You must set the Realm to the Office 365 Domain Name.

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

Exchange Role Entitlement

This entitlement is supported if you have upgraded to the 5.0.1 version of the driver. You need to import the **Azure AD Exchange Role Entitlement** package to use this entitlement.

NOTE: Before you use this entitlement, ensure that exchange service is running.

Use Exchange Roles Entitlement: Select **True** to enable the driver to manage exchange roles based on the driver's defined entitlements.

Advanced Settings: To enable the advanced settings such as data collection, role mapping, and resource mapping, select **Show**.

- **Allow data collection from exchange roles:** Select **Yes** if you want to allow data collection by Data Collection Service for exchange roles.
- **Allow role mapping of exchange roles:** Select **Yes** if you want to allow mapping of exchange roles in Identity Applications.
- **Allow resource mapping of exchange roles:** Select **Yes** if you want to allow mapping of exchange roles in Identity Reporting.
- **Exchange Role extensions:** The content of this field is added below the entitlement elements in the EntitlementConfiguration resource object.

Entitlements

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

- "Entitlements" on page 155
- "Data Collection" on page 156
- "Role Mapping" on page 157
- "Resource Mapping" on page 157
- "Entitlement Extensions" on page 157

Entitlements

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 Azure AD can be controlled by using entitlements.

NOTE: User Account Entitlement is supported in cloud-only mode. It is not supported in hybrid mode.

- **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 Azure AD database when a User Account entitlement is revoked from an Identity Vault user. The options are **Disable Account** or **Delete Account**.

Use Group Membership Entitlement: Select **True** to enable the driver to manage Azure AD group membership based on the driver's Group entitlement.

Select **False** to disable management of group membership based on entitlement.

Use License Entitlement: Select True to enable the driver to manage licenses based on the driver's defined entitlements. To assign multiple Azure AD licenses, you must create multiple resources on user application. This is required because an Azure AD license entitlement can have only single value.

Use Roles Entitlement: Select True to enable the driver to manage roles based on the driver's defined entitlements. Select False to disable management of role assignments for users based on the entitlements.

Teams Entitlement: Select True to enable the driver to manage roles based on the driver's defined entitlements. Select False to disable management of Teams assignments for users based on the entitlements.

- Add User as Owner to Team - Select "Yes" to add the User as "Owner" to Team.

Channel Entitlement: Select True to enable the driver to manage roles based on the driver's defined entitlements. Select False to disable management of channel assignments for users based on the entitlements.

- Add User as Owner to Channel - Select "Yes" to add the User as "Owner" to Channel.
- Team and Channel Name Separator - Specify the character to separate Team and Channel name. For example: Team::Channel

SKU Entitlement: Ensure the value of this parameter is set to true to enable the driver to manage SKU Subscription assignments based on the entitlement. By default, the value is set to True.

Data Collection

Data collection enables Identity Reporting to gather information to generate reports.

Enable data collection: Select Yes to enable data collection for the driver through 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 Data Collection Service for user accounts.

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

Allow data collection from licenses: Select Yes to allow data collection by Data Collection Service for licenses.

Allow data collection from Roles: Select Yes to allow data collection by Data Collection Service for roles.

Allow data collection from SKU: Select Yes to allow data collection by Data Collection Service for SKU.

Allow data collection from Teams: Select Yes to allow data collection by Data Collection Service for teams.

Allow data collection from Channels: Select Yes to allow data collection by Data Collection Service for channels.

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 the 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 licenses: Select Yes if you want allow mapping of licenses in Identity Applications.

Allow mapping of Roles: Select Yes if you want allow mapping of roles in Identity Applications.

Allow mapping of SKU: Select Yes if you want allow mapping of SKU in Identity Applications.

Allow mapping of Teams: Select Yes if you want allow mapping of teams in Identity Applications.

Allow mapping of Channels: Select Yes if you want allow mapping of channels in Identity Applications.

Resource Mapping

The Identity Applications allow you to map resources to users.

Enable 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 licenses: Select Yes if you want to allow mapping of licenses in Identity Applications.

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

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

Allow mapping of Teams: Select Yes if you want to allow mapping of teams in Identity Applications.

Allow mapping of Channels: Select Yes if you want to allow mapping of channels 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 elements in the EntitlementConfiguration resource object.

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

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

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

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

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

Managed System Information

These settings help Identity Reporting function to generate reports. There are different sections in the **Managed System Information** tab.

- "General Information" on page 158
- "System Ownership" on page 158
- "System Classification" on page 158
- "Connection and Miscellaneous Information" on page 159

General Information

Name: Specify a descriptive name for the managed system.

Description: Specify a brief description of the managed system.

Location: Specify the physical location of the managed system.

Vendor: Select Microsoft as the vendor of the managed system.

Version: Specify the version of the managed system.

System Ownership

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

System Classification

Classification: Select the classification of the connected application. 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 connected application.

Environment: Select the type of environment the connected application provides. The options are:

- Development
- Test
- Staging
- Production
- Other

  If you select **Other**, you must specify a custom classification for the connected application.

Connection and Miscellaneous Information

Connection and miscellaneous information: This set of options is always set to **hide**, so that you do not make changes to these options. These options are system options that are necessary for reporting to work.

B Known Issues and Limitations

NetIQ Corporation strives to ensure that our products provide quality solutions for your enterprise software needs. The following issues are currently being researched. If you need further assistance with any issue, please contact Technical Support.

- If you are adding the Manager attribute while creating the user, it doesn't synchronize in IDV. Hence the work around would be to update the Manager after the User Creation is done..
- When you modify the Office 365 user name or password and restart the Azure AD driver, there is an increase in the Exchange service memory.
- If you modify the description for a group on Identity Vault, the changes are reflected on Office 365 and Exchange portals, but not on Azure portal.
- If you delete security, distribution, or Office 365 groups from Office 365, the driver cannot restore these groups. Microsoft does not support restoring these groups. For more information, see the Microsoft documentation.
- User assignments to some exchange admin roles fail and displays the following error message:

  ```
  com.novell.nds.dirxml.driver.azure.exceptions.ChannelException: Add-
  RoleGroupMember
  ```

  This issue is observed because some exchange admin roles do not support direct user assignments.

  NOTE: This issue is specific to 5.0.1 version of the driver.

- **Add User as Owner to Team**, **Add User as Owner to Channel** and **Team and Name Channel** Separator are the GCV Values. These values can be ignored under driver configuration.
- Set-User cmdlet fails to update Phone and Mobile Phone for users with Recipient Type Details "UserMailbox". For updating these attributes use cmdlet Set-Msoluser.