Connect to Microsoft Graph PowerShell

22 March 2023 11:17

Microsoft is deprecating the Azure AD PowerShell module and MS Online module in 2023. So, admins need to migrate to either Microsoft Graph PowerShell SDK or Microsoft Graph API.

- Download and install the Microsoft Graph PowerShell module
- Connect to Microsoft Graph PowerShell
- PowerShell script to automatically install and connect to MS Graph (Recommended)
- Microsoft Graph PowerShell commands
- Migrate scripts from Azure AD PowerShell to Microsoft Graph
- Update and Uninstall Microsoft Graph SDK

Install the Microsoft Graph PowerShell SDK

You can install Microsoft.Graph module by running the below cmdlet in Windows PowerShell.

Install-Module Microsoft.Graph -Scope CurrentUser

It will install the module for the currently logged-in user profile. If you want to install it for all the users, you must run the following cmdlet by opening Windows PowerShell in admin mode.

Install-Module Microsoft.Graph -Scope AllUsers

Installing Microsoft.Graph module installs 38 sub-modules. To get a list of available modules, you can use the Find-Module cmdlet.

Find-Module Microsoft.Graph.*

After installation, you can check the version of the Microsoft Graph PowerShell module by using the following cmdlet.

Get-InstalledModule Microsoft.Graph

How to Connect to Microsoft Graph PowerShell

Connect to Microsoft Graph – Things to Know:

While connecting to Microsoft Graph PowerShell, you need to determine the 2 important things.

1. API Version
2. Required Scope

API version: By default, the module uses a stable version of the API. If you want to explore new features, you can use the beta version using the ‘Select-MgProfile’ cmdlet.

Select-MgProfile –Name “beta”

Scope: Microsoft Graph provides access to all APIs in Microsoft 365. Each API is protected by one or more permission scopes. The user connecting to Microsoft Graph PowerShell must consent to one of the required scopes. For example, to view all the Microsoft 365 users, you need the ‘User.Read.All’ permission.

Finding the required scope is challenging. You can refer to the Microsoft doc to determine the required scope.

Connect to Microsoft Graph:

The ‘Connect-MgGraph’ cmdlet allows you to connect to Microsoft Graph PowerShell. You will need to sign in with an admin account to consent to the required scopes.

Connect-MgGraph –Scopes “User.Read.All”

If you want to connect to Microsoft Graph with multiple scopes, you can provide them as comma- separated values.

Connect-MgGraph –Scopes “User.Read.All”,”Group.ReadWrite.All”

Some scopes require you to grant permission on behalf of your organization. Once you connect successfully, you will receive a “Welcome to Microsoft Graph!” message.

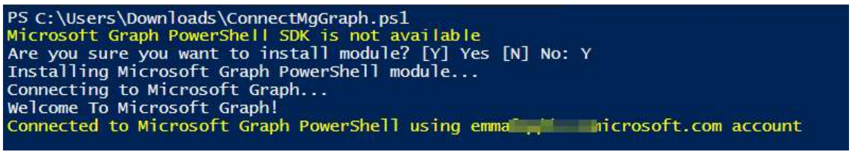
Automatically Install and Connect to Microsoft Graph:

This PowerShell script will automatically install Microsoft.Graph module (if not installed already) upon your confirmation. After installation, it will prompt you to connect to Microsoft Graph.

Place Download Script Here

You can execute the script using the following format.

.\ConnectMgGraph.ps1



```
PS C:\Users\Downloads\ConnectMgGraph.ps1
Microsoft Graph PowerShell SDK is not available
Are you sure you want to install module? [Y] Yes [N] No: Y
Installing Microsoft Graph PowerShell module...
Connecting to Microsoft Graph...
Welcome To Microsoft Graph!
Connected to Microsoft Graph PowerShell using emma[redacted]icrosoft.com account
```

Microsoft graph keeps encrypted token and refreshes it automatically. So, the session remains open for a long time. If you want to connect to a different tenant, you can run the script with the –CreateSession parameter.

.\ConnectMgGraph.ps1 -CreateSession

Connect MS Graph using Certificate:

If you don’t want to enter credentials, consider using certificates. Admins can easily create self-signed certificates for internal and testing purposes, eliminating the need for expensive third-party CA certificates. To connect Microsoft Graph using certificate, run the following cmdlet.

Connect-MgGraph -ClientID <Client ID> -TenantId <Tenant ID> -CertificateThumbprint <Thumbprint> #You can use -CertificateName <Certificate subject> instead of –CertificateThumbprint

Note: To use certificate-based authentication, you must register app in Azure AD or you can automate Azure app registration using PowerShell script.

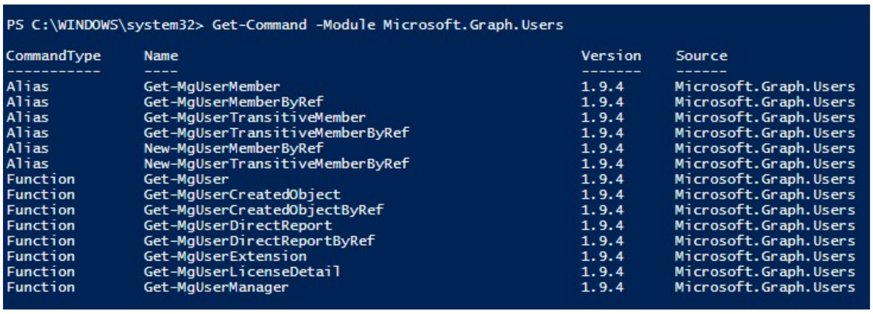
View Microsoft Graph PowerShell Commands:

When migrating to a new PowerShell module, it is challenging to find the right cmdlets. For example, to view all the Azure AD users, you need to use the ‘Get-MgUser’ cmdlet. To easily find the required cmdlets, you can follow the below tips.

To view Microsoft Graph PowerShell cmdlets for a specific module, run the following cmdlet.

Get-Command -Module Microsoft.Graph.Users

It will list all the cmdlets related to Azure AD users.



```
PS C:\WINDOWS\system32> Get-Command -Module Microsoft.Graph.Users

CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Alias           Get-MgUserMember                                   1.9.4      Microsoft.Graph.Users
Alias           Get-MgUserMemberByRef                              1.9.4      Microsoft.Graph.Users
Alias           Get-MgUserTransitiveMember                         1.9.4      Microsoft.Graph.Users
Alias           Get-MgUserTransitiveMemberByRef                    1.9.4      Microsoft.Graph.Users
Alias           New-MgUserMemberByRef                              1.9.4      Microsoft.Graph.Users
Alias           New-MgUserTransitiveMemberByRef                    1.9.4      Microsoft.Graph.Users
Function        Get-MgUser                                         1.9.4      Microsoft.Graph.Users
Function        Get-MgUserCreatedObject                            1.9.4      Microsoft.Graph.Users
Function        Get-MgUserCreatedObjectByRef                       1.9.4      Microsoft.Graph.Users
Function        Get-MgUserDirectReport                             1.9.4      Microsoft.Graph.Users
Function        Get-MgUserDirectReportByRef                        1.9.4      Microsoft.Graph.Users
Function        Get-MgUserExtension                                1.9.4      Microsoft.Graph.Users
Function        Get-MgUserLicenseDetail                            1.9.4      Microsoft.Graph.Users
Function        Get-MgUserManager                                  1.9.4      Microsoft.Graph.Users
```

As earlier said, you can use Find-Module Microsoft.Graph.* to view the list of modules.

To view all the Microsoft Graph cmdlets, execute the following cmdlet.

Get-Command -Module Microsoft.Graph.*

You can also refer our dedicated blog on the top 10 Microsoft Graph PowerShell cmdlets to generate Office 365 reports.

Migrate from Azure AD PowerShell to the Microsoft Graph PowerShell SDK

As Microsoft is retiring the Azure AD and MSOL module at the end of 2022, admins need to upgrade their current Azure AD scripts. There is no native tool available to convert Azure AD scripts to Microsoft Graph PowerShell. Admins need to upgrade their scripts manually.

Admins must find the equivalent Microsoft Graph cmdlets for the Azure AD PowerShell cmdlets to convert their scripts. For example, the equivalent of the Get-AzureADUser cmdlet is Get-MgUser.

Disconnect Microsoft Graph PowerShell

The Microsoft Graph session lasts until you disconnect it. It keeps an encrypted token cache and refreshes it automatically. So, you won’t encounter the ‘Session time out’ error, unlike other PowerShell modules.

To connect to another tenant, you must disconnect the Microsoft Graph session using the following cmdlet.

Disconnect-MgGraph

To avoid using an earlier token cache, you can connect to Microsoft Graph using ‘TenantId’ as below.

Connect–MgGraph –TenantId <TenantId>

Update and Uninstall Microsoft Graph SDK PowerShell:

To update the SDK, you can use the following cmdlet.

Update-Module Microsoft.Graph

If you want to uninstall the Microsoft Graph PowerShell module, you must uninstall the main module first. And then all its dependency modules.

Uninstall-Module Microsoft.Graph
Get-InstalledModule Microsoft.Graph.* | %{ if($_.Name -ne "Microsoft.Graph.Authentication"){ Uninstall-Module $_.Name } }
Uninstall-Module Microsoft.Graph.Authentication