

# Microsoft Graph

- [Disable Microsoft 365 / Entra ID Federation with PowerShell](#)
- [Connect to Microsoft 365 with Microsoft Graph PowerShell](#)

# Disable Microsoft 365 / Entra ID Federation with PowerShell

1. [Install the Microsoft Graph PowerShell.](#)
2. Set the Execution Policy to Remote Signed:

```
Set-ExecutionPolicy RemoteSigned
```

3. Connect to your Microsoft 365 / Entra ID tenant:

```
Connect-MGGraph -Scopes "Domain.ReadWrite.All", "Directory.AccessAsUser.All",  
"Organization.ReadWrite.All", "Directory.ReadWrite.All"
```

4. Enter your Office 365 Global Administrator Credentials.
5. Consent and Accept the requested scopes.
6. Verify the domain is federated:

```
Get-MgDomain -DomainId "<YourO365Domain.com>"
```

7. Change Federation Authentication from **federated** to **managed**:

```
Update-MgDomain -DomainId "<YourO365Domain.com>" -AuthenticationType Managed
```

8. To check Federation status:

```
Get-MgDomain -DomainId "<YourO365Domain.com>"
```

9. Disconnect Microsoft Graph:

```
Disconnect-MGGraph
```

# Connect to Microsoft 365 with Microsoft Graph PowerShell

<https://learn.microsoft.com/en-us/microsoft-365/enterprise/connect-to-microsoft-365-powershell?view=o365-worldwide#connect-with-the-microsoft-azure-active-directory-module-for-windows-powershell>

## Prerequisites

PowerShell 7 and later is the recommended PowerShell version for use with the Microsoft Graph PowerShell SDK on all platforms. There are no other prerequisites to use the SDK with PowerShell 7 or later.

The following prerequisites are required to use the Microsoft Graph PowerShell SDK with Windows PowerShell.

- Upgrade to PowerShell 5.1 or later
- Install .NET Framework 4.7.2 or later
- Update PowerShellGet to the latest version using Install-Module PowerShellGet

The PowerShell script execution policy must be set to remote signed or less restrictive. Use Get-ExecutionPolicy to determine the current execution policy. For more information, see [about\\_Execution\\_Policies](#). To set the execution policy, run:

PowerShell 

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

## Operating system

You must use a 64-bit version of Windows. You can use the following versions of Windows:

- Windows 11, Windows 10, Windows 8.1, Windows 8, or Windows 7 Service Pack 1 (SP1)
- Windows Server 2019, Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, or Windows Server 2008 R2 SP1

## Note

For Windows 8.1, Windows 8, Windows 7 Service Pack 1 (SP1), Windows Server 2012 R2, Windows Server 2012, and Windows Server 2008 R2 SP1, download and install the [Windows Management Framework 5.1](#).

To use Microsoft Graph PowerShell, you must use at least PowerShell version **5.1**.

## Note

These procedures are intended for users who are members of a Microsoft 365 admin role. For more information, see [About admin roles](#).

# Connect with Microsoft Graph PowerShell

In this section, you'll learn how to connect to your Microsoft 365 organization using the Microsoft Graph PowerShell SDK. You can visit [Install the Microsoft Graph PowerShell SDK](#) for more guidance.

## Step 1: Install the required software

The Microsoft Graph PowerShell SDK is published in the [PowerShell Gallery](#).

These steps are required only one time on your computer. However, you'll likely need to update the software periodically.

## Install the Microsoft Graph PowerShell SDK and beta module

The Microsoft Graph PowerShell SDK comes in two modules, Microsoft.Graph and Microsoft.Graph.Beta, that you'll install separately. These modules call the Microsoft Graph v1.0 and Microsoft Graph beta endpoints, respectively. You can install the two modules on the same PowerShell version.

1. Open a Windows PowerShell Command Prompt window. Depending on the permissions of your logged-in account, you may need to open the PowerShell window in Administrator mode.

2. To install the v1 module of the SDK in PowerShell Core or Windows PowerShell, run the following command:

PowerShell Copy

```
Install-Module Microsoft.Graph -Scope CurrentUser
```

3. Run this command to install the beta module:

PowerShell Copy

```
Install-Module Microsoft.Graph.Beta
```

After the installation is completed, you can verify the installed version with the following command:

PowerShell Copy

```
Get-InstalledModule Microsoft.Graph
```

## Step 2: Connect to your Microsoft 365 subscription

The PowerShell SDK supports two types of authentication: delegated access, and app-only access. In this guide, you'll use delegated access to sign in as a user, grant consent to the SDK to act on your behalf, and call the Microsoft Graph.

For details on using app-only access for unattended scenarios, see [Use app-only authentication with the Microsoft Graph PowerShell SDK](#).

### Determine required permission scopes

Each API in the Microsoft Graph is protected by one or more permission scopes. The user logging in must consent to one of the required scopes for the APIs you plan to use. In this example, we'll use the following APIs.

- List users to find the user ID of the logged-in user.
- List joinedTeams to get the Teams the user is a member of.
- List channels to get the channels in a Team.
- Send message to send a message to a Team's channel.

The **User.Read.All** permission scope enables the first two calls, and the **Group.ReadWrite.All** scope enables the rest. These permissions require an admin account.

For more information about how to determine what permission scopes you'll need, see [Using Find-MgGraphCommand](#).

To connect to your Microsoft 365 Organization, run the following command:

PowerShell 

```
Connect-MgGraph -Scopes "User.Read.All","Group.ReadWrite.All"
```

The command prompts you to go to a web page to sign in with your credentials. Once you've done that, the command indicates success with a **Welcome To Microsoft Graph!** message. You only need to sign in once per session.