88993

Get Microsoft Teams Member Lists

POWERSHELL ****

Author:

[**Michael D. Newby**](http://www.recursivecreativity.com/)

[www.recursivecreativity.com](http://www.recursivecreativity.com/)

[www.excelusersgroup.org](http://www.excelusersgroup.org/)

(not active)

Last Updated:

**October 23, 2022**

**Table of Contents**

[**Windows PowerShell ISE** 4](#_Toc117459973)

[**PowerShell Scripts** 6](#_Toc117459974)

[**Installation of MicrosoftTeams module for PowerShell** 7](#_Toc117459975)

[**Create Member List(s)** 9](#_Toc117459976)

[**Getting Started** 10](#_Toc117459977)

[**Installing the PowerShell MicrosoftTeams Module** 11](#_Toc117459978)

[**1.** **Open PowerShell ISE** 11](#_Toc117459979)

[**2.** **Install MicrosoftTeams module for PowerShell** 12](#_Toc117459980)

[**Create Member List for Team** 13](#_Toc117459981)

[**Create New Script for Different Team Member List** 16](#_Toc117459982)

[**Update MicrosoftTeams PowerShell Module** 17](#_Toc117459983)

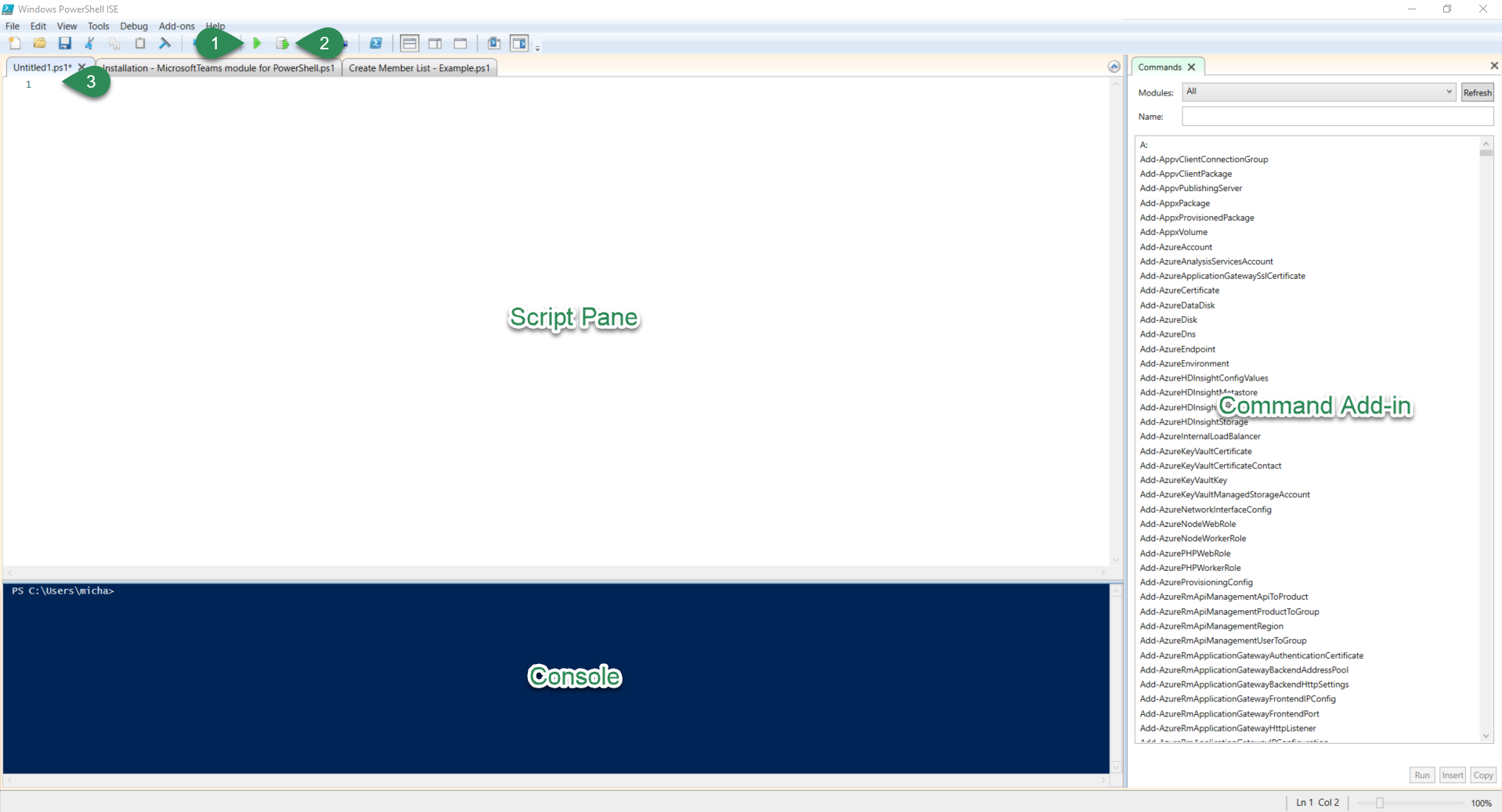
[**Miscellaneous PowerShell Information** 18](#_Toc117459984)

[**How to create and run a PowerShell script file on Windows 10 | Windows Central** 18](#_Toc117459985)

[**Useful PowerShell Commands**: 18](#_Toc117459986)

[**History Log** 18](#_Toc117459987)

# [**Windows PowerShell ISE**](#TOC)



Clicking on View in the menu will provide options to make each of the above panes viewable or displayed.

In additional to the displayed panes that are labelled,

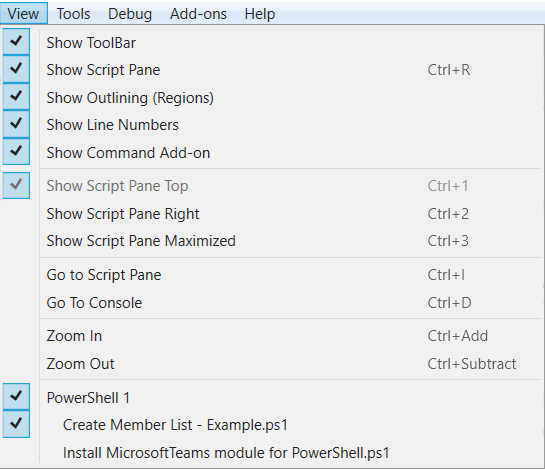
1 = Run button This runs \ executes all code; code for each line in numbered order.

2 = Run Selected Only selected lines of code are executed \ run.

3 = Line item Each new line of code is numbered; easier to reference when discussing.

The main difference between the ISE and the command line environments is that in the command line window \ console, only supports one line item and code is executed one at a time. In ISE, all lines of code are entered first and then can be executed in order (all; 1) or highlighted lines can be executed separately (2).

My preference is to make everything visible:



# **[PowerShell Scripts](#TOC)**

There are two scripts for creating member lists for Teams.

1. Install the MicrosoftTeams module for PowerShell:

Both a text (.txt) and PowerShell script (.ps1) are provided. This script checks if the Documents folder exists. If not, the folder is created. In either case, the script proceeds to install the MicrosoftTeams module for PowerShell.

The entire script in file above can be found [**below**](#_Installation_of_MicrosoftTeams). Hopefully, the scripts are sufficiently commented (#).

1. Create a member list:

Both a text (.txt) and PowerShell script (.ps1) are provided. There are numerous variables at the beginning of the script that can be changed as needed. If default path and file extension are not changed, only the first two variables need to be changed for each new script (or Team member list) the code is to be used for.

The entire script in file above can be found [below](#_Create_Member_List(s)_1). Hopefully, the scripts are sufficiently commented (#).

If needed, the text in embedded script files above or text below can be copied to Windows PowerShell ISE Script Pane to be run \ executed. If that is choice, remember to save the scripts after the first one has been run. The first script creates the path I recommend where all scripts should be saved. The first script is also required before running the second script.

**Please note**

To comment line or text, # is used. Thus, in embedded files above and code below, # is used to designate comments and provide documentation.

## **[Installation of MicrosoftTeams module for PowerShell](#TOC)**

######################################################################################

# If folder does not exist, this creates a Documents folder for logged in user.

#

# This is mostly applicable in corporate environments where the Documents folder

# is mapped to network drive in data center.

######################################################################################

# Change to system drive (local computer)

C:

# Set variable for required path and folder for current user

$strPath = $env:UserProfile + "\Documents"

# Create folder in given path, if it does not exist

if (-not (Test-Path -LiteralPath $strPath -PathType Any))

{

New-Item -Path $strPath -ItemType Directory

}

else

{

cls

Write-Host "Path,"$strPath“, already exists”

}

# Last line above provides notification in Console pane if folder exists.

# Installation of MicrosoftTeams module for PowerShell continues regardless.

######################################################################################

# Please note: several windows will appear requiring you to answer questions.

#

# For example:

# 1. PowerShell requires NuGet provider version...

#

# Click "Yes" if above prompt appears

#

# 2. You are installing the modules from an untrusted repository. If...

#

# Click "Yes to All" when above prompt appears

#

# If you do not click "Yes to All" above, I do not know all the prompts you see.

#

# UPDATE: ECHO Y has been added to force answering "Yes to All"

######################################################################################

# Final note:

#

# This installs to the Documents folder, created above, on local computer,

# which is my preference for a number of reasons.

#

# 1. The module should not be installed to network drive folder.

# 2. This is folder I save all my PowerShell scripts to and want them available

# offline and in backup path.

######################################################################################

#Check if MicrosoftTeams module exists

# if exists = update, else install

if (Get-Module -ListAvailable -Name MicrosoftTeams)

######################################################################################

# Update the MicrosoftTeams module for PowerShell

######################################################################################

{

# Change to system drive

C:

# Make C:\users\{logged in user}\Documents; see variable above.

# If variable was changed, it will be reflected here.

cd $strPath

# Update the MicrosoftTeams module for PowerShell for logged in user

Update-Module MicrosoftTeams | ECHO Y

}

else

######################################################################################

# Install the MicrosoftTeams module for PowerShell

######################################################################################

{

# Change to system drive

C:

# Make C:\users\{logged in user}\Documents; see variable above.

# If variable was changed, it will be reflected here.

cd $strPath

# Install the MicrosoftTeams module for PowerShell for logged in user

Install-Module MicrosoftTeams -Scope CurrentUser

}

## [**Create Member List(s)**](#TOC)

# These first FOUR lines of script (red font) are editable variables that can be changed to create other Scripts.

#

# Reminder to save with new name for each Team member list

#

# If this script is used for member list in csv format, you only need change variables 1 (line 3) and 2 (line 4)

$GroupID = "GROUP ID goes here" # GroupId of Microsoft Team

$File = "name of Team; file name" # Name of Team or short name for member list

#

# This assumes default path should be C:\users\{WindowsID}\Documents

$Path = ".\Documents\Members-" # Initial path and start of file name to identify "member" list

# If this variable is changed, the last line needs to be changed accordingly

$FileType = "csv" # This is file extension designated in last line: Export-Csv

#

# This changes path to what is normally the system drive (Where Windows is installed)

C:

#

# This changes the default path to where MicrosoftTeams module for PowerShell was installed

# and most likely place to save script files such as these

cd\

cd users

cd $env:UserName

#

# open MicrosoftTeams module for PowerShell

Get-Module MicrosoftTeams -ListAvailable

Connect-MicrosoftTeams

#

# set date stamp

$DateStamp = Get-Date -Format " - yyyy-MM-dd"

#

# set path and date stamped file name

$PathFileName = $Path + $File + $DateStamp + '.' + $FileType

#

# create member list for specified file type (csv) in given path for Team identified by -GroupId

# $FileType variable above must match Export-??? file type below

Get-TeamUser -GroupId $GroupID | Export-Csv -Path $PathFileName -NoTypeInformation

# [**Getting Started**](#TOC)

As mentioned above, the text from the embedded script files or text for each script above can be copied to the Windows PowerShell ISE. Alternatively, the above [PowerShell Script](#_PowerShell_Scripts) files should be copied to your local computer and renamed. Copy them anywhere to get started.

Once copied to your computer or other preferred location, **rename them by changing the file extension from .txt to .ps1.**

If you DO NOT make changes to the scripts, the above [PowerShell Scripts](#_PowerShell_Scripts) can be run as is.

Eventually, all scripts, including the above scripts, should reside in either of the following paths:

1. **Local computer**:

C:\users\{Logged on User Name}\Documents\**WindowsPowerShell**\Scripts

1. **If your organization maps Documents folder to network drive (data center)**:

?\Documents\**WindowsPowerShell**\Scripts

Where ? is drive letter mapped to Documents folder on network drive.

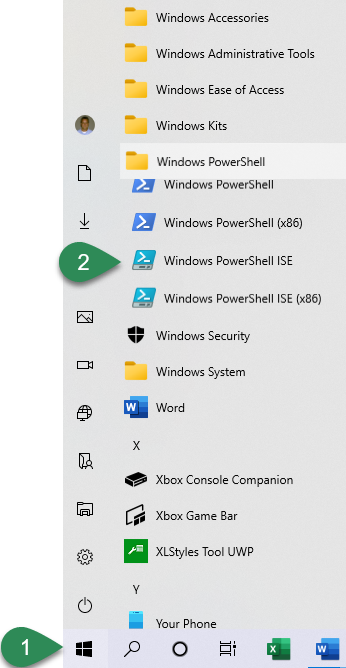
In either of the above or other path, make sure the path is backed up.

## [**Installing the PowerShell MicrosoftTeams Module**](#TOC)

These are instructions for using Windows PowerShell Integrated Scripting Environment (ISE).

### **Open PowerShell ISE**

* 1. Click on **Start** (1)
  2. Click on **Windows PowerShell ISE** (2)



Once opened you can pin this icon to your Taskbar by right-clicking on and selecting **Pin to Taskbar**.

### **Install MicrosoftTeams module for PowerShell**

* 1. Click on Browse (folder) icon to navigate to Script files. Double-click on the **Install MicrosoftTeams module for PowerShell.ps1** script to open it.
  2. If all defaults are acceptable, click on Run button or press F5.

Graphical user interface, text, application

Description automatically generated

When finished, the Console pane will display verbiage indicating the script ran successfully or display some type of error message.

Before continuing, you may want to

* Click on **Clear Console Pane** button (window squeegee; see green arrow above)
* Save the current script
* Close the current script

## **[Create Member List for Team](#TOC)**

1. Click on Browse (folder) icon (**a** in image below) to navigate to Script files. Double-click on the following script to open it: **Create Member List - Example.ps1**
2. Edit the first two variables by replacing the verbiage between the double quotes

$GroupID = "GroupID goes here"

$ListName = "name of Team; file name"

#### [**Get Team GroupID**](#_Creating_Your_Own)

1. Open **Teams**
2. Scroll to team
3. Click on team
4. To right of Team name are three dots (ellipsis), **click on the ellipsis** to display menu
5. Click on **Get link to team**
6. Click on **Copy** button
7. Somewhere in PowerShell Scrip Pane, press Ctrl+V to paste the link to team
8. Highlight and copy (Ctrl+C) everything between the yellow highlighted example verbiage below

groupId=**abc12d34-5678-9e0f-12gh-i3456789j0k1**&tenantId=

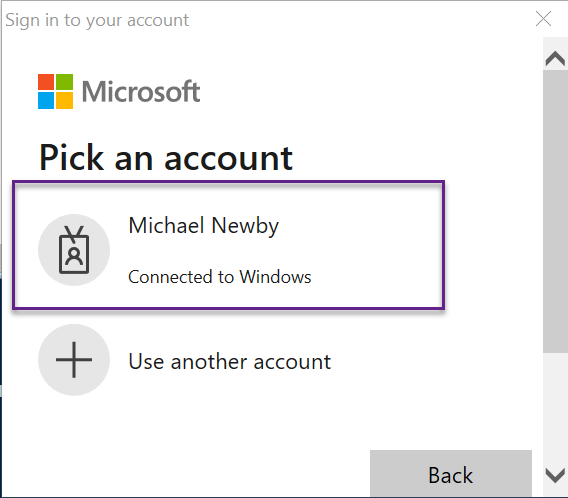
1. Remove information copied and pasted from **Get link to team**
2. If all other information is acceptable, click on Run button \ icon (**b** in image below) or press F5.

Text

Description automatically generated

When the above script is run, you will be prompted to validate your login credentials…

**Click on your name or Account that you are logged in with…**



When finished, the Console pane will display verbiage indicating the script ran successfully or display some type of error message.

Before continuing, you may want to…

1. Click on **Clear Console Pane** button (window squeegee)

Save the script using Team name…

1. Click on **File**
2. Click on **Save As…**
3. Change **File Name:** by replacing **Example** with **Team name**

# [**Create New Script for Different Team Member List**](#TOC)

1. Either rename the original script, **Create Member List - Example.ps1**, or open and using **File**, **Save As**, save the script after the Team you want member list for.

In either scenario above, replace **Example** with the **Team name**.

1. Repeat [**Create Member List for Team**](#_Create_Member_List) steps 1 through 3 above.

# [**Update MicrosoftTeams PowerShell Module**](#TOC)

Below are a text (.txt) and PowerShell script (.ps1) for updating the MicrosoftTeams PowerShell module.

The following is the script:

# Change to system drive (local computer)

C:

# Set variable for required path and folder for current user

$strPath = $env:UserProfile + "\Documents"

# Check if MicrosoftTeams module is installed

# if exists = update, else install

if (Get-Module -ListAvailable -Name MicrosoftTeams)

######################################################################################

# Update the MicrosoftTeams module for PowerShell

######################################################################################

{

# Change to system drive

C:

# Make C:\users\{logged in user}\Documents; see variable above.

# If variable was changed, it will be reflected here.

cd $strPath

# Update the MicrosoftTeams module for PowerShell for logged in user

Update-Module MicrosoftTeams | ECHO Y

}

# **[Miscellaneous PowerShell Information](#TOC)**

# [**How to create and run a PowerShell script file on Windows 10 | Windows Central**](https://www.windowscentral.com/how-create-and-run-your-first-powershell-script-file-windows-10)

## [**Useful PowerShell Commands**:](#TOC)

* Get-Help {PowerShell command}

example: Get-Help Export-Csv

* Uninstall-Module MicrosoftTeams
* Update-Module MicrosoftTeams

These commands must be run in folder MicrosoftTeams module was installed in.

* CLS (clear screen)

## [**History Log**](#TOC)

In above [**Connect to MicrosoftTeams Module**](#Connect) steps, rather than copying pasting code into PowerShell, the up and down arrows can be used to scroll through previously used commands.

The following can be used with Administrative privileges:

* Clear-History
* Clear-EventLog "Windows PowerShell"