# **Export All Office 365 members and their respective groups**

As an admin, monitoring groups and their membership is one of the routine tasks. Most admins prefer PowerShell over Microsoft admin center for better customization.

PowerShell has several cmdlets like Get-MSolGroup, Get-AzureADGroup, Get-UnifiedGroup, Get-DistributionGroup, Get-DistributionGroupMember to get group information but in this tutorial, we will get all those group information in a single report. We the **All-in-One** PowerShell script, you will be able to export groups and their membership details to CSV file.

Graphical user interface, text, application, table

Description automatically generated

Note: We are running a long script, so I recommend the use of PowerShell ISE

**Step1:**

Open Windows PowerShell

**Step1:**

Copy the script below and paste in the PowerShell console

*<#*

*=============================================================================================*

*Name: Microsoft 365 Group Report*

*Description: This script exports Microsoft 365 groups and their membership to CSV*

*Script by: O365Reports Team*

*============================================================================================*

*#>*

*Param*

*(*

*[Parameter(Mandatory = $false)]*

*[string]$GroupIDsFile,*

*[switch]$DistributionList,*

*[switch]$Security,*

*[switch]$MailEnabledSecurity,*

*[Switch]$IsEmpty,*

*[Int]$MinGroupMembersCount,*

*[string]$UserName,*

*[string]$Password*

*)*

*Function Get\_members*

*{*

*$DisplayName=$\_.DisplayName*

*Write-Progress -Activity "`n Processed Group count: $Count "`n" Getting members of: $DisplayName"*

*$EmailAddress=$\_.EmailAddress*

*$GroupType=$\_.GroupType*

*$ObjectId=$\_.ObjectId*

*$Recipient=""*

*$RecipientHash=@{}*

*for($KeyIndex = 0; $KeyIndex -lt $RecipientTypeArray.Length; $KeyIndex += 2)*

*{*

*$key=$RecipientTypeArray[$KeyIndex]*

*$Value=$RecipientTypeArray[$KeyIndex+1]*

*$RecipientHash.Add($key,$Value)*

*}*

*$Members=Get-MsolGroupMember -All -GroupObjectId $ObjectId*

*$MembersCount=$Members.Count*

*#Filter for security group*

*if(($Security.IsPresent) -and ($GroupType -ne "Security"))*

*{*

*Return*

*}*

*#Filter for Distribution list*

*if(($DistributionList.IsPresent) -and ($GroupType -ne "DistributionList"))*

*{*

*Return*

*}*

*#Filter for mail enabled security group*

*if(($MailEnabledSecurity.IsPresent) -and ($GroupType -ne "MailEnabledSecurity"))*

*{*

*Return*

*}*

*#GroupSize Filter*

*if(([int]$MinGroupMembersCount -ne "") -and ($MembersCount -lt [int]$MinGroupMembersCount))*

*{*

*Return*

*}*

*#Check for Empty Group*

*elseif($MembersCount -eq 0)*

*{*

*$MemberName="No Members"*

*$MemberEmail="-"*

*$RecipientTypeDetail="-"*

*Print\_Output*

*}*

*#Loop through each member in a group*

*else*

*{*

*foreach($Member in $Members)*

*{*

*if($IsEmpty.IsPresent)*

*{*

*return*

*}*

*$MemberName=$Member.DisplayName*

*$MemberType=$Member.GroupMemberType*

*$MemberEmail=$Member.EmailAddress*

*if($MemberEmail -eq "")*

*{*

*$MemberEmail="-"*

*}*

*#Get Counts by RecipientTypeDetail*

*foreach($key in [object[]]$Recipienthash.Keys)*

*{*

*if(($MemberType -eq $key) -eq "true")*

*{*

*[int]$RecipientHash[$key]+=1*

*}*

*}*

*Print\_Output*

*}*

*}*

*#Order RecipientTypeDetail based on count*

*$Hash=@{}*

*$Hash=$RecipientHash.GetEnumerator() | Sort-Object -Property value -Descending |foreach{*

*if([int]$($\_.Value) -gt 0 )*

*{*

*if($Recipient -ne "")*

*{ $Recipient+=";"}*

*$Recipient+=@("$($\_.Key) - $($\_.Value)")*

*}*

*if($Recipient -eq "")*

*{$Recipient="-"}*

*}*

*#Print Summary report*

*$Result=@{'DisplayName'=$DisplayName;'EmailAddress'=$EmailAddress;'GroupType'=$GroupType;'GroupMembersCount'=$MembersCount;'MembersCountByType'=$Recipient}*

*$Results= New-Object PSObject -Property $Result*

*$Results | Select-Object DisplayName,EmailAddress,GroupType,GroupMembersCount,MembersCountByType | Export-Csv -Path $ExportSummaryCSV -Notype -Append*

*}*

*#Print Detailed Output*

*Function Print\_Output*

*{*

*$Result=@{'GroupName'=$DisplayName;'GroupEmailAddress'=$EmailAddress;'Member'=$MemberName;'MemberEmail'=$MemberEmail;'MemberType'=$MemberType}*

*$Results= New-Object PSObject -Property $Result*

*$Results | Select-Object GroupName,GroupEmailAddress,Member,MemberEmail,MemberType | Export-Csv -Path $ExportCSV -Notype -Append*

*}*

*Function main()*

*{*

*#Check for MSOnline module*

*$Module=Get-Module -Name MSOnline -ListAvailable*

*if($Module.count -eq 0)*

*{*

*Write-Host MSOnline module is not available -ForegroundColor yellow*

*$Confirm= Read-Host Are you sure you want to install module? [Y] Yes [N] No*

*if($Confirm -match "[yY]")*

*{*

*Install-Module MSOnline*

*Import-Module MSOnline*

*}*

*else*

*{*

*Write-Host MSOnline module is required to connect AzureAD.Please install module using Install-Module MSOnline cmdlet.*

*Exit*

*}*

*}*

*Write-Host Connecting to Office 365...*

*#Storing credential in script for scheduling purpose/ Passing credential as parameter*

*if(($UserName -ne "") -and ($Password -ne ""))*

*{*

*$SecuredPassword = ConvertTo-SecureString -AsPlainText $Password -Force*

*$Credential = New-Object System.Management.Automation.PSCredential $UserName,$SecuredPassword*

*Connect-MsolService -Credential $credential*

*}*

*else*

*{*

*Connect-MsolService | Out-Null*

*}*

*#Set output file*

*$ExportCSV="C:\M365Group-DetailedMembersReport\_$((Get-Date -format yyyy-MMM-dd-ddd` hh-mm` tt).ToString()).csv" #Detailed report*

*$ExportSummaryCSV="C:\M365Group-SummaryReport\_$((Get-Date -format yyyy-MMM-dd-ddd` hh-mm` tt).ToString()).csv" #Summary report*

*#Check for input file*

*if([string]$GroupIDsFile -ne "")*

*{*

*#We have an input file, read it into memory*

*$DG=@()*

*$DG=Import-Csv -Header "DisplayName" $GroupIDsFile*

*foreach($item in $DG)*

*{*

*Get-MsolGroup -ObjectId $item.displayname | Foreach{*

*$Count++*

*Get\_Members}*

*}*

*}*

*else*

*{*

*#Get all Office 365 group*

*Get-MsolGroup -All | Foreach{*

*$Count++*

*Get\_Members*

*}*

*}*

*#Open output file after execution*

*Write-Host `nScript executed successfully*

*if((Test-Path -Path $ExportCSV) -eq "True")*

*{*

*Write-Host Detailed report available in: $ExportCSV*

*Write-host Summary report available in: $ExportSummaryCSV*

*$Prompt = New-Object -ComObject wscript.shell*

*$UserInput = $Prompt.popup("Do you want to open output file?",`*

*0,"Open Output File",4)*

*If ($UserInput -eq 6)*

*{*

*Invoke-Item "$ExportCSV"*

*Invoke-Item "$ExportSummaryCSV"*

*}*

*}*

*Else*

*{*

*Write-Host No Group found.*

*}*

*}*

*. main*

The script exports output to **2 CSV files**. One with summary of group information and another with detailed group membership information

You can decide to change the export location from the following lines

*$ExportCSV=****"C:\M365Group-DetailedMembersReport\_$((Get-Date -format yyyy-MMM-dd-ddd` hh-mm` tt).ToString()).csv****" #Detailed report*

*$ExportSummaryCSV=****"C:\M365Group-SummaryReport\_$((Get-Date -format yyyy-MMM-dd-ddd` hh-mm` tt).ToString()).csv"*** *#Summary report*