How to use Get-ADGroupMember in PowerShell

lazyadmin.nl/powershell/get-adgroupmember

January 31, 2022

Active Directory Groups allow you to easily assign permissions or software to your users. But how do you get all members of a group? To export or update all users of an ADGroup we can use the **Get-ADGroupMember cmdlet** in **PowerShell**.

You can of course also view the group members in the Active Directory, but as you might have noticed, this is quite inconvenient. The advantage of using PowerShell to get all the group members is that you can easily export it to Excel for example, or use the results in other scripts.

In this article, we are going to take a look at how you can use the Get-ADGroupMember cmdlet in PowerShell.

Requirements

The Get-ADGroupMember cmdlet is part of the PowerShell Active Directory Module. This module is installed by default on domain controllers. However, I recommend installing the module on your own computer, so that you don't have to work on the domain controller.

Check out this article to install the PowerShell Active Directory Module on your computer or server.

Get all Group members with Get-ADGroupMember

The `Get-ADGroupMember` command will get all objects that are members of the group. This can be users, computers, and also other (nested) groups. To simply list all members of a group we can use the following command in PowerShell:

```
# Replace SG_M365_E3 with your group name
Get-ADGroupMember -Identity SG_M365_E3 | ft
```

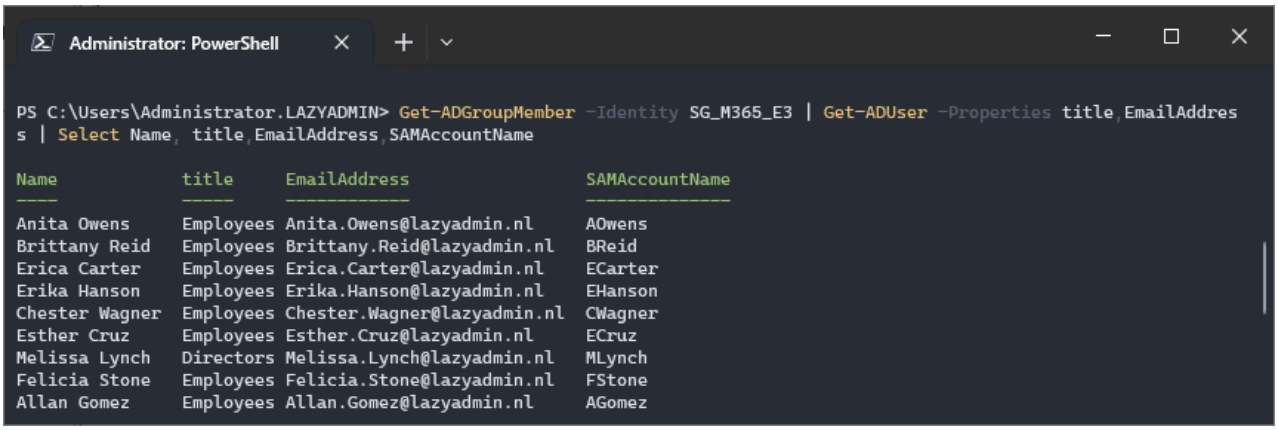
This will list all members of the group `SG_M365_E3` and format them into a table (ft).

Get-ADGroupMember

A simple list of all group members is in most cases not what you are looking for. You probably want to get more user details, like the email address or the display name of all the users.

To do this we can simply pipe the `Get-ADuser` cmdlet behind it, request all the details that we need from each user in de group:

Get-ADGroupMember -Identity SG_M365_E3 | Get-ADUser -Properties DisplayName,EmailAddress | Select Name,DisplayName,EmailAddress,SAMAccountName

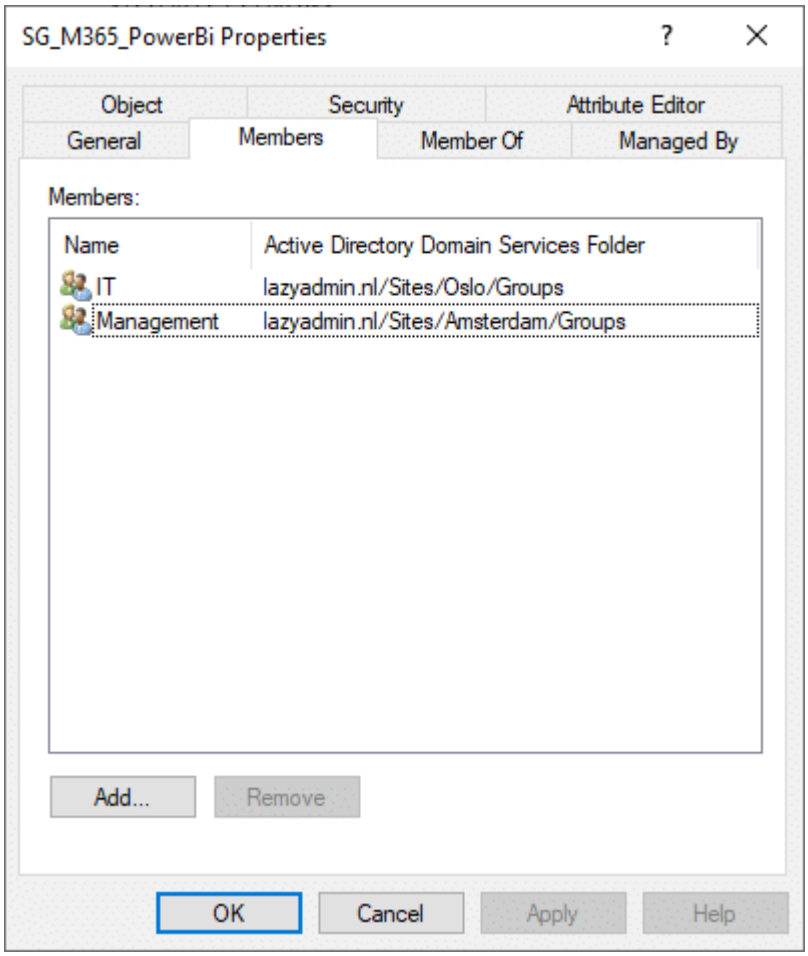


Get all group members with PowerShell

You can select any attribute that you need in the Get-ADUser cmdlet, but make sure that you also add it to the select add the end of the cmdlet.

Nested Groups

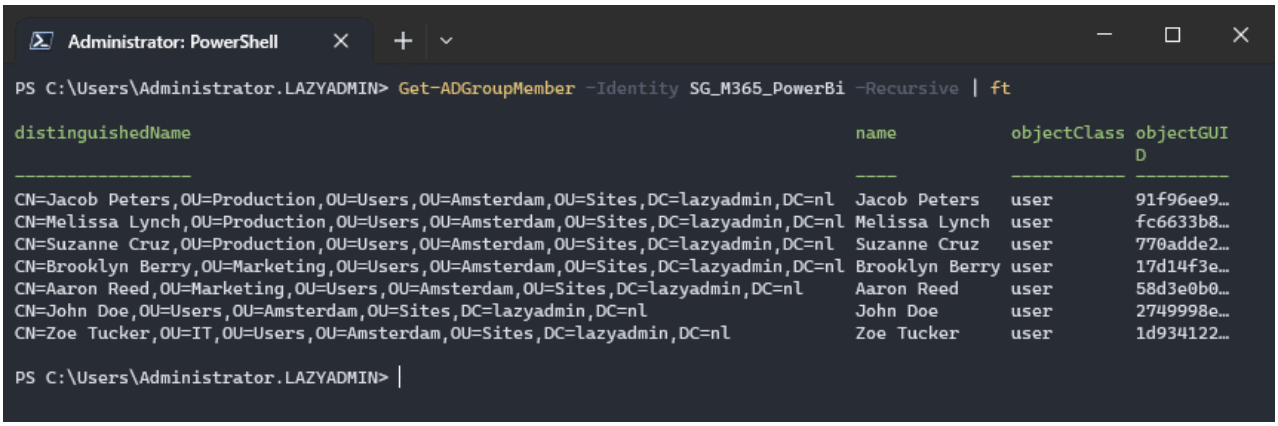
Nested groups are a common practice in the Active Directory. They allow you to assign permissions or policies to users based on their group membership. Let's take the following example where we have assigned PowerBi to the sales management and managing board:



To get the actual users that have access to PowerBi, we can't simply do `Get-ADGroupMember`, because that will only return the two groups:

What we want is to get the members of the two nested groups. To do this, we can use the `-recursive` parameter. This way the `Get-ADGroupMember` cmdlet will also go through all nested groups in the Active Directory.

Get-ADGroupMember -Identity SG_M365_PowerBi -Recursive | ft



Get Nested Group Membership

Get only Users, Computers or Nested groups

When you have a group mixed with users, computers, or nested groups, you might want to get only the users from that group. Or only the other nested groups.

To do this we can filter the results on the `objectClass` of the group member. This can be:

- user
- computer
- group

```
# Get only the users from a group
Get-ADGroupMember -Identity SG_PowerBi | Where-Object {$_.objectClass -eq "user"} | ft
# Or get only the nested groups
Get-ADGroupMember -Identity SG_PowerBi | Where-Object {$_.objectClass -eq "group"} | ft
```

Export Group Members to CSV with PowerShell

Most of the time when I use the Get-ADGroupMember cmdlet I want to export the results to Excel. To do this we can use the `Export-CSV` cmdlet in PowerShell. This will export all the results of your PowerShell cmdlet to a CSV file.

> Tip
>
> *You can also export directly to Excel with the PowerShell ImportExcel module*

Let's say we want to export all members of the SG_M65_E3 group with their email address to Excel:

```
Get-ADGroupMember -Identity SG_M365_E3 | Get-ADUser -Properties DisplayName,EmailAddress | Select Name,DisplayName,EmailAddress,SAMAccountName | Export-CSV -Path c:\temp\M365E3-users.csv -NoTypeInformation
```

This will export the results to a CSV file and store it in C:\Temp. The `-NoTypeInformation` cmdlet removes that header information from the CSV file. If you want to know more about exporting to CSV, then make sure you read this article.

The size limit for this request was exceeded error

When you are working with large groups that contain more than 5000 members, then it's possible that you get the error, **The size limit for this request was exceeded error**. The reason that you get this error is that Active Directory Web Services is limited to return is limited to return 5000 items by default.

There are two ways that we can resolve this problem. We can either change the configuration of the Active Directory Web Services to increase the limit. Or we can use the Get-ADGroup cmdlet. This cmdlet isn't limited.

Using the Get-ADGroup cmdlet

The Get-ADGroup cmdlet is probably the easiest way to get the members if the problem only occurs occasionally. We can simply get the group and expand the member property to view all the members of the groups.

Get-ADGroup "SG_M365_E3" -Properties Member | select -ExpandProperty member
If you want more information about the user then we can pipe the Get-ADUser cmdlet behind, just like we did before with the Get-ADGroupMember cmdlet:'

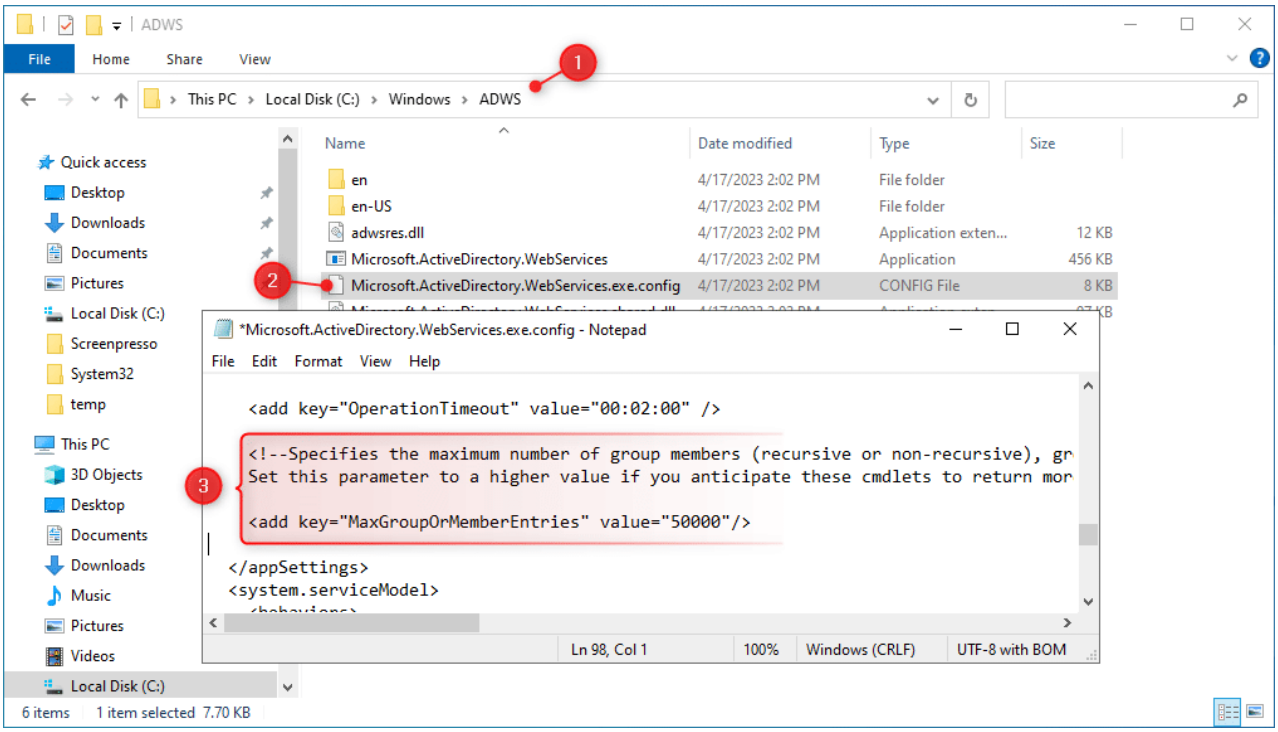
(Get-ADGroup "SG_M365_E3" -Properties Member).Member | Get-ADUser -Properties DisplayName,EmailAddress | Select Name,DisplayName,EmailAddress,SAMAccountName

Increasing the Limit

If you often need to retrieve large groups, then it's probably a better idea to increase the limit. To do this, you will need to have access to the domain controller.

1. In Explorer navigate to **C:\Windows\ADWS**
2. Open **Microsoft.ActiveDirectory.WebServices.exe.config** with **Notepad**
3. Add the code below, between the line **<add key="OperationTimeout" value="00:02:00" />** and **</appSettings>**

<!--Specifies the maximum number of group members (recursive or non-recursive), group memberships, and authorization groups that can be retrieved by the Active Directory module Get-ADGroupMember, Get-ADPrincipalGroupMembership, and Get-ADAccountAuthorizationGroup cmdlets.
Set this parameter to a higher value if you anticipate these cmdlets to return more than 5000 results in your environment.-->
<add key="MaxGroupOrMemberEntries" value="50000"/>
The result of the file should be like this:



After you have saved the file, you will need to restart the Active Directory Web Services service. You can use the PowerShell command below for it, or restart the services in the Services management console.

Restart-Service -name ADWS

Wrapping Up

As you have seen, it's pretty easy to get all members of a group with Get-ADGroupMember. Make sure that you use the `-recursive` parameter when you have nested groups.

When you want to export the members of the group to Excel, then make sure you try out ImportExcel module. This module allows you to create formatted Excel files straight from PowerShell.

If you have any questions, just drop a comment below.

Did you **Liked** this **Article**?
Get the latest articles like this **in your mailbox**
or share this article

I hate spam to, so you can unsubscribe at any time.