


# How to Add User to Group in PowerShell with Add-ADGroupMember

 lazyadmin.nl/powershell/add-user-to-group-add-adgroupmember

March 25, 2022

When you need to add multiple users to an Active Directory group then using PowerShell is really your best option. Instead of looking up all the users manually, you can quickly add users to the group with PowerShell.

Adding or removing users from AD groups is one of the daily tasks of a system administrator. When it's only a single user, then you can perfectly use the Active Directory Management console. But when you need to bulk update users, then PowerShell is the way to go.

In this article, I will explain how you can add single or multiple users to a group with PowerShell. Also, we are going to take a look at how to use a CSV file with users to add them to a group. And at the end, you will find two PowerShell scripts that allow you to easily import CSV files.

## PowerShell – add User to Group with Add-ADGroupMember

To add users to a group in PowerShell we are going to use the [Add-ADGroupMember](#) cmdlet. Make sure that you have [installed the PowerShell Active Directory Module](#) before we start.

The Add-ADGroupMember cmdlet can be used to add users, service accounts, computers, or even other groups to an AD Group. The cmdlet only requires the identity of the group to which you want to add the members to and a list of members to add.

To specify the group and members we can use the following values:

- distinguished name
- objectGUID (GUID)
- objectSid (security identifier)
- SAMAccountName (Security Account Manager account name)

So to add a single user to the group "SG\_PowerBi" we can use the group name, which is the SAMAccountName of the group. And also for the user, we are going to use the SAMAccountName:

```
# Adding user Alan Rhodes to the group SG_PowerBi
Add-ADGroupMember -Identity "SG_PowerBi" -Members arhodes
```

The command won't give any output, but we can verify the member with PowerShell using the [Get-ADGroupMember](#) cmdlet:

```
Get-ADGroupMember -Identity "SG_PowerBi" | ft
```

```
Select Administrator: Windows PowerShell
PS C:\> Get-ADGroupMember -Identity "SG_PowerBi" | ft

distinguishedName      name      objectClass objectGUID
-----
CN=Flenn Parker,OU=Production,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=nl Flenn Parker user ca3b2f67-1272-4e...
CN=Alan Rhodes,OU=Production,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=nl Alan Rhodes user ded676de-c50e-41...
CN=Gavin Stevens,OU=IT,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=nl Gavin Stevens user 9287c158-c780-46...
CN=Erik Parker,OU=IT,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=nl Erik Parker user 7aeec61-987b-4a...
CN=Sales Management,OU=Groups,DC=Lazyadmin,DC=nl Sales Management group 45a64d1f-d326-44...
CN=Managing Board,OU=Groups,DC=Lazyadmin,DC=nl Managing Board group 16eed8af-3475-48...

PS C:\>
```

PowerShell add user to group

## Adding Multiple Users to an Group

Adding a single user to a group can also be done with the Active Directory User and Computers console. But when you need to add multiple users to a group then using PowerShell can be a lot quicker.

There are a couple of ways to add multiple users to a group with PowerShell. The most common option is to simply specify the users:

# Adding the users Alan Rhodes, Flenn Parker and Mike Reyes to the group SG\_PowerBi  
Add-ADGroupMember -Identity "SG\_PowerBi" -Members arhodes, fparker, mreyes  
But that would require you to know each user's SAMAccountName. Another option is to first get the users based on a filter, the job title for example, and then add them to the group. To get the users, we are going to use the Get-ADUser cmdlet.

We are first going to get all the users with the job title "account manager", and then add each user to the group SG\_PowerBi:

```
# Get all users with the job title "Account Manager" and add them to the group SG_PowerBi
Get-ADUser -Filter "title -eq 'account manager'" | ForEach-Object { Add-ADGroupMember -Identity "SG_PowerBi" -Members $_ }
```

## Copy Group Members to another Group with PowerShell

It's also possible to copy group members to another group with PowerShell. But keep in mind, you can also add a group to another group in the Active Directory.

To copy the group members we are first going to get all the members of the existing group and copy them to the new group. For this, we are going to use the Get-ADGroupMember cmdlet.

```
# Get all the users from the group Sales Management and add them to the group SG_PowerBi
```

```
Get-ADGroupMember -Identity "Sales Management" | ForEach-Object { Add-ADGroupMember -Identity "SG_PowerBi" -Members $_ }
```

As an alternative, you could thus also add the group "Sales Management" to the group "SG\_PowerBi":

```
Add-ADGroupMember -Identity "SG_PowerBi" -Members "Sales Management"
```

## Add Users to Group from CSV with PowerShell

---

When you need to add a lot of users to a group, it's sometimes easier to use a CSV file. In the CSV file, we can easily add all the users. With PowerShell, we can read out the CSV file and add all the users to the correct group.

I have created a small PowerShell script that will import a CSV file and add all the users from the CSV file to the selected group. I have made the script as flexible as possible, so for the users, you can use either their display name, email, or UserPrincipalName.

The scripts require a group name and path to the CSV file. By default, it will use the comma as a delimiter for the CSV file and searches the users based on the display name.

For the CSV file you can create a simple list with the users, no need for headings inside the CSV file:

```
Bob Davis,  
Erik Parker,  
Jean Miles,  
Kelly Rices,  
Margie Baker,  
Mathew Scott,  
Owen Hopkins,
```

You can copy the script from below or download it [here from my Github page](#).

```
[CmdletBinding()]  
param (  
[Parameter(  
Mandatory = $true,  
HelpMessage = "Group name"  
)]  
[string] $groupName = "",  
[Parameter(  
Mandatory = $true,  
HelpMessage = "Path to CSV file"  
)]  
[string] $path = "",  
[Parameter(  
Mandatory = $false,  
HelpMessage = "CSV file delimiter"
```

```

)]
[string] $delimiter = ",",
[Parameter(
Mandatory = $false,
HelpMessage = "Find users on DisplayName, Email or UserPrincipalName"
)]
[ValidateSet("DisplayName", "Email", "UserPrincipalName")]
[string] $filter = "DisplayName"
)
Function Add-UsersToGroup {
<#
.SYNOPSIS
Get users from the requested DN
#>
process{
# Import the CSV File
$users = (Import-Csv -Path $path -Delimiter $delimiter -header "name").name
# Find the users in the Active Directory
$users | ForEach {
$user = Get-ADUser -filter "$filter -eq '$_'" | Select ObjectGUID
if ($user) {
Add-ADGroupMember -Identity $groupName -Members $user
Write-Host "$_ added to the group"
}else {
Write-Warning "$_ not found in the Active Directory"
}
}
}
}
# Load the Active Directory Module
Import-Module -Name ActiveDirectory
# Add user from CSV to given Group
Add-UsersToGroup
To add users, based on their display name, to the group "SG_PowerBi" from the CSV file
you can run the following command:

```

```

.\Add-UsersToGroup.ps1 -GroupName "SG_PowerBi" -Path c:\temp\users.csv

```

If you want to use the **UserPrincipalName** you will need to set the filter parameter:

```

.\Add-UsersToGroup.ps1 -GroupName "SG_PowerBi" -Path c:\temp\users.csv -Filter
"UserPrincipalName"

```

The script will show a warning if it was unable to find the user in the Active Directory. And it will list all the users that it has added to the group.

```
Administrator: Windows PowerShell
PS C:\Temp> .\Add-UsersToGroup.ps1 -GroupName "SG_PowerBi" -Path c:\temp\users.csv -Filter "DisplayName"
Bob Davis added to the group
Erik Parker added to the group
Jean Miles added to the group
WARNING: Kelly Rices not found in the Active Directory
Margie Baker added to the group
Mathew Scott added to the group
Owen Hopkins added to the group
PS C:\Temp>
```

The script doesn't check if a user is already a member of a group, the Add-ADGroupMember cmdlet does that already internally. You could add the parameter -DisablePermissiveModify to the cmdlet on Windows Server 2019 to throw an error if a user already exists.

## Adding Users to different Groups from CSV

I have also created a variant of the script that allows you to bulk add users to different groups. This way you can create a CSV file with in one column the users and the other group to which you want to add them to.

This CSV file needs column names, so the first column must be labeled **User** and the second column **Group**

```
User,Group
Bob Davis,SG_PowerBi
Erik Parker,SG_Visio
Jean Miles,SG_Visio
Kelly Rices,SG_PowerBi
Margie Baker,SG_PowerBi
Mathew Scott,SG_Visio
Owen Hopkins,SG_PowerBi
```

This time we don't need to specify the group, only the path to the CSV file, optionally the delimiter and filter.

```
.\Add-UsersToDiffGroup.ps1 -path c:\temp\users.csv -filter "DisplayName"
```

```
Administrator: Windows PowerShell
PS C:\Temp> .\Add-UsersToGroup.ps1 -Path c:\temp\users.csv -Filter "DisplayName"
Bob Davis added to SG_PowerBi
Erik Parker added to SG_Visio
Jean Miles added to SG_Visio
WARNING: Kelly Rices not found in the Active Directory
Margie Baker added to SG_PowerBi
Mathew Scott added to SG_Visio
Owen Hopkins added to SG_PowerBi
PS C:\Temp>
```

You can copy the complete script below or download it [here from my Github page](#).

```

[CmdletBinding()]
param (
[Parameter(
Mandatory = $true,
HelpMessage = "Path to CSV file"
)]
[string] $Path = "",
[Parameter(
Mandatory = $false,
HelpMessage = "CSV file delimiter"
)]
[string] $Delimiter = ",",
[Parameter(
Mandatory = $false,
HelpMessage = "Find users on DisplayName, Email or UserPrincipalName"
)]
[ValidateSet("DisplayName", "Email", "UserPrincipalName")]
[string] $Filter = "DisplayName"
)
Function Add-UserToGroup {
<#
.SYNOPSIS
Get users from the requested DN
#>
process{
# Import the CSV File
$users = Import-Csv -Path $path -Delimiter $delimiter
# Find the users in the Active Directory
$users | ForEach {
$user = Get-ADUser -filter "$filter -eq '$($_.user)'" | Select ObjectGUID
if ($user) {
Add-ADGroupMember -Identity $_.Group -Members $user
Write-Host "$($_.user) added to $($_.Group)"
}else {
Write-Warning "$($_.user) not found in the Active Directory"
}
}
}
}
# Load the Active Directory Module
Import-Module -Name ActiveDirectory
# Add user from CSV to given Group
Add-UserToGroup

```

## Wrapping Up

---

Using PowerShell we can quickly add a user to a group. If you don't know the SAMAccountName of the user, then you can always look it up with the Get-AdUser cmdlet. If you are new to using PowerShell scripts, then make sure that you [read this article](#) as well.

I hope you found this article useful, 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.