

Add FullAccess for a Mailbox without Automapping

- [How to connect with Powershell](#) (Exchange/Msol)
- [Preparing the CSV file](#)
- [remove FullAccess + add FullAccess without automapping](#)
- [list access rights for SCV](#)

Script requires %userprofile%\Documents\automapping.CSV

1st column **Identity** (accessed mailbox)

2nd column **User** (accessing user)

First step for all bulk scripts below:

- Import the data for **Identity**, **User** from **CSV file** under

%userprofile%\Documents\automapping.CSV

```
$automapping = Import-Csv $ENV:UserProfile\Documents\automapping.CSV
```

Add FullAccess for a Mailbox without Automapping

- remove prior FullAccess + add new FullAccess without automapping

**copy/paste the below script and insert into Powershell*

```
$automapping | ForEach-Object { Remove-MailboxPermission -Identity $_.Identity -User  
$_.User -AccessRights FullAccess -Confirm:$false | Add-MailboxPermission -Identity  
$_.Identity -User $_.User -AccessRights FullAccess -AutoMapping:$false }
```

- list access rights for CSV

```
$automapping | ForEach-Object { Get-MailboxPermission -Identity $_.Identity -User  
$_.User }
```

- Remove 1

```
remove-MailboxPermission -Identity accessed@mailbox.com -User accessing@user.com -  
AccessRights FullAccess -Confirm:$false
```

- Add 1

```
Add-MailboxPermission -Identity accessed@mailbox.com -User accessing@user.com -  
AccessRights FullAccess -AutoMapping:$false
```

- Remove only from CSV

```
$automapping | ForEach-Object { Remove-MailboxPermission -Identity $_.Identity -User  
$_.User -AccessRights FullAccess -Confirm:$false }
```

- Add from CSV

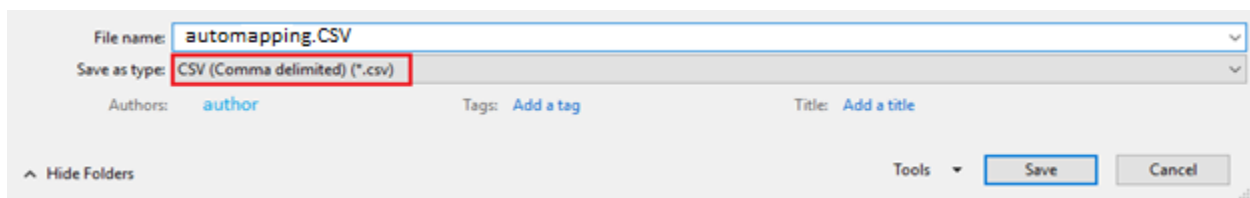
```
$automapping | ForEach-Object { Add-MailboxPermission -Identity $_.Identity -User
$_User -AccessRights FullAccess -AutoMapping:$false }
```

- remove prior FullAccess + add new SendAs without automapping from CSV

```
$automapping | ForEach-Object { Remove-MailboxPermission -Identity $_.Identity -User
$_User -AccessRights FullAccess -Confirm:$false | Add-MailboxPermission -Identity
$_.Identity -User $_.User -AccessRights SendAs -AutoMapping:$false }
```

- Preparing the CSV File

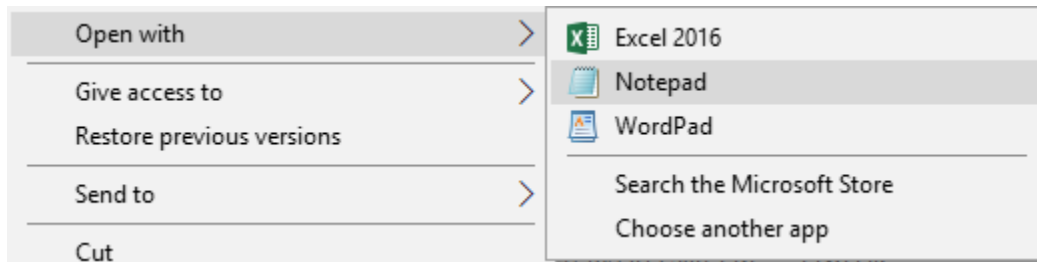
Make sure the used file is saved as **CSV Comma separated*



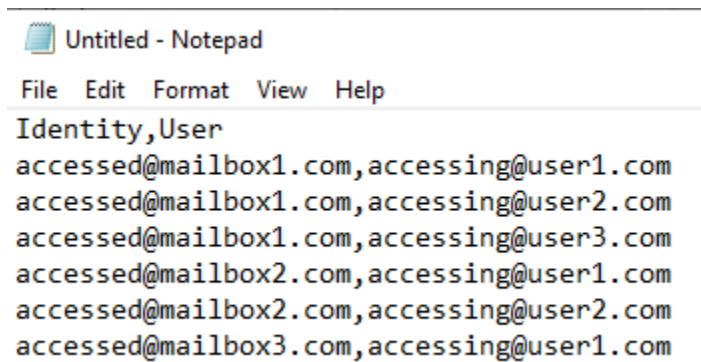
The data will look like this

	A	B
1	Identity	User
2	accessed@mailbox.com	accessing@user.com

In Notepad ...



.. it will look like this



CSV template format

Identity	User
accessed@mailbox.com	accessing@user.com

below the TXT template format

Identity,User
accessed@mailbox1.com,accessing@user1.com
accessed@mailbox1.com,accessing@user2.com
accessed@mailbox1.com,accessing@user3.com
accessed@mailbox2.com,accessing@user1.com
accessed@mailbox2.com,accessing@user2.com
[accessed@mailbox3.com,accessing@user1.com](#)

How to connect with Powershell (Exchange/MSOL)

- Download 64 Bit Version [PowerShell-6.2.2-win-x64.msi](#)

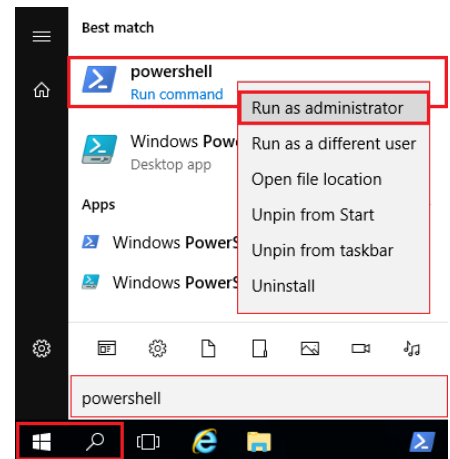
Get-Host | Select-Object Version

- *Official Documentation: Connect to Powershell*

*Below is not officially supported
Only provided "as is" for your convenience*

- [All-in-One](#)
- [First time only](#)
- [Step by step](#)

All-in-One



```
Set-ExecutionPolicy RemoteSigned -Force
```

```
Get-Credential "YOUR_ADMIN@DOMAIN.COM" | Export-Clixml $ENV:UserProfile\Documents\MyCredential.xml
```

**Stores encrypted password under %userprofile%\Documents\MyCredential.xml*

- copy below directly
- Alternative save as .TXT and rename .PS1
- Then type .\log.PS1 to connect to EXO / MSOL

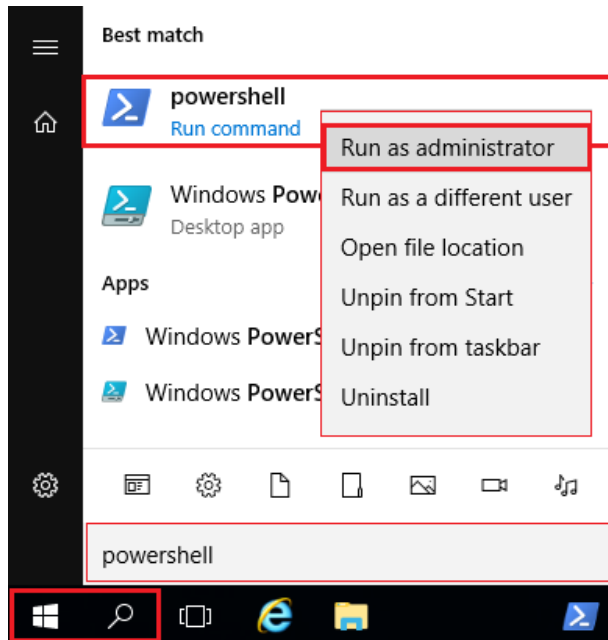
```
PS C:\Users\User> cd current_folder
PS current_folder> .\log.PS1
```

```
$cred = Import-Clixml $ENV:UserProfile\Documents\MyCredential.xml
```

```
Connect-MsolService -Credential $cred
```

```
$Session = New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri  
https://outlook.office365.com/powershell-liveid/ -Credential $cred -Authentication  
Basic -AllowRedirection
```

```
Import-PSSession $Session -DisableNameChecking
```



- Type "Powershell" near Start Menu
- search will open
- **right-click** the appearing Powershell icon
- choose "Run as **Administrator**"

- It should look like this

```
Administrator: Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\User> Set-ExecutionPolicy RemoteSigned -Force
PS C:\Users\User> $cred = Import-Clixml $ENV:UserProfile\Documents\MyCredential.xml
PS C:\Users\User> Connect-MsolService -Credential $cred
PS C:\Users\User> $session = New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri https://outlook.office365.com/powershell-liveid/ -Credential $cred -Authentication Basic -AllowRedirection
PS C:\Users\User> Import-PSSession $session -DisableNameChecking

ModuleType Version Name ExportedCommands
-----
Script 1.0 tmp_.dux {Add-AvailabilityAddressSpace, Add-DistributionGroupMember, Add
```

If not successful, make sure that you are logged in with:

- Admin
- correct password
- MFA disabled
- outdated Powershell Version *see above

- First time only

Powershell Download 64 Bit [PowerShell-6.2.2-win-x64.msi](#)

Check the installed Powershell version:

Get-Host | Select-Object Version

Below is not supported, only provided “as is” for your convenience

[Official Documentation to connect to Powershell for reference](#)

- Allow signed scripts – the **-Force** parameter will skip Y/N confirmation

Alternative: Allow unsigned scripts with: `Set-ExecutionPolicy Bypass -Force`

```
Set-ExecutionPolicy RemoteSigned -Force
```

modify below to your ADMIN login

only enter your credentials once and stored encrypted

located under %userprofile%\Documents\MyCredential.xml

```
Get-Credential "YOUR_ADMIN@DOMAIN.COM" | Export-Clixml  
$ENV:UserProfile\Documents\MyCredential.xml
```

- Step by step

you can save the below in a TXT file and change it to log.PS1

then put the File in current dir and run as .\log.PS1

* read out credentials from %userprofile%\Documents\MyCredential.xml

```
$cred = Import-Clixml $ENV:UserProfile\Documents\MyCredential.xml
```

connect to Microsoft Online (user creation, tenant settings ...)

```
Connect-MsolService -Credential $cred
```

If above is not successful, install MSOL Module

```
Install-Module MSOnline -Force
```

- Connect to Exchange

```
$Session = New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri  
https://outlook.office365.com/powershell-liveid/ -Credential $cred -Authentication  
Basic -AllowRedirection
```

```
Import-PSSession $Session -DisableNameChecking
```

- Alternative to MSOL:

AzureAD **sometimes referred to as "MSOL 2.0"*

Note: *different commands + only used in specific scripts*

```
Install-Module AzureAD -Force
```

```
Connect-AzureAD -Credential $cred
```

- Close the session with

```
Remove-PSSession $Session
```