How to use Set-ADUser in PowerShell

lazyadmin.nl/powershell/set-aduser

January 11, 2022

Need to update multiple users in your Active Directory? Or looking for a more efficient way of modifying users' properties? Then the Set-ADuser cmdlet in PowerShell is really going to help you.

We have all learned to manage our users through the Active Directory Users and Computers management console (ADUC). It's perfect to make quickly some changes to a single user. But when you need to update the properties of multiple users in different OU's then using PowerShell is way more efficient.

In this article, we are going to take a look at how to use the Set-ADuser cmdlet. What options do we have, and I will give you a couple of useful examples.

PowerShell Active Directory Module

Before we can start we first need to make sure that we have the PowerShell Active Directory Module installed. To check if you have the module installed, you can simply run the following command in PowerShell:

```
Get-Module -name ActiveDirectory
# Result:
ModuleType Version Name ExportedCommands
---------- ------- ---- ----------------
Manifest 1.0.0.0 ActiveDirectory {Add-ADCentralAccessPolicyMember, Add-ADComputerServiceAccount, Add-A...
```

If the module isn't listed then we need to install the RSAT module (Remote Server Administration). Use the following PowerShell command for this:

```
Install-WindowsFeature RSAT-AD-PowerShell
```

The module is automatically installed on the domain controller.

Install PowerShell Active Directory Module on Windows 10

But on Windows 10 or 11 we need to enable the RSAT feature. Instead of clicking through the settings screens, we are going to use PowerShell for this:

1. Press **Windows key + X** (or right-click start)
2. Open **Windows PowerShell (Admin)**
3. Enter the following command:

```
Add-WindowsCapability –online –Name “Rsat.ActiveDirectory.DS-LDS.Tools~~~~0.0.1.0”
```

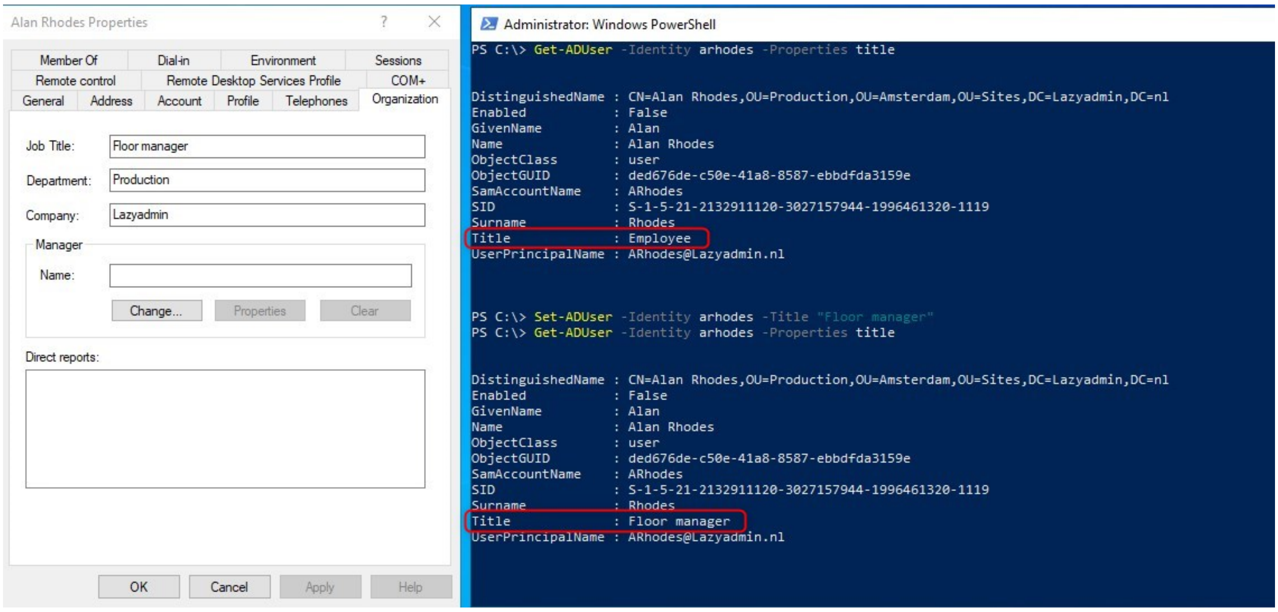
Using the Set-ADUser cmdlet

With the Set-ADUser cmdlet, we can modify all properties of an Active Directory user. To do this we can use one of the parameters of the cmdlet or use Add, Update, Replace parameter. All parameters of Set-ADUser are listed here in the Microsoft documentation.

Let's say we want to change the job title of our user Alan Rhodes. His current job title is employee, and we are going to change this to floor manager.

Set-ADUser -Identity arhodes -Title "Floor manager"
We select the user based on the userprincipalname (identity) and set the title to Floor manager. As you can see in the screenshot below, the `Set-ADUser` the cmdlet doesn't give any output.

But we can simply check the results in PowerShell with `Get-ADUser`



Job Title: Floor manager
Department: Production
Company: Lazyadmin
Name:
Direct reports:

```
PS C:\> Get-ADUser -Identity arhodes -Properties title


DistinguishedName : CN=Alan Rhodes,OU=Production,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=nl
Enabled           : False
GivenName         : Alan
Name              : Alan Rhodes
ObjectClass       : user
ObjectGUID        : ded676de-c50e-41a8-8587-ebbdfda3159e
SamAccountName    : ARhodes
SID               : S-1-5-21-2132911120-3027157944-1996461320-1119
Surname           : Rhodes
Title             : Employee
UserPrincipalName : ARhodes@Lazyadmin.nl



PS C:\> Set-ADUser -Identity arhodes -Title "Floor manager"
PS C:\> Get-ADUser -Identity arhodes -Properties title


DistinguishedName : CN=Alan Rhodes,OU=Production,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=nl
Enabled           : False
GivenName         : Alan
Name              : Alan Rhodes
ObjectClass       : user
ObjectGUID        : ded676de-c50e-41a8-8587-ebbdfda3159e
SamAccountName    : ARhodes
SID               : S-1-5-21-2132911120-3027157944-1996461320-1119
Surname           : Rhodes
Title             : Floor manager
UserPrincipalName : ARhodes@Lazyadmin.nl
```

Set-ADUser cmdlet

Set Multiple Attributes

We can also change or set multiple attributes at once. Let's say we also want to update Alan's phone number, office, and department attributes.

Set-ADUser -Identity arhodes -OfficePhone "(012)-157-8923" -Office "A3.20" -Department Operations

Clear AD User Attributes

Sometimes you need to remove or clear an attribute. To do this you will need to use the clear parameter, this will remove the content of all select fields. If we want to remove the mobile phone we can do the following:

Set-ADUser -identity arhodes -Clear mobile
Or to clear multiple AD attributes you can use

```
Set-ADUser -identity arhodes -Clear mobile,title,department
```

Add and Remove Attributes with Set-ADUser

Some attributes can contain multiple values, like **ProxyAddress** or **otherTelephone** for example. To set or update values in these attributes we can use the Add and Remove parameter in Set-ADUser.

You can set multiple values by using a comma-separated list:

```
Set-ADUser -Identity arhodes -Add
@{proxyAddresses="alan@lazyadmin.nl","rhodes@lazyadmin.nl"}
```

To remove a single value from the list we can simply specify which value we want to remove:

```
Set-ADUser -Identity arhodes -Remove @{proxyAddresses="rhodes@lazyadmin.nl"}
```

Active Directory Disable Account with Set-ADuser

We can also use the Set-ADuser cmdlet to enable or disable an account in the Active Directory. The `enabled` parameter is a boolean type, so we can set it to true or false.

To disable a user in your Active Directory you can use the following command:

```
# Disable a user
Set-ADUser -identity cparker -Enabled:$false
# Enable a user
Set-ADUser -identity cparker -Enabled:$false
```

Bulk Update AD Users

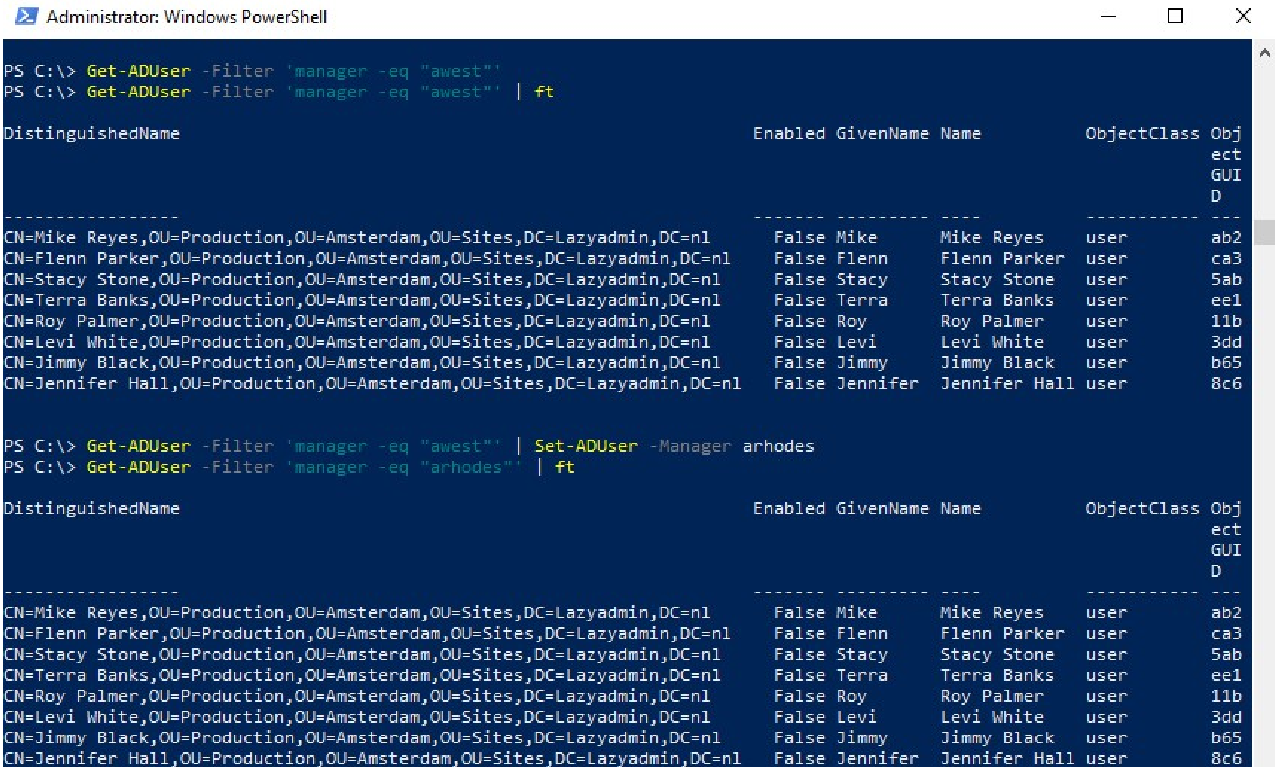
Until now we only updated a single user with PowerShell, but the true power of the Set-ADUser cmdlet is with bulk updating AD users. Changing an attribute on a single user can also easily be done through the management console.

But how do you change the manager of 10 users or update the company address of all your users? To do this we are going to use the `Get-ADUser` cmdlet together with `Set-ADUser`. Get allows us to filter the users, where we can use Set to update each user in the result.

In the first example, we made Alan Rhodes a floor manager. So let’s give him some direct reports. We are going to select all users that have currently Andre West as manager and change that to Alan Rhodes.

```
Get-ADUser -Filter 'manager -eq "awest"' | Set-ADUser -Manager arhodes
```

It’s always a good idea to check the results of your Get-ADUser cmdlet first before you change attributes with Set-ADuser.



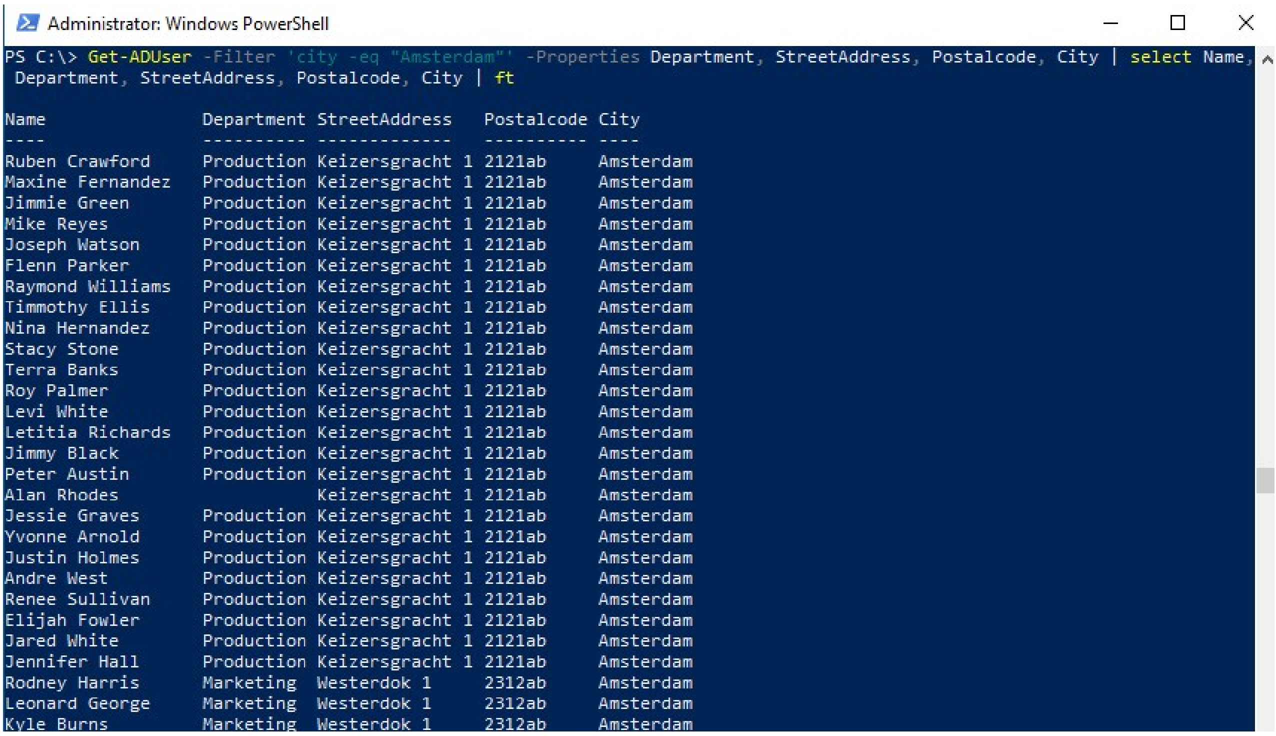
```
PS C:\> Get-ADUser -Filter 'manager -eq "awest"'
PS C:\> Get-ADUser -Filter 'manager -eq "awest"' | ft

DistinguishedName                                                       Enabled GivenName Name          ObjectClass Obj
                                                                                                                    ect
                                                                                                                    GUI
                                                                                                                    D
-----------------                                                       ------- --------- ----          ----------- ---
CN=Mike Reyes,OU=Production,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=nl      False Mike      Mike Reyes    user        ab2
CN=Flenn Parker,OU=Production,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=nl    False Flenn     Flenn Parker  user        ca3
CN=Stacy Stone,OU=Production,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=nl     False Stacy     Stacy Stone   user        5ab
CN=Terra Banks,OU=Production,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=nl     False Terra     Terra Banks   user        ee1
CN=Roy Palmer,OU=Production,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=nl      False Roy       Roy Palmer    user        11b
CN=Levi White,OU=Production,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=nl      False Levi      Levi White    user        3dd
CN=Jimmy Black,OU=Production,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=nl     False Jimmy     Jimmy Black   user        b65
CN=Jennifer Hall,OU=Production,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=nl   False Jennifer  Jennifer Hall user        8c6


PS C:\> Get-ADUser -Filter 'manager -eq "awest"' | Set-ADUser -Manager arhodes
PS C:\> Get-ADUser -Filter 'manager -eq "arhodes"' | ft

DistinguishedName                                                       Enabled GivenName Name          ObjectClass Obj
                                                                                                                    ect
                                                                                                                    GUI
                                                                                                                    D
-----------------                                                       ------- --------- ----          ----------- ---
CN=Mike Reyes,OU=Production,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=nl      False Mike      Mike Reyes    user        ab2
CN=Flenn Parker,OU=Production,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=nl    False Flenn     Flenn Parker  user        ca3
CN=Stacy Stone,OU=Production,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=nl     False Stacy     Stacy Stone   user        5ab
CN=Terra Banks,OU=Production,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=nl     False Terra     Terra Banks   user        ee1
CN=Roy Palmer,OU=Production,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=nl      False Roy       Roy Palmer    user        11b
CN=Levi White,OU=Production,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=nl      False Levi      Levi White    user        3dd
CN=Jimmy Black,OU=Production,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=nl     False Jimmy     Jimmy Black   user        b65
CN=Jennifer Hall,OU=Production,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=nl   False Jennifer  Jennifer Hall user        8c6
```

Filter on OU

The Get-ADUser also allows us to filter on OU. This can be really useful when you have a lot of users and want to limit the results or when you only need a selection of the users.

Let’s say that our marketing department is going to move to another office. We can’t simply select the users based on the office address, because we only want the marketing department. So what we can do is this:

Get-ADUser -Filter 'city -eq "Amsterdam"' -SearchBase "OU=Marketing,OU=Amsterdam,OU=Sites,DC=Lazyadmin,DC=NL" | Set-ADUser -StreetAddress "Westerdok 1" -PostalCode "2312ab"



```
PS C:\> Get-ADUser -Filter 'city -eq "Amsterdam"' -Properties Department, StreetAddress, Postalcode, City | select Name,
 Department, StreetAddress, Postalcode, City | ft

Name               Department StreetAddress   Postalcode City
----               ---------- -------------   ---------- ----
Ruben Crawford     Production Keizersgracht 1 2121ab     Amsterdam
Maxine Fernandez   Production Keizersgracht 1 2121ab     Amsterdam
Jimmie Green       Production Keizersgracht 1 2121ab     Amsterdam
Mike Reyes         Production Keizersgracht 1 2121ab     Amsterdam
Joseph Watson      Production Keizersgracht 1 2121ab     Amsterdam
Flenn Parker       Production Keizersgracht 1 2121ab     Amsterdam
Raymond Williams   Production Keizersgracht 1 2121ab     Amsterdam
Timmothy Ellis     Production Keizersgracht 1 2121ab     Amsterdam
Nina Hernandez     Production Keizersgracht 1 2121ab     Amsterdam
Stacy Stone        Production Keizersgracht 1 2121ab     Amsterdam
Terra Banks        Production Keizersgracht 1 2121ab     Amsterdam
Roy Palmer         Production Keizersgracht 1 2121ab     Amsterdam
Levi White         Production Keizersgracht 1 2121ab     Amsterdam
Letitia Richards   Production Keizersgracht 1 2121ab     Amsterdam
Jimmy Black        Production Keizersgracht 1 2121ab     Amsterdam
Peter Austin       Production Keizersgracht 1 2121ab     Amsterdam
Alan Rhodes                   Keizersgracht 1 2121ab     Amsterdam
Jessie Graves      Production Keizersgracht 1 2121ab     Amsterdam
Yvonne Arnold      Production Keizersgracht 1 2121ab     Amsterdam
Justin Holmes      Production Keizersgracht 1 2121ab     Amsterdam
Andre West         Production Keizersgracht 1 2121ab     Amsterdam
Renee Sullivan     Production Keizersgracht 1 2121ab     Amsterdam
Elijah Fowler      Production Keizersgracht 1 2121ab     Amsterdam
Jared White        Production Keizersgracht 1 2121ab     Amsterdam
Jennifer Hall      Production Keizersgracht 1 2121ab     Amsterdam
Rodney Harris      Marketing  Westerdok 1     2312ab     Amsterdam
Leonard George     Marketing  Westerdok 1     2312ab     Amsterdam
Kyle Burns         Marketing  Westerdok 1     2312ab     Amsterdam
```

Use a CSV list to update users

Another great way to bulk update users in your Active Directory is to use a CSV list. The advantage of the CSV file is that you can set different values for each user, using the convenience of an Excel file.

I have created in Excel a simple CSV file within the first column the display names of the users and in the second and third columns their new telephone numbers. Simply save the excel file as CSV.

```
Name;MobilePhone;OfficePhone
Bob Davis;(732)-016-9810;(933)-701-6542
Erik Parker;(732)-016-9720;(933)-701-6543
Jean Miles;(732)-016-9821;(933)-701-6544
Kelly Rice;(732)-016-9813;(933)-701-6545
Mathew Scott;(732)-016-9620;(933)-701-6546
Oscar May;(732)-016-9851;(933)-701-6547
Regina Clark;(732)-016-9811;(933)-701-6548
```

We can now create a small PowerShell script that will read the CSV file, find each user and update the attributes with the correct value:

```
Import-Csv -Delimiter ";" -Path c:\temp\PhoneNumberUpdate.csv | Foreach {
# Find user
$ADUser = Get-ADUser -Filter "name -eq '$($_.name)'"
if ($ADUser){
Set-ADUser -Identity $ADUser -MobilePhone $_.MobilePhone -OfficePhone $_.OfficePhone
}else{
Write-Warning ("Failed to update " + $($_.name))
}
}
```

Using the Instance parameter

A common issue with importing data from a CSV file is null values. Take the example above, not all users might have a mobile phone number. If we run the script using the method above, we will get an error because some fields don't contain a value. To solve this we can use the `-Instance` parameter.

With the Instance parameter, only values that are changed will be updated. This also allows us to handle empty values in the CSV file. I have changed the example CSV by removing some of the (mobile)phone numbers:

```
Name;MobilePhone;OfficePhone
Bob Davis;(732)-016-9810;(933)-701-6542
Erik Parker;;(933)-701-6543
Jean Miles;;(933)-701-6544
Kelly Rice;(732)-016-9813;(933)-701-6545
Mathew Scott;(732)-016-9620;
Oscar May;;(933)-701-6547
Regina Clark;(732)-016-9811;(933)-701-6548
```

To update the fields, we first get the user with the required properties, then set the new values and update the user with Set-ADUser. If the field is empty in the CSV file, then we will skip it. Keep in mind that this method doesn't clear the value in the Active Directory.

```
Import-Csv -Delimiter ";" -Path c:\temp\PhoneNumberUpdate.csv | Foreach {
# Find user
$ADUser = Get-ADUser -Filter "name -eq '$($_.name)'" -Properties MobilePhone,OfficePhone
if ($ADUser){
if ($_.MobilePhone) {$ADUser.MobilePhone = $_.MobilePhone}
if ($_.OfficePhone) {$ADUser.OfficePhone= $_.OfficePhone}
Set-ADUser -Instance $ADUser
}else{
Write-Warning ("Failed to update " + $($_.name))
}
}
```

In this article, I go more into detail on how to use a CSV file to update user attributes in the Active Directory.

Wrapping Up

The Set-ADUser cmdlet makes it really easy to update the attributes of multiple users in your Active Directory. Always make sure that you verify your filters with Get-ADUser first or by using the –`whatif` parameter.

If you have any questions, then 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.