Active directory management  
In a month on lunches

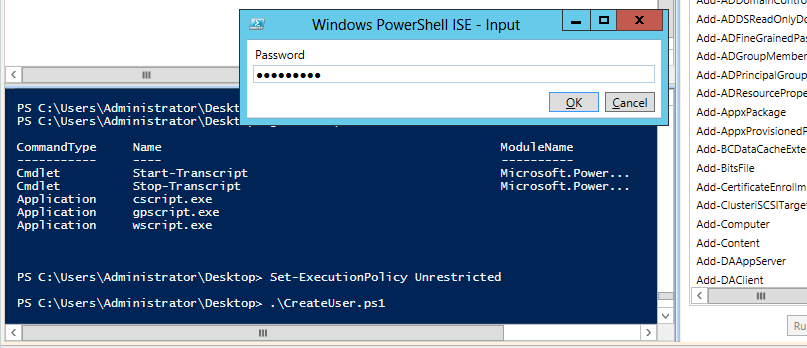
# Part 1

## Chapter 2: Creating user accounts

Command om Active directory cmdlets te verkrijgen: *Import-Module ActiveDirectory*Command om scripts te laten runnen: *Set-ExecutionPolicy Unrestricted*  
Daarnaast altijd powershell als administrator runnen.

### Aanmaken van een enkel account

Script om een nieuwe gebruiker toe te voegen:  
*$secpass = Read-Host “Password” –AsSecureString  
New-ADUser –Name “GREEN Dave” –SamAccountName dgreen –UserPrincipalName “dgreen@Assengraaf.nl” –AccountPassword $secpass –Path “cn=Users, dc= Assengraaf, dc = nl” –Enabled:$true*



Let wel, als je dit wil uitvoeren moet je uiteraard de toestemming hebben om scripts uit te voeren. Die toestemming kan je veranderen met het commando *Set-ExecutionPolicy.*

Er zal een pop-up verschijnen die je vraagt om een wachtwoord op te geven. Zeer belangrijk dat je dit zo doet want anders staat het wachtwoord gewoon in het script zelf wat een veiligheidsrisico met zich mee brengt.

Bij het aanmaken van een nieuwe gebruiker kan men ook gebruik maken van een al bestaand account. Als er verschillende instellingen hetzelfde zijn kan dit dus handig zijn.  
Bijvoorbeeld:

*$secpass = Read-Host “Password” –AsSecureString  
$user = Get-ADUser –Identity jgreen* ***–Properties memberof, office*** *New-ADUser –Name “GREEN Dave” –SamAccountName dgreen –UserPrincipalName “dgreen@Assengraaf.nl” –AccountPassword $secpass –Path “cn=Users, dc= Assengraaf, dc = nl” –Enabled:$true* ***–Instance $user***

### Aanmaken van meerdere accounts

Hierbij maak je gebruik van een .csv bestand waar al de gebruikers in staan.

Een script hiervoor:

*$secpass = Read-Host “Password” –AsSecureString  
Import-Csv gebruikersnamen.csv |  
foreach {  
 $name = “$($\_.LastName) $($\_.FirstName)”*

*New-ADUser –GivenName $($\_.FirstName) –Surname $($\_.LastName) –Name $name`  
-SamAccountName $($\_.SamAccountName) –UserPrincipalName “$($\_.SamAccountName)@Assengraaf.nl” –AccountPassword $secpass –Path “cn=Users, dc= Assengraaf, dc = nl” –Enabled:$true*

*}*

OPMERKING: $\_.FirstName kan je ook noteren als $PSItem.FirstName sinds powershell v3

### Aanmaken Managed service account

Deze accounts kunnen gebruikt worden om services zoals SQL server, exchange en IIS te beheren.  
Het commando om dit type account aan te maken is iets anders:

*New-ADServiceAccount –Name TestAccount –Path “CN= Managed Service Accounts,DC=Assengraaf, DC=nl” –DNSHostName TestAccount.Assengraaf.nl –Enabled $true  
Add-KdsRootKey –EffectiveImmediately*

Het boek gaat hier niet echt op in, het geeft gewoon aan dat indien je dit moet toe voegen je het zo kan doen.

## Chapter 3: Managing user accounts

Om van een bestaand account de gegevens te veranderen bestaat volgend commando:

*Set-ADUser –Identity “CN= GREEN Bill, CN=Users, DC=Assengraaf, DC=nl –OfficePhone “01234 567895”*

Met deze ‘Set-ADUser’ kan bij elk bestaand account een property toegevoegd of gewijzigd worden. Er zijn 4 paramters: Remove, Add, Replace en Clear.  
En ter info: na ‘-identity’ is het ook voldoende de SamAccountName ingeven, hier dan ‘bgreen’.

Soms kan het ook voorkomen dat je properties van een ander account wil kopiëren. Dit kan door bv. *$source = Get-ADUser –Identity bgreen –Properties OfficePhone, otherTelephone*.

Daarna:  
*Set-ADUser –Identity dgreen –OfficePhone $($source.officePhone) –Replace @{otherTelephone = $($source.otherTelephone)}*

Zo zijn er eerst 2 properties van een account meegegeven aan een variabele en daarna toegewezen aan een ander account. Dit op deze manier doen bespaart veel knip-en plakwerk in de GUI.

### Enabling en desabling accounts

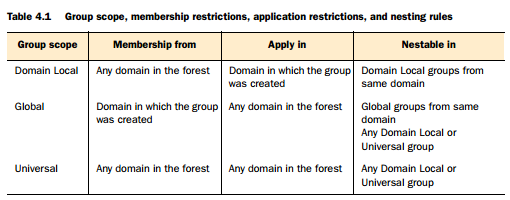
Vb: *Disable –ADAccount –Identity dgreen* of *Enable –ADAccount –Identity dgreen*

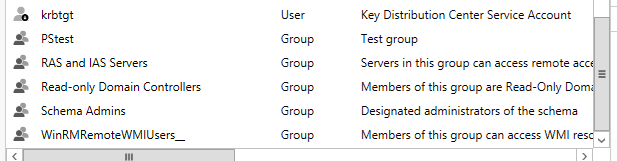
Of een script om te zien welke accounts niet geactiveerd zijn:  
*Search-ADAccount –AccountDisabled | Format-Table Name, DistinguishedName –AutoSize*

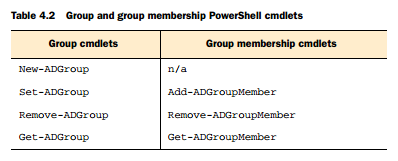
# Chapter 4: Managing groups

Een nieuwe gebruikersgroep aanmaken ziet era ls volgt uit:

*New-ADGroup –Name PStest –Path “CN=Users, DC=Assengraaf, DC=nl” –GroupCategory Security`  
–GroupScope global –Description “Test group”*

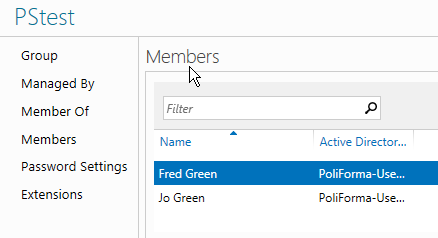
Ter verduidelijking, parameters name en path kennen we al. De ‘GroupCategory’ kan 2 waarden hebben, security of distribution. Wij zullen vaak security gebruiken, distribution is voor het aanmaken van e-mailgroepen in exchange.  
“Groupscope” daarna heeft 3 mogelijke waarden: DomainLocal, Global of Universal.

  
Na het uitvoeren van eerder genoteerd script zou de groep zonder problemen aangemaakt moeten zijn binnen de container Users.

   
Natuurlijk is het binnen powershell ook nog mogelijk na het aanmaken van de groep die te wijzigen zoveel je wil. Daar bestaan een aantal commandlets voor, ik geef ze even aan de hand van een tabel.

Bijvoorbeeld men wil alle gebruikers met de achternaam Green toevoegen aan de groep. Dit gaat zeer gemakkelijk binnen powershell aangezien er binnen haakjes kan gewerkt worden.

*Add-ADGroupMember –Identity PStest –Members (Get-ADUser –LDAPFilter “(sn=Green)”)*

(‘sn’ geeft hier de achternaam aan)

Groepen kunnen natuurlijk ook in bulk gemaakt worden, hierbij wordt simpelweg dezelfde syntax gevolgd als bij het creëren van gebruikers.

*Import-Csv –Path ./groups.csv |  
foreach{  
 New-ADGroup –Name $PSItem.Name –Path “CN= Users, DC=Assengraaf,DC=nl”*

*-GroupCategory Security –GroupScope global –Description $PSItem.Description  
}*

#Reminder: *$PSItem kan ook geschreven worden als $\_*

# Hoofdstuk 5: Troubleshooting users and groups