

10 PowerShell Opdrachten (oplossing)

10.1 Basis opdrachten PowerShell

Opgestart als gewone gebruiker

De antwoorden kunnen best bijgehouden worden in een document. Ofwel typ je de antwoorden ofwel schrijf je het gebruikte commando (= cmdlet met parameters) met ook de info voor de prompt of je voegt een knipsel toe. Op het examen worden knipsels verwacht als antwoord op de vragen. Zorg ervoor dat op het knipsel alles voor de prompt zichtbaar is, maar best ook de titel van het terminal venster.

Indien er meer cmdlets op het knipsel zichtbaar zijn, duid dan het juiste antwoord aan.

10.1.1 Opdracht 1: basis

- Open PowerShell gewoon en als administrator. Wat is het verschil?
de huidige map: bij gewoon is dit de gebruikersbestandslocatie en bij admin is dit de system32 map van Windows.
- Open PowerShell en PowerShell ISE. Type Get-C aan de prompt en ga op zoek naar de verschillende cmdlets die nu kunnen aangevuld worden. Is er een verschil in werken in de beide omgevingen?
in de ISE omgeving is er intellisense die al suggesties geeft tijdens typen, in de gewone terminal moet je die suggesties 1 voor 1 doorlopen met de tabtoets
- Wat is het meest opvallende verschil tussen commando's in CMD en cmdlets in PowerShell?
commando's zijn meestal kort => afkortingen, enkele letters → door veel gebruiken leer je deze wel van buiten, maar voor een beginnening is het niet altijd duidelijk wat een commando doet
cmdlets zijn tekstgebaseerd (verb-noun) waardoor ze duidelijker weergeven wat ze doen.

10.1.2 Opdracht 2: help

Test volgende cmdlets en geef een omschrijving van wat ze doen. Gebruik hiervoor de help pagina's. Som ook de belangrijkste parameters voor deze cmdlets op.

1. Update-Help :ingebouwde help updaten (als admin uitvoeren)
2. Get-Help oproepen van de helppagina's, belangrijkste parameter waarvan je de helppagina's wil oproepen
3. Get-Command ophalen van commando's (alias, functies en cmdlets)
4. Get-Service ophalen van een service (geeft ook de status): paramater naam of displaynaam
5. Start-Service service die wordt meegegeven opstarten
6. Get-Alias met alias → cmdlet weergeven waar deze voor staat met parameter -Definition en cmdlet → lijst van alle aliassen voor dit cmdlet

7. `Get-ChildItem` inhoud huidige map of map meegegeven pad ophalen (subfolders en bestanden).
8. `Show-Command` opent het command venster (hetzelfde dat in de ISE omgeving zit)
9. `New-Item` nieuw bestand of map aanmaken. verwacht een pad en als default is het itemtype een bestand, indien een map `Directory` als waarde voor parameter itemtype meegeven
10. `Set-Location` huidige map wijzigen in meegegeven (pad) map

Bekijk ook de help pagina's gebruikmakend van de parameter `-Examples` en `-Online`. Krijg je dezelfde help pagina's? Wat zijn hier de voordelen of nadelen van?

examples: enkele uitgewerkte voorbeelden hoe het cmdlet met bepaalde parameters kan gebruikt worden

online: de helppagina's van Microsoft worden opgeroepen.

10.1.3 Opdracht 3 instructies en cmdlets

Probeer de volgende instructies en noteer wat ze doen.

1. Navigatiepijl omhoog roept de vorige commandolijn op
2. `Get-History` geeft alle commandolijnen van deze sessie
3. `CLS` maakt het scherm zuiver
4. `Exit` sluit de terminal op een propere manier
5. `$PSVersionTable` geeft de versie van PowerShell
6. `ctrl + spatiebalk` suggesties voor aanvullingen (ISE)
7. `Set-Location` en dan tab toets mogelijkheden in huidige map (dus submappen)

10.1.4 Opdracht 4 Zoeken (naar cmdlets)

Zoeken naar cmdlets. Verschillende mogelijkheden bekijken. Vermits we weten dat een cmdlet opgesteld is als Verb-Noun kan bij het zoeken naar een cmdlet hiervan gebruik gemaakt worden.

Tip: cmdlet om cmdlets te krijgen: `Get-Command`

1. Welke parameter kan je gebruiken om enkel de cmdlets te krijgen. Geef naam en waarde van de parameter. `-commandType cmdlet // get-help -category cmdlet`
opmerking: gebruik `|out-GridView` voor een leesbaar resultaat.
2. Zoek naar alle cmdlets die met `Set-` beginnen gebruikmakend van een wildcard. Neem enkele waar je door de naam al kan veronderstellen waarvoor ze gebruikt worden. Controleer in de help pagina's of dit ook zo is.
`Get-Command -commandType cmdlet Set-* // get-command -verb set`

`Get-Command -Verb set | Where-Object -FilterScript{$_ .CommandType -eq "cmdlet"} ➔` filtert enkel de cmdlets eruit. Het gebruik van `variabele.property` is typisch dotnet.

3. Zoek naar alle cmdlets die met 'Get-' beginnen gebruikmakend van de parameter `-Verb`. Neem enkele waar je door de naam al kan veronderstellen waarvoor ze gebruikt worden. Controleer in de help pagina's of dit ook zo is.
`Get-Command -Verb Get | Where-Object -FilterScript{$_ .CommandType -eq "cmdlet"}`
4. Is er nog een manier om op zoek te gaan naar cmdlets die beginnen met een bepaald werkwoord zonder het cmdlet `Get-Command` te gebruiken?
`Get-Help // via PS-ISE and command add-on // get-verb //show-command`
en in de pop-up het werkwoord typen
5. Zoek cmdlets die betrekking hebben met het zelfstandig naamwoord 'Path' gebruikmakend van een wildcard.
`Get-Command -CommandType cmdlet *-Path`
Bij welke kan je door de naam al veronderstellen waarvoor ze gebruikt worden? Ga in de help pagina's kijken of dit juist is.
6. Zoek cmdlets die betrekking hebben met het zelfstandig naamwoord 'Item' gebruikmakend van de parameter `-Noun`.
`et-Command -Noun Item | Where-Object -FilterScript{$_ .CommandType -eq "cmdlet"}`
Bij welke kan je door de naam al veronderstellen waarvoor ze gebruikt worden? Ga in de help pagina's kijken of dit juist is.
7. Is er nog een manier om op zoek te gaan naar cmdlets die eindigen op een bepaald zelfstandig naamwoord, zonder het cmdlet `Get-Command` te gebruiken? `Get-Help // via PS-ISE and command add-on //show-command`
8. Start PowerShell terminal op in de verkenner. Hoe doe je dat?
`powershell` typen in adresbalk

10.1.5 Opdracht 5 aanmaken mappen en bestanden

Vertrek voor de oplossing van deze oefeningen steeds vanuit de gebruikersbestandsmap van de huidige ingelogde gebruiker.

1. Zoek de werking van `New-Item` op en bekijk welke parameters hier mogelijk zijn.
2. Maak met behulp van dit cmdlet een nieuwe map aan in de documentenmap van de ingelogde gebruiker met als naam 'Hoofdstuk7' met een relatief pad.

`New-item .\documents\Hoofdstuk7 -ItemType Directory`

3. Maak met behulp van dit cmdlet een nieuwe map aan in de documentenmap van de ingelogde gebruiker, nu met een absoluut pad gebruikmakend de omgevingsvariabele waarmee het pad naar de gebruikersmap van de huidige ingelogde gebruiker automatisch wordt ingevuld. Naam van de map

‘Hoofdstuk7_opdrachten’.

```
PS C:\Users\User> New-Item $env:UserProfile\Documents\Hoofdstuk10_opdrachten -ItemType Directory
```

4. Gebruik New-Item om in de map ‘Hoofdstuk7_opdrachten’ een bestand aan te maken met als naam ‘opdrachten.txt’ en als inhoud ‘Dit zijn de opdrachten voor hoofdstuk 7 PowerShell’.

```
PS C:\Users\User> New-Item -Path .\Documents\Hoofdstuk10_opdrachten -Name opdrachten.txt -Value "Dit zijn de opdrachten voor hoofdstuk 10 PowerShell"
```

5. Voeg aan dit bestand ‘Vraag 1: leg uit hoe je de help pagina’s kan oproepen voor een bepaald cmdlet’ toe. (Ga op zoek naar het juiste cmdlet, denk aan de Engelse term voor toevoegen)

```
PS C:\Users\User> "Vraag 1: Leg uit hoe je de help pagina's kan oproepen voor een bepaald cmdlet" | Out-File .\Documents\Hoofdstuk10_opdrachten\opdrachten.txt -Append
```

6. Maak nu met behulp van het cmdlet Set-Content hetzelfde txt bestand aan met dezelfde inhoud in de map ‘hoofdstuk7’.

```
PS C:\Users\User> Set-Content .\Documents\Hoofdstuk10\opdrachten.txt "Dit zijn de opdrachten voor hoofdstuk 10 PowerShell. `r`nVraag 1: leg uit hoe je de help pagina's kan oproepen voor een bepaald cmdlet"
```

let op: backtick gebruik: `r`n is new line

7. Controleer nu de inhoud van beide bestanden.

Zoek hiervoor het verschil in werking op tussen Get-Content

```
PS C:\Users\User> Get-Content .\Documents\Hoofdstuk10\opdrachten.txt
Dit zijn de opdrachten voor hoofdstuk 10 PowerShell.
Vraag 1: leg uit hoe je de help pagina's kan oproepen voor een bepaald cmdlet
```

Get-Item

```
PS C:\Users\User> Get-Item .\Documents\Hoofdstuk10\opdrachten.txt
```

Directory: C:\Users\User\Documents\Hoofdstuk10

Mode	LastWriteTime	Length	Name
----	-----	-----	----
-a----	12/6/2020 6:19 AM	133	opdrachten.txt

en Invoke-Item.

Verklaar de werking.

8. Maak in map ‘hoofdstuk10’ een nieuw bestand ‘testformulier’ met als inhoud ‘het doet er niet toe’.
9. Verplaats het document naar de map ‘Hoofdstuk7_opdrachten’.
10. Kopieer het bestand terug naar de map ‘Hoofdstuk7’.
11. Verwijder nu het bestand uit ‘Hoofdstuk7_opdrachten’.

```
PS C:\Users\User> Set-Content .\Documents\Hoofdstuk10\testformulier.txt "het doet er niet toe"
PS C:\Users\User> Move-Item .\Documents\Hoofdstuk10\testformulier.txt .\Documents\Hoofdstuk10_opdrachten\
PS C:\Users\User> Copy-Item .\Documents\Hoofdstuk10_opdrachten\testformulier.txt .\Documents\Hoofdstuk10
PS C:\Users\User> Remove-Item .\Documents\Hoofdstuk10_opdrachten\testformulier.txt
```

10.1.6 Opdracht 6 extra parameters

Soms zijn er ook parameters die geen waarde verwachten maar die wel extra diepgang geven of kracht bij het gebruikte cmdlet.

Ga voor de volgende cmdlets op zoek of deze parameters daar gebruikt kunnen worden en wat dit toevoegt aan de werking van het cmdlet.

Parameters:

- -Recurse
- -Hidden
- -System
- -File of -Directory
- -Force
- -Unique
- -First <getal>, -Last <getal>, -Skip<getal>
- -Tale

cmdlets:

- Get-ChildItem:
 - Recurse: ook de inhoud van alle submappen (maar niet de lege mappen)
 - Hidden: ook de verborgen mappen en bestanden
 - System: enkel systeemmappen en bestanden
 - Directory: enkel de mappen
 - Force: ook die dingen die anders verborgen blijven, maar security restrictie blijven intact
enz. zie help pagina voor parameters
- Remove-Item
- Copy-Item
- Move-Item
- New-Item
- Select-Object
- Sort-Object

10.1.7 Opdracht 7 Gebruik wildcards

Verklaar de werking van volgend cmdlet met parameter.

Get-item ??[aeiouy]*

→ gaat op zoek naar alle items in de huidige map die op de derde plaats een klinker in hun naam hebben.

10.1.8 Opdracht 8 Aliassen

Hoewel binnen deze cursus en dus ook op het examen steeds gevraagd wordt om de cmdlets te gebruiken, is het toch ook wel belangrijk dat je weet dat er alassen bestaan. In online video's worden deze alassen immers regelmatig gebruikt en dan is het zaak deze commando's ook te begrijpen en eventueel zelf te maken met het juiste cmdlet.

1. Zoek uit hoe je kan zien van welk cmdlet een bepaalde alias is. [Get-Alias 'alias'](#)
2. Zoek uit hoe je van een cmdlet kan achterhalen wat de mogelijke aliassen zijn. [Get-Alias -Definition 'cmdlet'](#)
3. Zoek de aliassen voor Get-ChildItem. Als er een bekende tussen staat(bijvoorbeeld uit Linux), test deze eens. Werkt dat? Zorg er nu voor dat ook de inhoud van de submappen getoond worden. Werken hier de parameters van Linux op dezelfde manier? Wat stel je vast?
[ls → parameters Linux werken niet](#)

10.1.9 Opdracht 9 Pipelines of pipe

1. Zoek de cmdlets die de output kunnen stijlen. Tip: format.

```
Format-Custom
Format-List
Format-SecureBootUEFI
Format-Table
Format-Wide
```

2. Dit zijn typische cmdlets die worden uitgevoerd na een pipe op de objecten die aan de pijplijn worden meegegeven.
Test de werking met het cmdlet Get-Command en verklaar telkens wat er gebeurt en of de output gegevens hetzelfde blijven.
Weet nog, als de output meer dan 1 pagina groot is kan in de gewone PS nog een pipe met 'more' toegevoegd worden om alles in pagina's te tonen.

```
PS C:\Users\User> get-command -CommandType Cmdlet |Format-List

Name           : Add-AppvClientConnectionGroup
CommandType    : Cmdlet
Definition     : Add-AppvClientConnectionGroup

Path           :
AssemblyInfo   :
DLL            :
HelpFile       :
ParameterSets  : {}
ImplementingType :
Verb           : Add
Noun           : AppvClientConnectionGroup

Name           : Add-AppvClientPackage
CommandType    : Cmdlet
Definition     : Add-AppvClientPackage

Path           :
AssemblyInfo   :
DLL            :
HelpFile       :
ParameterSets  : {}
ImplementingType :
Verb           : Add
Noun           : AppvClientPackage
```

```
PS C:\Users\User> get-command -CommandType Cmdlet |Format-Table
```

CommandType	Name	Version	Source
Cmdlet	Add-AppvClientConnectionGroup	1.0.0.0	AppvClient
Cmdlet	Add-AppvClientPackage	1.0.0.0	AppvClient
Cmdlet	Add-AppvPublishingServer	1.0.0.0	AppvClient
Cmdlet	Add-AppxPackage	2.0.1.0	Appx
Cmdlet	Add-AppxProvisionedPackage	3.0	Dism
Cmdlet	Add-AppxVolume	2.0.1.0	Appx
Cmdlet	Add-AzureAccount	5.1.2	Azure
Cmdlet	Add-AzureAnalysisServicesAccount	0.5.0	Azure.AnalysisServices
Cmdlet	Add-AzureApplicationGatewaySslCertificate	5.1.2	Azure
Cmdlet	Add-AzureCertificate	5.1.2	Azure
Cmdlet	Add-AzureDataDisk	5.1.2	Azure
Cmdlet	Add-AzureDisk	5.1.2	Azure
Cmdlet	Add-AzureDns	5.1.2	Azure
Cmdlet	Add-AzureEndpoint	5.1.2	Azure
Cmdlet	Add-AzureEnvironment	5.1.2	Azure
Cmdlet	Add-AzureHDInsightScriptAction	5.1.2	Azure
Cmdlet	Add-AzureInternalLoadBalancer	5.1.2	Azure
Cmdlet	Add-AzureKeyVaultCertificate	4.3.0	AzureRM.KeyVault

3. Doe dit zelfde, maar ga nu aan het format cmdlet als parameter bepaalde eigenschappen toevoegen. Dit kunnen bijvoorbeeld Name en Source zijn.

```
PS C:\Users\User> get-command -CommandType Cmdlet |Format-List Name, Source
```

```

Name   : Add-AppvClientConnectionGroup
Source : AppvClient

Name   : Add-AppvClientPackage
Source : AppvClient

Name   : Add-AppvPublishingServer
Source : AppvClient

Name   : Add-AppxPackage
Source : Appx

Name   : Add-AppxProvisionedPackage
Source : Dism

Name   : Add-AppxVolume
Source : Appx

Name   : Add-AzureAccount
Source : Azure

```

4. Een andere manier om output te filteren kan met '|Select-Object'. Deze verwacht dan natuurlijk ook de waarde waarop moet gefilterd worden. Zoek de werking eens uit zoals punt 3, maar dus nu niet met format maar met

Select-Object. [output is een tabel](#)

```
PS C:\Users\User> get-command -CommandType Cmdlet |Select-Object Name, Source

Name                                     Source
----                                     -
Add-AppvClientConnectionGroup           AppvClient
Add-AppvClientPackage                   AppvClient
Add-AppvPublishingServer                 AppvClient
Add-AppxPackage                         Appx
Add-AppxProvisionedPackage              Dism
Add-AppxVolume                         Appx
Add-AzureAccount                       Azure
Add-AzureAnalysisServicesAccount        Azure.AnalysisServices
Add-AzureApplicationGatewaySslCertificate Azure
Add-AzureCertificate                   Azure
Add-AzureDataDisk                      Azure
Add-AzureDisk                         Azure
Add-AzureDns                          Azure
Add-AzureEndpoint                     Azure
Add-AzureEnvironment                  Azure
Add-AzureHDInsightScriptAction          Azure
Add-AzureInternalLoadBalancer          Azure
Add-AzureKeyVaultCertificate            AzureRM.KeyVault
Add-AzureKeyVaultCertificateContact     AzureRM.KeyVault
Add-AzureKeyVaultKey                   AzureRM.KeyVault
Add-AzureKeyVaultManagedStorageAccount AzureRM.KeyVault
Add-AzureNetworkInterfaceConfig         Azure
Add-AzureNodeWebRole                  Azure
Add-AzureNodeWorkerRole                Azure
Add-AzureRmWebRole                    Azure
```

5. Er bestaan nog cmdlets met -Object. Zoek ze en test de werking. Deze cmdlets gebruiken na de pipe, want ze verwachten input.

```
PS C:\Users\User> get-command -CommandType Cmdlet *-object

CommandType Name
-----
Cmdlet       Compare-Object
Cmdlet       ForEach-Object
Cmdlet       Group-Object
Cmdlet       Measure-Object
Cmdlet       New-Object
Cmdlet       Select-Object
Cmdlet       Sort-Object
Cmdlet       Tee-Object
Cmdlet       Where-Object
```

6. Zoek ook het commando met 'Grid' in de naam (zonder Azure in de naam), dit is ook een veelgebruikt cmdlet. Zoek ook hier de werking van uit. (tip: - Examples)

```
CommandType Name
-----
Cmdlet       Get-AzureRmEventGridSubscription
Cmdlet       Get-AzureRmEventGridTopic
Cmdlet       Get-AzureRmEventGridTopicKey
Cmdlet       Get-AzureRmEventGridTopicType
Cmdlet       New-AzureRmEventGridSubscription
Cmdlet       New-AzureRmEventGridTopic
Cmdlet       New-AzureRmEventGridTopicKey
Cmdlet       Out-GridView
Cmdlet       Remove-AzureRmEventGridSubscription
Cmdlet       Remove-AzureRmEventGridTopic
Cmdlet       Set-AzureRmEventGridTopic
Cmdlet       Update-AzureRmEventGridSubscription
```


10.1.10 Opdracht 10 Variabelen

De werking met variabelen komt uitgebreider aan bod in het hoofdstuk PowerShell scripting en bij opdrachten over input en output.

- Typ \$env: en bekijk de mogelijkheden die er zijn van omgevingsvariabelen.
- Om alle omgevingsvariabelen te bekijken kan je deze ophalen met Get-Item en als pad Env: => nu worden ze getoond met het bijhorende pad.
- Voor het navigeren in PowerShell is er het cmdlet Set-Location.
 - Verander de huidige map naar de afbeeldingen map van de huidige gebruiker gebruikmakend van een omgevingsvariabele.
 - Start de CMD op met Invoke-Item en gebruik makend van een omgevingsvariabele.

10.1.11 Opdracht 11 Operatoren

- Maak de variabele (array) \$namen="Carine", "Niek", "Lode", "Marie", "Jef"
- Geef via pipe deze variabele door aan het cmdlet Where-Object om alle namen op te halen die alfabetisch voor de letter M komen. (Tip: variabele \$_)

```
PS C:\Users\User> $namen = "Carine", "Niek", "Lode", "Marie", "Jef"
PS C:\Users\User> $namen | Where-Object -FilterScript {$_ -lt "M"}
Carine
Lode
Jef
```

10.1.12 Opdracht 12 Input en Output

Om meerdere commando's na elkaar te laten uitvoeren, waar de verbinding niet kan met het pipe teken, kan shift en enter gebruikt worden om in de terminal naar een volgende regel te gaan. Zo kan je eigenlijk al een voorloper van een script maken.

Natuurlijk kan je ook in ISE het scriptvenster hiervoor gebruiken. Als je dan in het script deelvenster dit zet, kan je met F5 dit uitvoeren.

Vooraf voor oefeningen rond input en output is deze werking natuurlijk interessant omdat er nu waardes kunnen opgevraagd worden en deze waardes bijgehouden worden in een variabele om er dan verder mee aan de slag te gaan. Dit kan dan ook als geheel getest worden.

Onderstaande opdracht dus eerst helemaal opmaken en dan uitvoeren.

- Vraag aan de gebruiker de naam en stop die waarde in een string variabele met de naam persoon
- Vraag de leeftijd en stop die in een int met als naam leeftijd
- Zorg ervoor dat nu op het scherm de boodschap komt:
"Amai, (hier de naam van de persoon), jij ziet er wel nog goed uit voor iemand van (hier leeftijd) jaar."
- Toon het type van persoon

- Geef of het type van leeftijd een string is. (output= True or False)
- Maak nu de variabelen persoon en leeftijd terug leeg

```
PS C:\Users\User> $persoon=Read-Host "Wat is uw naam?"
$leeftijd = Read-Host "Wat is uw leeftijd?"
Write-Host "Amai, $persoon, jij ziet er wel nog goed uit voor iemand van $leeftijd jaar!"
$persoon.GetType().FullName
$leeftijd.GetType().FullName
Clear-Variable persoon
Clear-Variable leeftijd
Wat is uw naam?: Miet
Wat is uw leeftijd?: 25
Amai, Miet, jij ziet er wel nog goed uit voor iemand van 25 jaar!
System.String
System.String
```

Druk nu op enter (of F5) om dit kleine scriptje te runnen. **dus leeftijd is ook een string: Read-Host geeft altijd een string, tenzij je de castoperator gebruikt.**

Meer oefeningen rond input, output, variabelen en operatoren in hoofdstuk 11 rond scripting.

10.1.13 Opdracht 13 filehash

Neem de zip in hoofdstuk 5: oefeningen op blackboard en bereken hier de md5 hash van. Vergelijk deze met : F36B54633B8D2C009011F4020724DB2C

10.2 Admin opdrachten PowerShell

Voor vele van deze opdrachten moet je admin rechten hebben, dus start je best vanaf 10.2.2 PowerShell als admin.

10.2.1 Test in gebruikersbestandslocatie en verklaar

- Get-ChildItem -Force |Select-Object -Unique Extension
alle bestanden met een extensie die hier maar 1 keer voorkomt
- Get-ChildItem -Recurse|Format-Table -GroupBy Extension
alle mappen en submappen met inhoud, waarvan de inhoud samen gezet wordt per extensie
- Get-Help Get-ChildItem -Parameter * |Select-Object -Last 3 -Skip 2
- Get-ChildItem |Select-Object Name, CreationTime |Format-Table -AutoSize
- Get-ChildItem | Format-Table Name, CreationTime -AutoSize

10.2.2 Services 1

Geef een commandoregel om:

- service(s) te tonen, waarbij enkel de actieve (running) services getoond worden. Sorteert de services op naam en toon ze in een grid view (tip: help Grid-View)
Get-Service |Where-Object {\$_.Status -eq "Running"} | Sort-Object -Property Dis
- Exporteer de lijst naar een html file
Get-Service |Where-Object {\$_.Status -eq "Running"} | Sort-Object -Property

```
DisplayName, Status | ConvertTo-Html | out-file  
.\Documents\Demo\exportHTML.html  
Invoke-Item .\Documents\Demo\exportHTML.html
```

10.2.3 Services 2

Geef een commandoregel om alle services op te halen en met het Where-Object alsoutput de status van de service 'WinDefend' te geven.

```
get-service | Where-Object {$_.Name -eq 'WinDefend'} | Select-Object -Property  
Status
```

10.2.4 Services 3

Geef een commandoregel om 'Bitlocker Drive Encryption' uit te zetten (disable).

Denk eraan 'Bitlocker Drive Encryption' is de displayname, ga dus eerst op zoek naar de naam. Geef ook die commandoregel.

```
get-service -DisplayName *bitlock* | Stop-Service
```

10.2.5 Register 1

Registerkeys zijn in PowerShell stations. Dit wil dus zeggen dat je ernaartoe kan navigeren. Een registerbased station is een container en alles in die container kan bereikt worden op dezelfde manier als files en folders op een gewoon station.

Maak een snapshot voor deze opdrachten, voor het geval er iets fout gaat in het register.

Opdracht:

- navigeer naar de current user register hive (tip: (HKEY_CURRENT_USER = HKCU)
[Set-Location hkcu:/](#)
- haal de rechtstreekse sleutels van deze hive op, toon enkel de naam.
[Get-ChildItem | Select-Object -Property name](#)
- haal nu alle sleutels van deze hive op (naam) en ga op zoek naar sleutels voor batterij-alarm en geef hiervan de eigenschappen.
[In verschillende deelstappen:
Get-ChildItem | Select-Object -Property name
Get-ChildItem -Recurse | Where-Object {\\$_.Name -like "*bat*"}
Set-Location
HKEY_CURRENT_USER\AppData\Local\Microsoft\Windows\CurrentVersion\LowBatteryAlarm](#)

10.2.6 Register 2

Maak een snapshot voor deze opdrachten, voor het geval er iets fout gaat in het register. Deze opdracht wordt uitgevoerd in de root en dit is de enige die je eerst moet koppelen aan stationsnaam (mounten) om deze als drive te kunnen bereiken. Je geeft dan ook de naam van de drive mee: HKCR (HKEY_CLASSES_ROOT)

New-PSDrive -PSProvider registry -Root HKEY_CLASSES_ROOT -Name HKCR

Indien je in de gui de key voor het kopiëren naar map reeds hebt toegevoegd, verwijder deze sleutel dan eerst (met PowerShell). (tip: Remove-Item -Path)

Maak in de key : *HKEY_CLASSES_ROOT\AllFileSystemObjects\shellex\ContextMenuHandler* een nieuwe sleutel

met als naam "Naar map kopiëren"

met als waarde {C2FBB630-2971-11d1-A18C-00C04FD75D13}

Dit kan op verschillende manieren: met New-Item om het pad te definiëren met de naam erin of het pad en de naam apart definiëren. Je kan de waarde al meegeven

New-Item -Path HKCR:\AllFileSystemObjects\shellex\ContextMenuHandlers -Name "Naar map kopiëren" -Value "{C2FBB630-2971-11D1-A18C-00C04FD75D13}"

of deze later toevoegen met Set-ItemProperty.

Set-ItemProperty -Path HKCR:\AllFileSystemObjects\shellex\ContextMenuHandlers -Name "Naar map kopiëren" -Value "{C2FBB630-2971-11D1-A18C-00C04FD75D13}"

Bekijk welke parameters je dan moet meegeven. => dezelfde als met new-item

als je de sleutel maakt met new-item moet je ook alles meegeven om er dan de waarde in te zetten. Dus waarom in 2 stappen als het in 1 keer kan. Om het te laten werken moet wel de computer herstart.

Herstart de computer en controleer of deze mogelijkheid is bijgekomen in het contextmenu.

Stap voor stap:

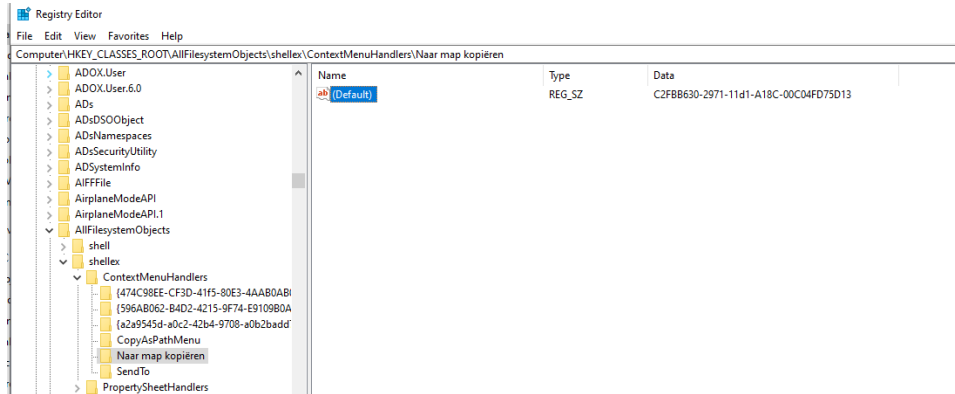
- *New-PSDrive -PSProvider Registry -Root HKEY_CLASSES_ROOT -name HKCR*
- *set-location HKCR:*
- *set-location .\AllFileSystemObjects*
- *set-location .\shellex*
- *Set-Location .\ContextMenuHandlers*
- *New-Item -Name "Naar map kopiëren" -Value C2FBB630-2971-11D1-A18C-00C04FD75D13*

In één keer: *New-item -path .\AllFileSystemObjects\shellex\ContextMenuHandlers\ -name "Naar Map Kopiëren" -value {C2...13} -force*

Opmerking 1: gebruik het cmdlet test-path om een path te controleren of het bestaat.

Opmerking 2: gebruik set-item om een waarde te plaatsen in een reeds bestaand

register. → beter is set-itemproperty



10.2.7 Netwerk 1

Geef het IPv4 en het IPv6 adres van je computer. Geef ook de gegevens van het subnetmask.

`Ipconfig /all` (werkt ook in Powershell) -> `Get-NetIpconfiguration`

`Get-NetIPConfiguration |format-list -Property interfaceAlias, IPv4address`

de gegevens van het subnetmask worden in PowerShell weergegeven door het aantal 1's in de subnetmasks. Dit heet 'prefixlength'

`Get-NetIPAddress` zonder parameters geeft alles, maar je kan ook specifiek voor IPv4 en IPv6 opvragen

```
Get-NetIPAddress -AddressFamily IPv4

IPAddress      : 192.168.113.129
InterfaceIndex  : 16
InterfaceAlias  : Ethernet0
AddressFamily   : IPv4
Type            : Unicast
PrefixLength    : 24
PrefixOrigin    : Dhcp
SuffixOrigin    : Dhcp
AddressState    : Preferred
ValidLifetime   : 00:19:14
PreferredLifetime : 00:19:14
SkipAsSource    : False
PolicyStore     : ActiveStore

IPAddress      : 127.0.0.1
InterfaceIndex  : 1
InterfaceAlias  : Loopback Pseudo-Interface 1
AddressFamily   : IPv4
Type            : Unicast
PrefixLength    : 8
PrefixOrigin    : WellKnown
SuffixOrigin    : WellKnown
AddressState    : Preferred
ValidLifetime   : Infinite ([Timespan]::MaxValue)
PreferredLifetime : Infinite ([Timespan]::MaxValue)
SkipAsSource    : False
PolicyStore     : ActiveStore
```

```
Get-NetIPAddress -AddressFamily IPv6

IPAddress      : fe80::b97d:7269:6a52:c0c2%16
```

```

InterfaceIndex      : 16
InterfaceAlias      : Ethernet0
AddressFamily       : IPv6
Type                : Unicast
PrefixLength        : 64
PrefixOrigin        : WellKnown
SuffixOrigin        : Link
AddressState        : Preferred
ValidLifetime       : Infinite ([Timespan]::MaxValue)
PreferredLifetime   : Infinite ([Timespan]::MaxValue)
SkipAsSource        : False
PolicyStore         : ActiveStore

IPAddress           : ::1
InterfaceIndex      : 1
InterfaceAlias      : Loopback Pseudo-Interface 1
AddressFamily       : IPv6
Type                : Unicast
PrefixLength        : 128
PrefixOrigin        : WellKnown
SuffixOrigin        : WellKnown
AddressState        : Preferred
ValidLifetime       : Infinite ([Timespan]::MaxValue)
PreferredLifetime   : Infinite ([Timespan]::MaxValue)
SkipAsSource        : False
PolicyStore         : ActiveStore

```

specifiek wat gevraagd werd:

```
Get-NetIPAddress |format-list AddressFamily, IPAddress, PrefixLength
```

```

AddressFamily : IPv6
IPAddress     : fe80::b97d:7269:6a52:c0c2%16
PrefixLength  : 64

AddressFamily : IPv6
IPAddress     : ::1
PrefixLength  : 128

AddressFamily : IPv4
IPAddress     : 192.168.113.129
PrefixLength  : 24

AddressFamily : IPv4
IPAddress     : 127.0.0.1
PrefixLength  : 8

```

10.2.8 Netwerk 2

Geef het fysisch adres van deze computer.

`Get-NetAdapter | Format-List -Property name, MacAddress`

10.2.9 Netwerk 3

Test de internetconnectie met pxl.be en geef elke hop (adres waarlangs de route loopt) weer.

```
Test-NetConnection -TraceRoute pxl.be
```

```

ComputerName      : pxl.be
RemoteAddress     : 193.190.154.242
InterfaceAlias    : Ethernet0
SourceAddress     : eigenadres
PingSucceeded     : True
PingReplyDetails (RTT) : 15 ms
TraceRoute        : eigenadres
                   : 8 verschillende adressen

```

```
tot hier en dan  
10.28.53.142  
193.191.8.29  
193.190.154.174  
193.190.154.242
```

10.2.10 Netwerk 4

Geef de volledige commandoregel om dynamische adressering uit te zetten.

Zet het ook terug aan.

[Get-NetAdapter -Name Ethernet0 | Set-NetIPInterface -DHCP Enabled of Disabled](#)

`Set-netIpInterface DHCP`

10.2.11 Hashing

Welke zijn alle mogelijke algoritmes waarmee je binnen PowerShell in Windows 10 hash van een file kan berekenen? Hoe heb je dit gevonden (cmdlet, commandoregel ...)

10.2.12 Gebruikers en groepen

- Maak een nieuwe gebruiker 'student1' en voeg hem toe aan de 'Users'. Geef de commandoregel(s).
- Maak een nieuwe groep '1TIN' en voeg de student hieraan toe (1 commandoregel)

10.2.13 Access control list

Log in als gebruiker student1 en maak in zijn documentenmap een document 'admin'. Test met behulp van het oproepen van de rechten welke rechten de 'user' hierop heeft.

Geef alle commandoregels.