

INTRODUCTION TO POWERSHELL

THE BASICS

MARK VAN DE WAARSENBURG



VOORSTELLEN: MARK VAN DE WAARSENBURG

- WERKZAAM IN IT SINDS EIND 1997
- TECHNISCH CLOUD CONSULTANT
- WERK SINDS 2010 MET POWERSHELL
- INTRODUCTION TO POWERSHELL AT



<https://www.linkedin.com/in/mark-van-de-waarsenburg-8b201414/>

<https://www.linkedin.com/company/d2c-it/>



<https://d2c-it.nl/blog/>



<https://github.com/D2CIT>

<https://github.com/D2CIT/IntroductionPowershell>



<https://d2c-it.nl>



AGENDA

01. Powershell?

05. Formatting

08. WMI

```
Function get-continue
{
<#
.Synopsis
    get-continue
.DESCRIPTION
    get-continue
.EXAMPLE
    get-continue -message "Would you like to continue (yes/no)"
#>

[cmdletbinding()]

param(
    [Parameter(Mandatory=$false)]
    $message = "Would you like to continue (yes/no)" ,
    [switch]$destroy
)

Begin {
    if($Destroy){
        Write-host ##### -for Red -BackgroundColor black
        Write-host " LET OP : Je gaat een destroy uitvoeren!!" -for Red -BackgroundColor black
        Write-host ##### -for Red -BackgroundColor black
    }
    #Find TF
}

Process {
    $Res = $null
    $Res = Read-Host $message
    if($Res -eq "y" -or $Res -eq "Y" -or $Res -eq "yes" -or $Res -eq "Yes" -or $Res -eq "1" -or $Res -eq "True" -or $Res -eq "true"){
        $Continue = $true
    } else {
        $Continue = $false
    }
}

End {
    return $Continue
}
} #EndFunction
```



A screenshot of a Windows PowerShell window titled "Select Administrator: Windows PowerShell". The window shows the command "PS C:\Windows\system32> Install-WindowsFeature DHCP" entered at the prompt.

WAT IS POWERSHELL?

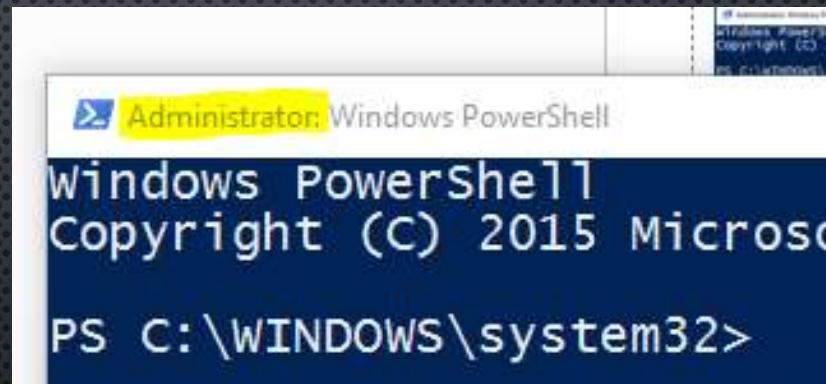
- MANAGEMENTTOOL VOOR SYSTEEMBEHEER
- GUI NIET OM TE AUTOMATISEREN
- COMMANDLINE TOOL, GEEN SCRIPTAAL ...
- VERGELIJKBAAR UNIX/LINUX
- OBJECT GEORIËNTEERDE SHELL (.NET)

BASIS : CONSOLE VERUS ISE

Powershell Console

Powershell ISE

Tip : Run as Administrator



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\WINDOWS\system32>
```



BASIS : CMDLETS AND ALIASES

Alias	CMdlet
dir, ls	Get-ChildItem
copy, cp	Copy-Item
dir /s	dir -recurse

```
PS C:\> get-alias -Definition get-childitem
 CommandType      Name
 -----          ---
 Alias           dir -> Get-ChildItem
 Alias           gci -> Get-ChildItem
 Alias           ls -> Get-ChildItem

PS C:\> get-Alias dir
 CommandType      Name
 -----          ---
 Alias           dir -> Get-ChildItem
```

Naamgeving: Werkwoord-Zelfstandig naamwoord (Verb-Noun)

```
get-eventlog -logname Security -ComputerName localhost,Server01 -Verbose
```

BASIS : TABCOMPLETION, AFKORTINGEN, ALIAS

- HANDIGE SHORTCUTS VOOR HET TYPEN VAN COMMANDO'S
- TAB COMPLETION
 - TYP: GET-SE[TAB]
 - TYP: GET-SERVICE –CO[TAB]
- PARAMETER AFKORTINGEN
 - TYP: GET-SERVICE –COMP LOCALHOST
- PARAMETER ALIASES
 - TYP: GET-SERVICE –CN LOCALHOST

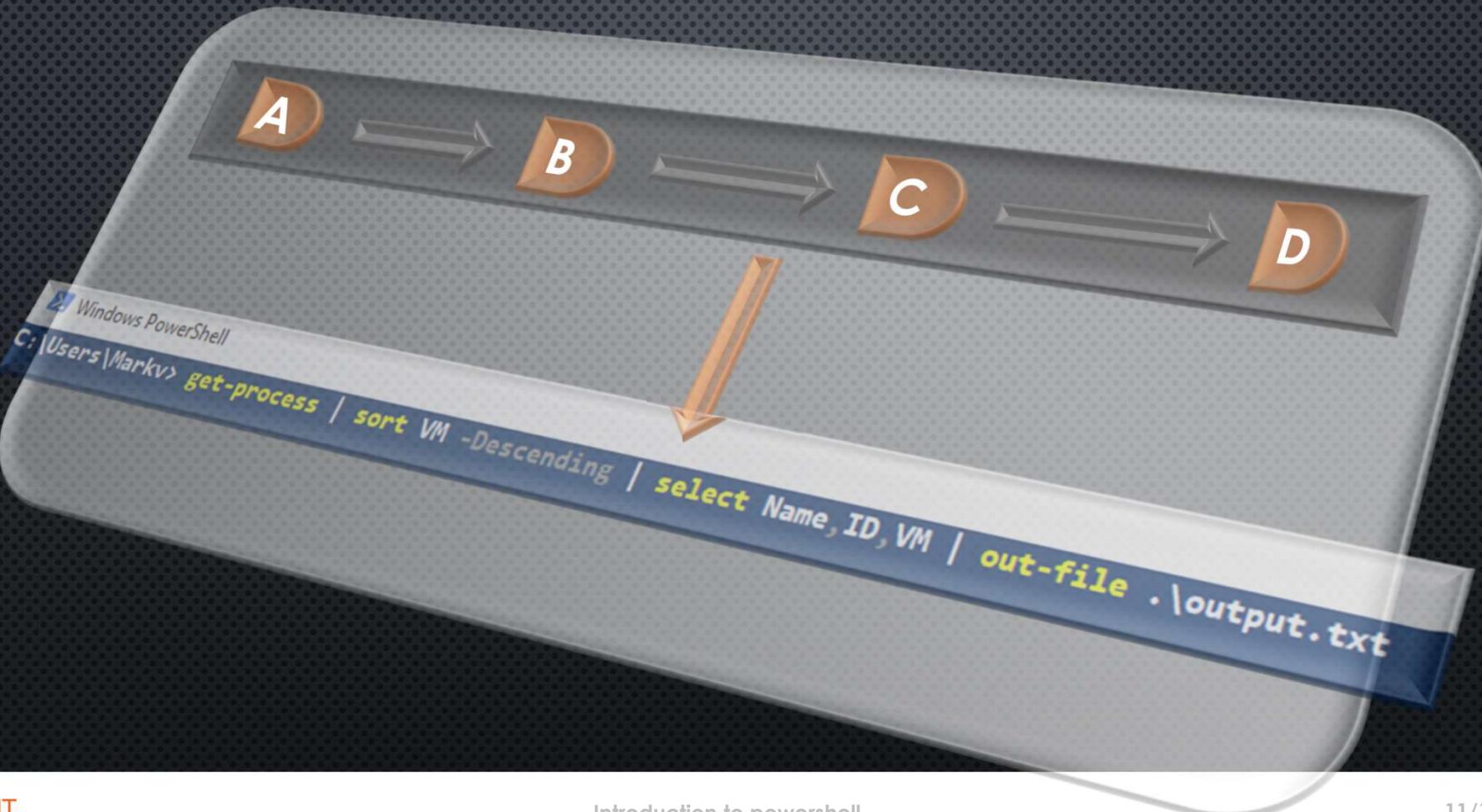
BASIS : PARAMETER

- NAMED EN POSITIONELE PARAMETERS
 - TYP: GET-SERVICE –COMPUTERNAME LOCALHOST
 - TYP: GET-SERVICE –NAME WUAUSERV
 - TYP: GET-SERVICE WUAUSERV

SYNTAX

```
Get-Service [[-Name] [<String[]>]] [-ComputerName [<String[]>]] [-DependentServices] [-Exclude [<String[]>]] [-Incl  
ude [<String[]>]] [-InformationAction {SilentlyContinue | Stop | Continue | Inquire | Ignore | Suspend}] [-Informat  
ionVariable [<System.String>]] [-RequiredServices] [<CommonParameters>]
```

BASIS : DE PIPELINE



DE PIPELINE (PARAMETER BINDING)



DE PIPELINE (PARAMETER BINDING)

- OBJECTEN WORDEN DOOR DE PIPELINE VERPLAATS VAN HET ENE CMDLET NAAR DE ANDERE
- HOE STUURT POWERSHELL DATA DOOR DE PIPELINE?
 - **ByValue**
 - **ByPropertyName**

HELP SYSTEEM

- HOE VINDEN WE NU AL DIE COMMANDO'S?
 - GET-COMMAND
 - GET-HELP, HELP, MAN
- HELP! ONZE HELP IS NIET UP TO DATE!
 - UPDATE-HELP ZORGT VOOR EEN UP TO DATE HELP
- HOE GEBRUIKEN WE DE HELP?
 - GET-HELP *SERV*
 - GET-HELP GET-SERVICE
 - GET-HELP GET-SERVICE –EXAMPLE

HELP SYSTEEM

- HELP INTERPRETATIE → GET-HELP <CMDLET>
 - [-VERPLICHT] <STRING> [-OPTIONEEL <STRING[]>]
[[-POSITIONEEL] <STRING[]>] [<COMMONPARAMETERS>]
- VOLLEDIGE HELP → HELP <CMDLET> -FULL
- PARAMETER WAARDEN → (STRING/INT/DATETIME)
- ABOUT HELP → GET-HELP ABOUT
- ONLINE HELP → GET-HELP <CMDLET> -ONLINE
GET-HELP <CMDLET> -SHOWWINDOW

QUIZ & DEMO

- Open je browser en open :



<https://kahoot.it/>

→ of installeer app op je telefoon

WAT IS HET JUISTE COMMANDO OM DE EIGENSCHAPPEN (PROPERTIES) VAN DE WINDOWS UPDATE SERVICE TE ZIEN ? (NAMED & POSITIONELE PARAMETERS)

get-process –name wuauserv

get-service wuauserv

get-process wuauserv

get-process –computers .

MET WELK COMMANDO KUN JE NIET ALLE E BEGINNEN MET GET?

Get-command –name get

Get-help –name get-*

Man get-*

Help get*

WELKE COMMANDO'S ZIJN ONJUIST?

Get-service –name wuauserv –c .

gsv wuauserv –computername .

Get-services -name wuauserv –com .

Get-service localhost –name wuauserv

HOE CHECK JE WAT DE ALIAS VAN HET CMDLET GET-PROCESS IS?

Get-Alias –name get-process

Get-Alias –Definition get-process

Get-Alias get-process

Get-Alias –name gps

WE RUNNEN ONDERSTAAND COMMAND. WAT IS HET RESULTAAT?

GET-PROCESS | SELECT-OBJECT NAME, ID | SORT-OBJECT VM -DECENDING

Alle processen gesorteerd op naam in alfabetische volgorde

Alle processen met alleen de properties naam en ID, gesorteerd op vm van klein naar groot.

Alle processen met alleen de properties naam en ID, gesorteerd op vm van groot naar klein.

Geen output.

WAT IS HET KORTSTE MOGELIJKE COMMAND VOOR :

GET-SERVICE -COMPUTER LOCALHOST | SORT-OBJECT STATUS | WHERE {\$_.NAME-LIKE "W*"}
gsv -c . | sort status | ? name -like w*

gsv -n w* | sort-object status

gsv w* | sort status

gsv | sort -status | ? name -like w*

WRITE-OUTPUT VS WRITE-HOST

Write-output



Out-default

Write-Host

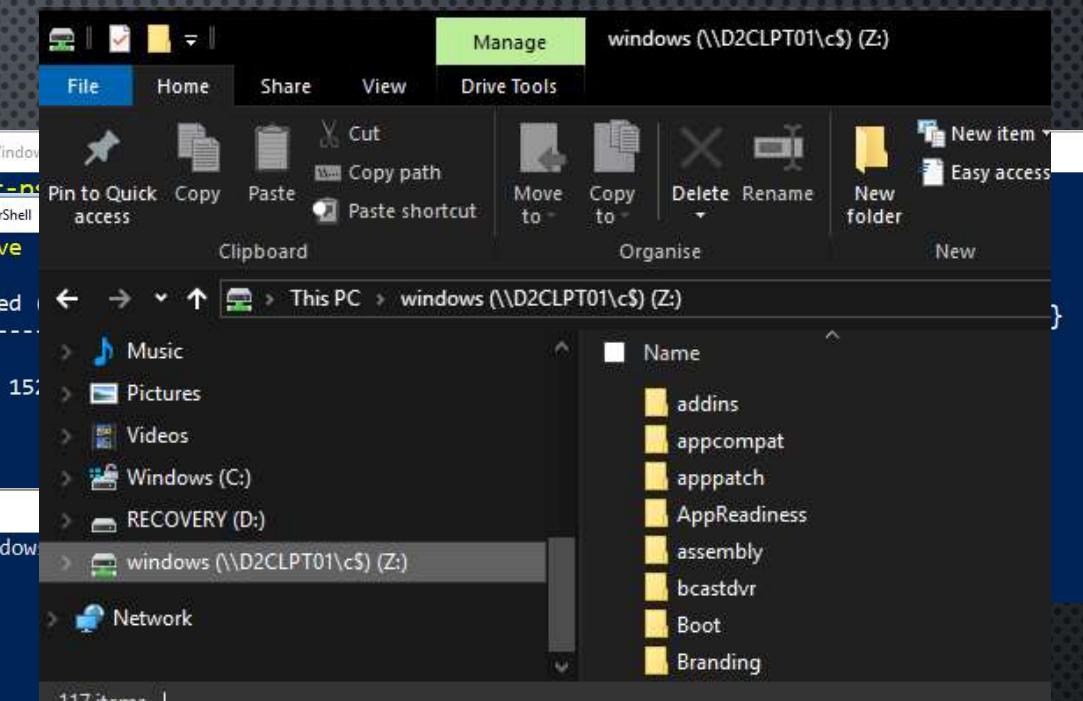


PS C:\> Get-Process							
Handles	NPM(I)	PM(K)	WS(K)	VM(M)	CPU(s)	ID	ProcessName
310	9	3348	532	85	0.83	2988	AdobeARM
257	11	4632	2444	88		1652	AppleMobileDeviceService
63	3	828	216	33		1624	armsvc
580	7	25240	19156	64	456	456	audiodg
61	4	1892	5292	34	3.32	916	conhost
518	9	1776	1500	67		456	cssrss
443	11	2152	26888	68		520	cssrss
1	9	6894	25276	1591	1.174,27	2416	dwm
1589	48	82972	45708	565,29		2924	explorer
116	6	3112	2584	21	0.69	2924	FileMon
156	6	2156	532	70	0.31	3140	hcmd
187	5	1788	532	55	0.09	2832	IABnotif
186	9	1716	1976	43		1860	Immersive
9	9	8	24	8		0	idle
92	4	1444	1008	50	0.42	3184	iufxpers
80	4	1200	532	48	0.08	3060	igfxtray
188	5	2688	1220	36		3688	iPodService
227	10	312	512	24	0.41	3408	ImmersiveHelper
958	14	5096	5508	36		624	lsass
178	4	1798	1240	14		632	lsm
69	4	2792	532	60	0.19	3468	MagicDisc

PROVIDER

- WAT IS EEN PROVIDER?
- GET-PSPROVIDER

```
Windows PowerShell
C:\> New-PSDrive -Name Z -PSProvider FileSystem -Root \\$env:COMPUTERNAME\c$\windows
Name      Used (GB)  Free (GB) Provider   Root
----      -----  -----  -----   -----
Z          423.34    39.40  FileSystem  \\D2CLPT01\c$\windows
C:\>
```



PROVIDER

- SET-LOCATION → CD

```
SET-LOCATION -PATH C:\WINDOWS  
CD C:\WINDOWS
```

- GET-CHILDITEM → DIR

```
GET-CHILDITEM HKCU:\SOFTWARE
```

- NEW-ITEM → CREËER EEN NIEUW ITEM

```
MKDIR TESTFOLDER
```

- WILDCARDS / LITERALPATH

```
GET-ITEM *.EXE  
SET-LOCATION -LITERALPATH HKCU:\SOFTWARE\*
```

OBJECTEN

- IN POWERSHELL BESTAAT ALLES UIT OBJECTEN
- TERMINOLOGIE:
- OBJECT -> DE KUBUS
- PROPERTY -> BREEDTE
- METHOD -> DRAAIEN
- COLLECTION -> MEERDERE KUBUSSEN



OBJECTEN : CORE COMMANDS

- CMDLET ALIAS
- SELECT-OBJECT SELECT
- SORT-OBJECT SORT
- WHERE-OBJECT WHERE / ?
- FOREACH-OBJECT FOREACH / %
- TIP: WAT DOET POWERSHELL MET HET OBJECT !!!!

OBJECTEN : CORE COMMANDS

SORTEREN EN SELECTEREN

- SORT-OBJECT

```
get-process | Sort-object VM -Descending
```

- SELECT-OBJECT

```
get-process | select-object -Property name, ID, VM, PM
```

```
get-process | Sort-object VM -Descending | select-object -Property name, ID, VM
```

- GET-MEMBER

```
get-process | Sort-object VM -Descending | select-object -Property name, ID, VM | GM
```

OBJECT : CORE COMMANDS

- **FILTEREN**

TWEE MODELLEN VOOR VERMINDEREN VAN RESULTATEN:

➤ LINKS FILTEREN

→ FILTEREN MET 1E CMDLET

```
get-service -name wuauserv
```

➤ RECHTS/ITERATIEF FILTEREN

→ PIPELINE CMDLET

```
get-service | where-object {$_.name -eq "wuauserv"}
```

```
get-service | where name -eq wuauserv
```



OBJECTEN : CORE COMMANDS

- **VERGELIJKEN**
VERGELIJKEN VAN TWEE OBJECTEN LEVERT ALTIJD TRUE OF FALSE
- -EQ / -NE
- -GE / -LE
- -GT / -LT
- -CEQ / -CNE / -CGT / CGE / CLE (HOOFDLETTER GEVOELIG)
- -AND / -OR
- -LIKE

OBJECTEN : CORE COMMANDS

- FILTEREN OBJECTEN UIT DE PIPELINE

WHERE-OBJECT (ALIAS → WHERE OF ?)

```
Get-Service | Where-Object -filter { $_.Status -eq 'Running' }
Get-Service | Where { $_.Status -eq 'Running' }
Gsv | ? { $_.Status -eq 'Running' }
```

FOREACH-OBJECT

(ALIAS → FOREACH OF %)

```
Get-content .\computers.txt |
  Foreach{
    Get-Service -ComputerName $_ | Where { $_.Status -eq 'Running' }
  }#Foreach

  Foreach($_ in (Get-content .\computers.txt)){
    Get-Service -ComputerName $_ | Where { $_.Status -eq 'Running' }
  }#Foreach
```

FORMATTING

- AAN HET EIND VAN DE PIPELINE KOMT ALTIJD OUT-DEFAULT
- STUURT OBJECTEN NAAR OUT-HOST
- OUT-HOST BEKIJKT HET OBJECT EN ZIJN REGELS
- REGELS EN DEFINITIES ZIJN OPGESLAGEN IN SPECIALE XML FILES IN \$PSHOME
 - DEFAULT VIEW?
 - DEFAULT PROPERTY SET?
 - HOEVEEL PROPERTIES?
- OUT-HOST GEBRUIKT DE FORMATTING CMDLETS
- JE KUNT DE DEFAULT VIEW ZELF AANPASSEN.

FORMATTING



QUIZ & DEMO

- Open je browser en open :



<https://kahoot.it/>

→ of installeer app op je telefoon

LIST ALLE RUNNING SERVICES EN SELECTEER ALLEEN DE RUNNING SERVICES ?

Get-service –status running

Get-service | where {\$_.status –like “running”}

get-status | where { \$_.service –eq
“running”}

Get-service | select-object running

CHECK WANNEER HET USERACCOUNT USERNAME VOOR HET LAATST IS AANGEPAST?

```
get-aduser -Identity username | select  
whenChanged
```

```
get-aduser -Identity username -Properties  
whenChanged
```

```
(get-aduser -Identity username -Properties  
* | where {$_.whenChanged -eq  
"Changed"})
```

```
(get-aduser -Identity username -Properties  
*).whenChanged
```

LIST ALLE PROCESSEN DIE BEGINNEN MET DE LETTER W ?

Get-process –name w

Get-process | where{\$_.name –like “w*”}

Get-process –name w*

Get-process | where {\$_.Name –eq “w*”}

HOE KUN JE ZIEN WAT VOOR PROPERTIES, METHODS EN TYPE OBJECT DE OUTPUT IS?

Get-wmiobject | where{\$_.object -like
Property}

Get-wmiobject | select -object

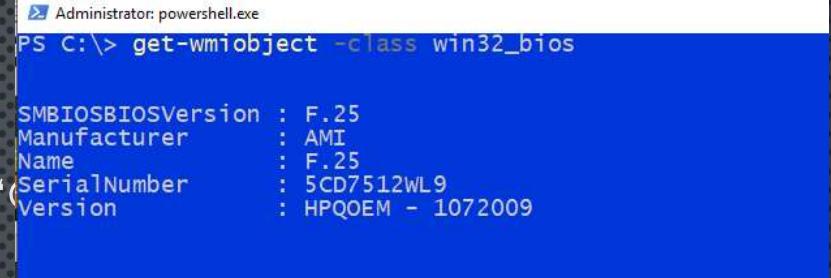
Get-wmiobject –type object

Get-wmiobject | get-member

VERANDER DE DEFAULT OUTPUT VAN HET COMMANDO
EN LAAT ALLEEN DE NAAM EN HET SERIENUMMER ZIEN.
ZORG ERVOOR DAT DE OUTPUT NOG BRUIKBAR IS.

```
get-wmiobject -class win32_bios | format-table -Property name,serialnumber
```

```
get-wmiobject -class win32_bios -Property name,serialnumber | format-table
```



```
Administrator: powershell.exe
PS C:\> get-wmiobject -class win32_bios

SMBIOSBIOSVersion : F.25
Manufacturer       : AMI
Name               : F.25
SerialNumber       : 5CD7512WL9
Version            : HPQ OEM - 1072009
```

```
get-wmiobject -class win32_bios | select name,serialnumber
```

```
get-wmiobject -class win32_bios | select name,serialnumber | format-table
```

EIND DAGDEEL 1

Opdracht : werken met objecten.

- <https://github.com/D2cit/introductionPowershell>
- Ga naar opdrachten → **Opdrachten avond 2.pdf**



VARIABELEN

- ZIE EEN VARIABELE ALS EEN DOOS
- HOE MAAK JE EEN VARIABELE?

```
$process = Get-Process  
$process = (get-process -name explorer)
```

- ALTERNATIEVE METHODES:

```
New-Variable -name Process -value (Get-Process)  
Get-Variable -name Process  
Set-Variable -name Process -Value (get-process -name explorer)
```



VARIABELEN

- VARIABELEN ZIJN PER POWERSHELL SESSIE
- ALLE VARIABELEN ZITTEN IN EEN PSDRIVE VARIABLE:
- NAMEN MOGEN HEEL LANG ZIJN
- HOUDT DE NAMEN LOGISCH `$SERVICE = GET-SERVICE`
- MEESTAL LETTERS, CIJFERS EN underscores
- SPATIES MOGEN, MAAR.. `${VARIABELE NAAM}`
- NIET MEER VARIABELE PREFIXEN MET TYPE `STRSERVICE`

VARIABELEN

- ER ZIJN TWEE TYPEN QUOTES
- HET ESCAPE KARAKTER (BACKTICK)
- MEERDERE OBJECTEN IN EEN VARIABELE (ARRAYS/HASHTABLES)
- COLLECTIE VARIABELEN BINNEN QUOTES (SUBEXPRESSION)
- TYPE BEPALEN VAN VARIABELEN

'WAARDE' & "WAARDE"

` , `N

\$VAR = 1,2,3

\$HT = @{'KEY'='VALUE'}

"\$(\$VAR[0])"

\$VAR.GETTYPE()

QUIZ & DEMO

- Open je browser en open :



<https://kahoot.it/>

→ of installeer app op je telefoon

HOE KUN MAAK JE EEN VARIABLE MET DE NAAM XYZ EN DE WAARDE 123 AAN IN POWERSHELL?

Create-Variable –name XYZ –value 123

New-variable –name XYZ –value 123

\$XYZ = 123

New-variable –name \$XYZ –value 123

HOE PAS JE DE WAARDE AAN VAN EEN VARIABLE?

\$XYZ = 345

get-variable XYZ | set-variable 345

get-variable –name xyz –value 345

Set-variable –name xyz –value 345

HOE VERWIJDER JE EEN VARIABLE?

Remove-variable \$XYZ

remove-variable –name XYZ

get-variable –name xyz | remove-variable

(dir variable:) | where {\$_.name -like "xyz"} |
remove-variable

SCRIPTING

- HET PLAATSEN VAN OPDRACHTEN IN EEN FILE
- EXTENTIE IS <FILENAME>.PS1
- SCRIPT-EDITORS
 - POWERSHELL ISE
 - VISUAL CODE
 - POWERGUI (QUEST)
 - NOTEPAD ++
 - PRIMALFORMS (NIET GRATIS)

SCRIPTING

- PS1 FILENAME EXTENSION STAAT NIET NAAR POWERSHELL
- SCRIPTS DRAAIEN ALLEEN MET HET VOLLEDIGE PAD
 - D:\SCRIPT.ps1
 - .\SCRIPT.ps1
 - SCRIPT.ps1 [TAB]
- SCRIPTS DRAAIEN NIET DOOR DE EXECUTION POLICY
 - RESTRICTED
 - ALLSIGNED
 - REMOTESIGNED
 - **Unrestricted**
 - **Bypass**

SCRIPTING : SCOPE

- GLOBAL
- SCRIPT
- FUNCTION

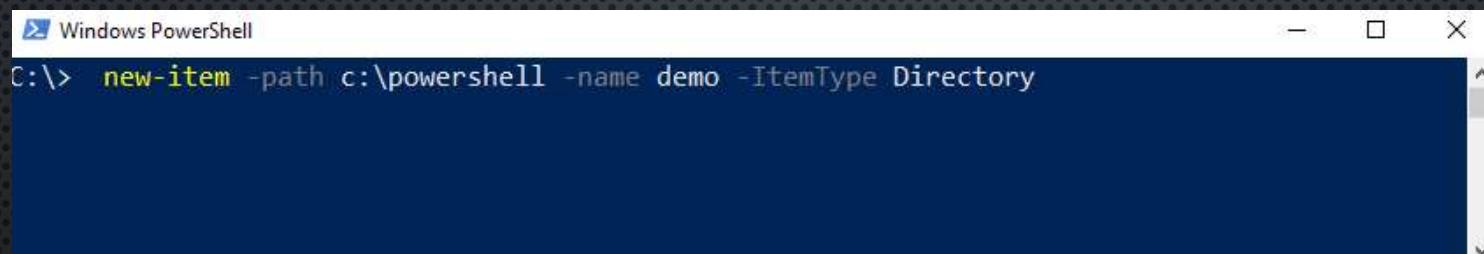


SCRIPTING

- PARAMETISE
- PARAM()
- BEGIN{ }
- PROCESS { }
- END { }
- TIP : CTRL + J

```
1 Function New-Folder {
2     <#
3     .Synopsis
4
5     Process{
6         #check if Folder exists
7         If(!(Test-Path -Path $Path)){
8             Write-verbose "    - Create Folder $Path"
9             Try{
10                 mkdir "$Path" -ErrorAction stop | Out-Null
11             }
12             End{
13                 Write-host "Endtime      : $((get-date).ToString('HH:mm:ss'))" -ForegroundColor Yellow
14                 Write-host "#####" -ForegroundColor Yellow
15
16                 return $result
17             }#End
18
19             $result = $true
20         }Else{
21             Write-verbose "    - Error creating folder $path !!"
22             $result = $false
23         }#end IF
24         }else{
25             Write-verbose " No action needed. Folder already exists"
26             $result = $true
27         } #end IF
28
29     }#process
30 }
```

DEMO



A screenshot of a Windows PowerShell window titled "Windows PowerShell". The window has a dark blue background. In the top-left corner, there is a small icon followed by the text "Windows PowerShell". On the right side of the title bar are three standard window control buttons: a minus sign for minimize, a square for maximize/minimize, and an X for close. The main area of the window contains a single line of text in white font: "C:\> new-item -path c:\powershell -name demo -ItemType Directory". The text is left-aligned and ends with a carriage return.

EIND DAGDEEL 1

Opdracht : maken van een script.

- <https://github.com/D2cit/introductionPowershell>
- Ga naar opdrachten → **3 - Introduction Powershell - Thuis Opdracht.pdf**



BELANGRIJKE CMDLETS

CTRL+J

- GET-HELP
- GET/SET-EXECUTIONPOLICY
- GET-SERVICE
- CONVERTTO-HTML
 - (GCM CONVERT*-*)
- EXPORT-CSV
 - (GCM EXPORT-*)
- SELECT-OBJECT
- GET-COMMAND
- GET-VERB
- GET-CHILDITEM
 - (GCM *ITEM*)
- GET-ALIAS
- SORT-OBJECT
- SELECT-OBJECT
- WHERE-OBJECT
- FOREACH-OBJECT
- FORMAT-LIST /FORMAT-TABLE/FORMAT-WIDE
- GET-PSPROVIDER
- GET-PSDRIVE
 - LET OP CONTEXT)
- NEW-VARIABLE (GET/SET/REMOVE)
 - (GCM *-VARIABLE)

AGENDA

01. Powershell?

02. BASIS

03. Pipeline

04. Objects
Core Commands

05. Formatting

06. Variables

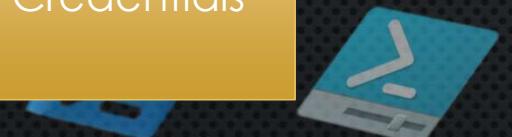
07. Scripting

08. WMI

09. Remoting

10. Jobs

11. Credentials



WMI (CIM)

- INTRODUCTIE WMI
- BROWSING/EXPLORING WMI
- REMOTE WMI
- ALTERNATIEVE CREDENTIALS
- FILTEREN WMI DATA

INTRODUCTION WMI





WMI & POWERSHELL

CMDLET: `GET-WMIOBJECT`

OPTIONS:

- `-CLASS` `<MANDATORY>`
- `-NAMESPACE` DEFAULT: `ROOT\CIMv2`
- `-FILTER` WILDCARD: `%` (NIET `*`)
- `-COMPUTERNAME`
- `-PROPERTY`
- `-LIST`

opdrachten

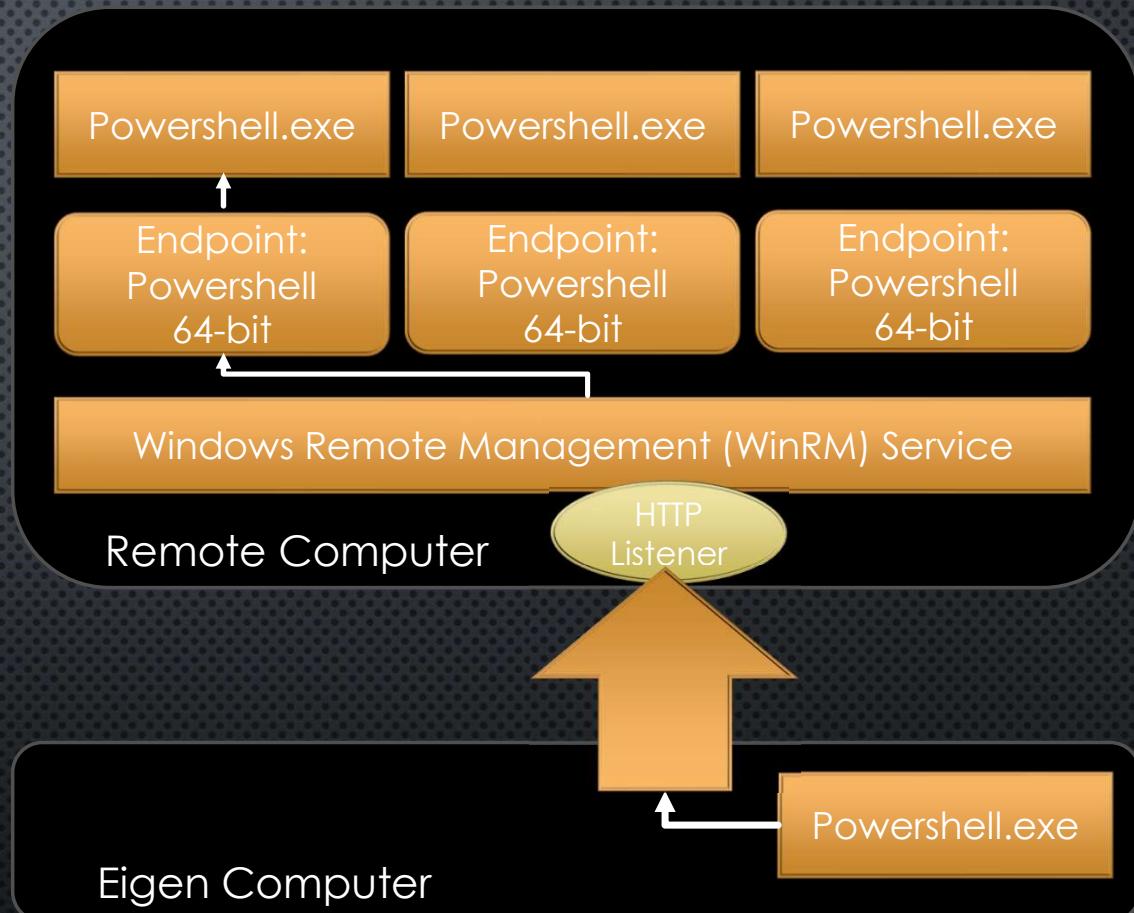
- WELKE WMI CLASS KAN GEBRUIKT WORDEN OM EEN IP ADRES VAN EEN NETWORK ADAPTER UIT TE LEZEN? HEEFT DEZE CLASS EEN METHODE OM DE DHCP LEASE VRIJ TE GEVEN?
- QUERY EEN LIJST VAN HOTFIXES MET WMI. HINT: MICROSOFT REFEREERT HIERNA ALS: QUICK FIX ENGINEERING. IS ER EEN VERSCHIL MET `Get-HotFix`?
- MAAK EEN TABEL DIE DE VOLGENDE INFORMATIE LAAT ZIEN: COMPUTER NAAM, OPERATING SYSTEM DESCRIPTION (CAPTION), OS BUILD NUMBER EN BIOS SERIAL NUMMER

REMOTING

- WINDOWS MANAGEMENT INSTRUMENTATION (WMI)
 - GEBRUIKT RPC's VOOR COMMUNICATIE
 - PRIMAIR VOOR HET ONTVANGEN VAN MANAGEMENT INFO
- DE -COMPUTER PARAMETER
 - bv GET-SERVICE AND GET-PROCESS
 - GEBRUIKT ONDERLIGGENDE TECHNOLOGIE VOOR COMMUNICATIE
 - ALLEEN BESCHIKBAAR OP EEN PAAR CMDLETS
- POWERSHELL REMOTING
 - GENERIEK, GOED VOOR ALLE COMANDO'S
 - GEBRUIKT WINRM /WS-MAN VOOR COMMUNICATIE
 - DE NIEUWE STANDAARD

REMOTING

- WS-MAN
- WINRM
- ENTER/EXIT-PSESSION
- INVOKE-COMMAND
- LOKALE VS. REMOTE COMMANDO'S



REMOTING WEETJES

- STAAT STANDAARD AAN OP SERVER 2008R2 EN 2012. NIET OP CLIENTS.
- AANZETTEN OP DE CLIENT:
 - ENABLE-PSREMOTING
 - GPO
- DEFAULT AUTHENTICATIE IS MET KERBEROS
- VERBINDINGEN WORDEN GELOGGED ALS EEN NETWORK LOGIN
- POWERSHELL PROFIELEN WERKEN NIET MET REMOTING
- EXECUTION POLICY VAN HET REMOTE SYSTEEM GELDT
- ALLEEN RECHTEN ALS JE LOCAL ADMIN OP HET REMOTE SYSTEEM BENT
- MINDER RESOURCE GEBRUIK OP HET SYSTEEM ALS MET EEN REMOTE DESKTOP
- OUTPUT KOMT TERUG ALS XML OBJECT

VOORBEELDEN VAN REMOTING

- 1 TO 1 REMOTING
- 1 TO N REMOTING
- WERKEN MET DE RESULTATEN
- INVOKE-COMMAND
- WERKEN MET SESSIONS
- DISCONNECTED SESSIONS

opdrachten

- MAAK EEN 1OP1 CONNECTIE MET EEN ‘REMOTE’ MACHINE EN START NOTEPAD. WAT GEBEURD ER? (EVT. MET LOCALHOST..)
- GEBRUIK `Invoke-Command` OM DE NIET DRAAIENDE SERVICES VAN 2 ‘REMOTE’ SYSTEMEN UIT TE LEZEN.
- GEBRUIKE `Invoke-Command` OM DE TOP 10 VIRTUEEL GEHEUGEN (VM) PROCESSEN UIT TE LEZEN VAN 2 ‘REMOTE’ SYSTEMEN.
- CREËER EEN TEKST FILE MET 3 ‘REMOTE’ SYSTEMEN EN GEBRUIK `Invoke-Command` OM VAN DEZE SYSTEMEN DE 100 NIEUWSTE APPLICATION EVENTS UIT TE LEZEN.

BACKGROUND JOBS

- POWERSHELL IS SINGLE THREAD (SYNCHROON)
- MAAR WE WILLEN MULTITASKEN (ASYNCHROON)
- MET JOBS CREËREN WE AANSPREEKBARE BACKGROUND THREADS
 - STATUS
 - RESULTAAT

WAT NOG MEER?

- MAKEN VAN MEERDERE THREADS
- BESPELEN VAN JOBS
- SCHEDULED JOBS

BACKGROUND JOBS

SYNCHROON

- ❖ Je kan reageren op vragen van commando
- ❖ Geeft direct output
- ❖ Missende parameters kunnen gevraagd worden

ASYNCHROON

- ❖ Loopt fout als er input nodig is
- ❖ Output moet opgevangen worden
- ❖ Loopt fout op missende parameters

GET-COMMAND -NOUN JOB

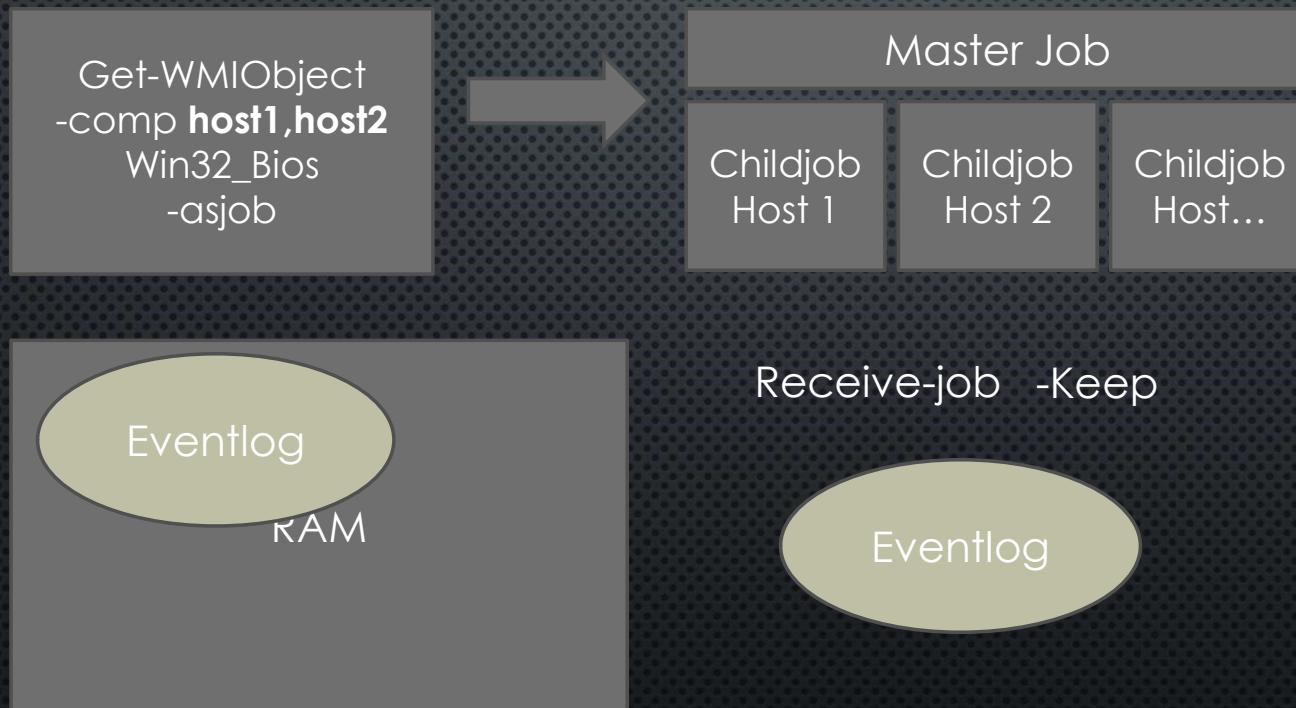
- GET-JOB
- RECEIVE-JOB
- REMOVE-JOB
- RESUME-JOB
- START-JOB
- STOP-JOB
- SUSPEND-JOB
- WAIT-JOB



A screenshot of a Windows PowerShell window titled "Windows PowerShell". The command "Get-Command -noun job" is run, and the output shows a table with two columns: " CommandType" and " Name". The " CommandType" column lists "Cmdlet" eight times, and the " Name" column lists "Get-Job", "Receive-Job", "Remove-Job", "Resume-Job", "Start-Job", "Stop-Job", "Suspend-Job", and "Wait-Job".

CommandType	Name
Cmdlet	Get-Job
Cmdlet	Receive-Job
Cmdlet	Remove-Job
Cmdlet	Resume-Job
Cmdlet	Start-Job
Cmdlet	Stop-Job
Cmdlet	Suspend-Job
Cmdlet	Wait-Job

MASTER AND CHILD JOBS



DE JOB LEVENSCYCLUS & SCHEDULED TASKS

- POWERSHELL VRIENDELIJKE TASK SCHEDULER
- WANNEER NEW-JOBTRIGGER
- OPTIES NEW-SCHEDULEDJOBOPTION
- AANMAKEN REGISTER-SCHEDULEDJOB



opdrachten

- CREEËER EEN BACKGROUND JOB DIE ALLE POWERSHELL SCRIPTS OP DE C: SCHIJF VIND.
- HOE VOER JE HET VORIGE COMMANDO UIT OP MEERDERE SERVERS?
- VOER DIT UIT OP 2 SYSTEMEN EN STOP DE OUTPUT VAN DE 2^E SERVER IN EEN VARIABELE
- CREEËER EEN BACKGROUND JOB DIE ELKE DAG OM 6 UUR 'S OCHTENDS DRAAIT EN DE LAATSTE 25 SYSTEM ERRORS UITLEEST EN WEGSCHRIJFT NAAR EEN XML FILE.

PASSWORD

JEA : Just enough Administration

- CLEAR TEKST !!!!
- GET-CREDENTIAL
- HASH YOU'RE PASSWORD
- HASH YOU'RE PASSWORD WITH AES KEY
 - LET'S ENRCYPT YOU'RE AES KEY ASWELL
- PASSWORD DATABASE (KEEPASS) AND POWERSHELL
- PASSWORD DATABASE HASHICORP VAULT

