# Windows PowerShell 3.0: Step by step

Ed wilson

## Overzicht hoofdstukken

Hoofdstuk 1: Overview of Windows PowerShell 3.0

Hoofdstuk 2: Using Windows PowerShell Cmdlets

Hoofdstuk 3: Understanding and Using PowerShell Providers

Hoofdstuk 4: Using PowerShell Remoting and Jobs

Hoofdstuk 5: Using PowerShell Scripts

Hoofdstuk 6: Working with Functions

Hoofdstuk 7: Creating Advanced functions and Modules

Hoofdstuk 8: Using the Windows PowerShell ISE

Hoofdstuk 9: Working with Windows PowerShell Profiles

Hoofdstuk 10: Using WMI

Hoofdstuk 11: Quering WMI

Hoofdstuk 12: Remoting WMI

Hoofdstuk 13: Calling WMI Methods on WMI Classes

Hoofdstuk 14: Using the CIM Cmdlets

Hoofdstuk 15: Working with Active Directory => Ga verder op Jasper’s

Hoofdstuk 16: Working with the AD DS Module

Hoofdstuk 17: Deploying Active Directory with Windows Server 2012

Hoofdstuk 18: Debugging Scripts

Hoofdstuk 19: Handling Errors

Hoofdstuk 20: Managing Exchange Server

### Inleiding:

Dit boek is geschreven door Ed Wilson. Het is de bedoeling van dit boek dat je een goede kennis opdoet van Windows PowerShell 3.0 technologie. Als je nieuw bent voor Windows Powershell, dan wordt er van je verwacht dat je je voornamelijk concentreert op de hoofdstukken 1 tot en met 3 en 5 tot en met 9. Aangezien wij beginnen vanaf nul, gaan wij deze volgorde nemen. Wij zullen de eerste 9 hoofdstukken, met uitzondering van hoofdstuk 4, doornemen en goed bijhouden wat belangrijk is voor dat hoofdstuk. Op deze manier krijgen we een bondige samenvatting voor beginnend powershellgebruik.

### Hoofdstuk 1: “Overview of Windows PowerShell 3.0”

Windows Powershell kan als vervanging gebruikt worden voor de CMD (command) shell. Zo kan je bijvoorbeeld gebruik maken van *cd* en *dir* in Windows PowerShell om te navigeren door je directorytree van je systeem. Belangrijk is dat je de structuur van cmdlets kent. Cmdlets, eigenlijk een andere naam voor commando’s, bestaan telkens uit een werkwoord gevolgd door een streepje en een zelfstandig naamwoord (bv: Get-Help,Set-Service). Get: Geeft informatie , Set: Veranderd informatie.

Het is mogelijk om meerdere commando’s tegelijk te gebruiken. Bv:

> ipconfig /all >tshoot.txt; route print >>tshoot.txt

Dit commando zal de uitvoer van ipconfig en route print in de tshoot.txt-file zetten.

Een van de belangrijkste updates van powershell, is het gebruik van –whatif en –confirm. Dit zijn argumenten die op het einde van een commando kunnen ingevoegd worden, zodanig dat vooraleer je een commando uitvoert, je eerst de mogelijk uitvoer krijgt. Op deze manier kom je niet voor verrassingen te staan, en zal je op voorhand weten wat het uitvoerende commando zal doen. Bij confirm zal je een confirmatie-vraag krijgen . vb:

1. Open Notepad door in de powershell notepad te typen.

2. Kijk wat het procesID is door gebruik te maken van het commando Get-Process note\*

Gebruik maken van een \* is een groot voordeel in powershell. Dit kan je beschouwen als een wildcard. Stel dat je een lijst wilt van alle get-commando’s die beginnen met een H, dan kan je get-H\* typen, en krijg je als resultaat alle mogelijk get commando’s die beginnen met een H.

3. Gebruik het –confirm/-whatif argument bij het stoppen van het proces van notepad:

> Stop-Process –id 1768 –confirm

Het processID van notepad is in ons geval 1768

4. Als je dit commando uitvoer zal je de vraag krijgen of je zeker bent dat je deze actie wilt uitvoeren. Dan moet de gebruiker vervolgens bevestigen of annuleren.

#### De helpfunctie

Vooraleer je start met het gebruik van windows powershell is het belan0grijk dat je het volgende commando uitvoert:

> Update-Help \_module \* -Force

Dit zorgt er voor dat je de laatste nieuwe versie hebt van de help-functie. De helpfunctie word enorm veel gebruikt. Daarom is het van belang dat je de laatste nieuwe info hebt.

Ik zou je nu kunnen uitleggen hoe de help-functie werkt, maar al doende leert men beter. Daarom type je het commando: > Get-Help Get-Help. Dit zal je een uitvoerige beschrijving geven over hoe de Get-help functie werkt. Dat is hoe de help-functie werkt. Je typt het commando Get-Help gevolgt door het commando waarover je hulp wilt krijgen. Stel dat je niet meer weet hoe het commando precies gaat, maar je weet dat het commando begint met een ‘p’, dan geef je het commando Get-Help get-p\*. Stel dat je wel weet hoe het commando geschreven wordt, maar je weet niet meer hoe de syntax daarvan uitziet, dan kan je gebruik maken van Get-Help met het –examples argument. (bv Get-Help Get-PSDrive –examples)

De helpfunctie heeft drie verschillende weergaven, een normale, een gedetailleerde en een volledige weergave. Dit zijn argumenten die op het einde van het commando moeten bijgevoegd worden. Om het verschil te weten tussen deze 3 raad ik u aan deze commando’s uit te proberen en te vergelijk.

> Get-Help Get-Help –detailed

> Get-Help Get-Help –full

> Get-Help Get-Help

### Hoofdstuk 2: “Using Windows PowerShell Cmdlets”

Commando’s in powershell zijn niet hoofdletter gevoelig. Dit wil zeggen dat bijvoorbeeld Get-Childitem dezelfde uitvoer heeft als get-Childitem of get-childitem.

De tabtoets is een heel handige toets in het gebruik van PowerShell. Deze toets bied je de mogelijkheid om je commando automatisch te laten aanvullen. Zo kan je get-ch typen en vervolgs de tab-toets gebruiken om zo Get-ChildItem te krijgen. Stel dat er meerdere mogelijkheden zijn, dan kan je met de tab-toets al deze mogelijkheden overlopen.

Samen met de tabtoets, zijn de aliases een handige tool voor het gebruik van PowerShell commando’s. Een alias maakt het mogelijk om een commando in een verkorte versie te schrijven. Zo is het commando dir een verkorte versie van het commando Get-ChildItem. Een lijst van alle mogelijke aliases kan je krijgen met het commando Get-Alias sort. Het is ook mogelijk om zelf een alias te maken. Dit kan via het commando New-Alias VerkorteVersie VolledigCommando. (bv New-Alias gh Get-Help)

Dit hoofdstuk gaat vooral over het gebruik van commando’s. Daarom verwijs ik je graag door naar de cheatsheet van deze samenvatting. *(Windows PowerShell 3.0 Step by Step cheatsheet.xlsx)*

### Hoofdstuk 3:”Understanding and Using PowerShell Providers”

#### Wat zijn providers?

Belangrijk is dat je begrijp wat providers zijn. Microsoft Windows Powershell heeft constant een externe manier om aan informatie te geraken. Dit door gebruik te maken van providers. Dit zijn eigenlijk .NET programma’s die “lelijke” details verbergen voor de gebruiker, om zo snel en gemakkelijk aan informatie te geraken. Om een lijst te krijgen van alle providers, kunnen we het commando Get-PSProvider gebruiker. Zo zie je bijvoorbeeld dat er een provider is genaamd Alias.

Mogelijke providers zijn:

- Alias: Deze alias provider zorgt ervoor dat je makkelijk te gebruiken toegang hebt tot alle aliases in PowerShell. Om op je machine te kunnen werken met aliases gebruik je Set-Location en specifieer je die met Alias:\ drive.

- Certificate: geeft u de mogelijkheid te zoeken, kopiëren, verplaatsen , verwijderen van certificaten.

- Omgeving: De omgeving -provider in Windows PowerShell wordt gebruikt om toegang tot de systeem omgevingsvariabelen te voorzien . Als je een CMD ( commando) shell opent, en vervolgs typ je set, dan ziet u een lijst van alle te verkrijgen omgevingsvariabelen gedefinieerd op het systeem.

- Bestandsysteem: Deze provider is de gemakkelijkste om te begrijpen. De provider bied toegang tot het bestandsysteem van de computer. Wanneer Windows PowerShell gestart wordt, opent het automatisch in de gebruikers documentenfolder. Met deze provider kan je bestanden en documenten creëren, verwijderen, wijzigen,…

- Functie: deze provider bied u toegang tot de functies die gedefinieerd zijn binnen PowerShell. Hiermee kan je een lijst krijgen van alle functies op je systeem. Je kan ook functies toevoegen, verwijderen en wijzigen. Deze provider is gebaseerd op een bestandsysteem model.

- variabele: Net zoals bovenstaande providers, biedt deze provider toegang tot de variabele binnen Windows PowerShell. Zowel variabele gedefinieerd door gebruikers, als door het systeem. Je kan een lijst krijgen van commando’s die speciaal gemaakt zijn om te werken met variabelen. Dit gaat via de Get-Help \*variabele. Stel dat je enkel commando’s terug wilt dan moet je | Where-Object category –eq “cmdlet” hieraan toevoegen.

### Hoofdstuk 4: “Using PowerShell Remoting and Jobs”

Remoting is een van de grootste verbeteringen in PowerShell 3.0