



PowerShell 4.0 & 5.0



Haiko Hertes
Dipl.-Inf. (FH), M.Sc.
ppedv AG
Senior Trainer (MCT)

Technical Summit 2015
The Next Level

@HHertes

#msts15

Wer spricht da?

- Haiko Hertes, Baujahr 1986
- Informatik-Studium in Leipzig (Dipl.-Inf. (FH), M.Sc.)
- Dabei: T-Systems North America, USA
- Bisher: Diverse MS Gold Partner, Datacenter Administration
- Aktuell: Senior IT Trainer bei ppedv AG, Regional Lead



Microsoft
CERTIFIED
Trainer

Microsoft
CERTIFIED
IT Professional

Microsoft
CERTIFIED
Technology Specialist

Microsoft
CERTIFIED
Solutions Associate

Microsoft
CERTIFIED
Solutions Expert



Was ist neu?

PowerShell 4.0



Get-Help about_Windows_PowerShell_4.0

- Desired State Configuration (DSC)
- Viele neue Cmdlets (u.a. „Get-FileHash“, „Get-ChildItem“)
- Neue Parameter (z.B. „-RunNow“ für Scheduled Jobs)
- Remote-Debugging von Skripten
- Hilfe kann mittels Save-Help für Offline-Systeme gespeichert werden
- Aufruf von Methoden über Variablen

Desired State Configuration

- Deklarative Skripte
- Beschreibt Soll-Konfiguration eines oder mehrere Systeme
- Kann entweder nur prüfen (Test-DSCConfiguration) oder auch umsetzen (Start-DSCConfiguration)
- Mehrere Systeme parallel

Desired State Configuration

- Kleines Beispiel:

```
1 Configuration ContosoWebsite
2 {
3     param ($MachineName)
4
5     Node $MachineName # Wer bekommt diese Config?
6     {
7         #Installation des IIS Webservers
8         WindowsFeature IIS
9         {
10            Ensure = "Present"
11            Name = "Web-Server"
12        }
13
14        #Installation von ASP.NET 4.5
15        WindowsFeature ASP
16        {
17            Ensure = "Present"
18            Name = "Web-Asp-Net45"
19        }
20    }
21 }
```

Name der DSC

Nodes = Zielserver

Erstes Konfigurations-Element

Zweites Konfigurations-Element

Desired State Configuration

File – Datei Operationen, kopiert Files und Ordner oder löscht sie

Archive – Enpackt Zip Files

Environment – Setzt/löscht Umgebungsvariablen

Group – Erstellt/löscht Lokale Gruppen

Log – Schreibt Konfigurationsmeldungen ins Log

Package – Paketdateien (wie MSI oder EXE) können ausgeführt werden

Registry – setzt/löscht RegKeys

Script – Führt Powershell Skriptblöcke aus

Service – Verwaltet Dienste

User – Erstellt/löscht Lokale User

WindowsFeature – konfiguriert Windows Features

WindowsProcess – Konfiguriert Prozesse

Ab PowerShell 5.0 zusätzlich:

WaitForAll

WaitForAny

WaitForSome

WindowsOptionalFeature

Erweiterbar durch Ressource Kit:

<https://gallery.technet.microsoft.com/scriptcenter/DSC-Resource-Kit-All-c449312d>

Desired State Configuration

The DSC Resource Kit consists of the following modules, which in turn contain the nested resources:

[xActiveDirectory](#): Allows configuration of Active Directory components.

[xAdcsDeployment](#): is used to configure Certificate Services after the role is enabled on a Windows Server.

[xAzure](#) : Provides a method of depoying virtual machines in Microsoft Azure.

[xAzurePack](#) : Contains resources for installation and configuration of Windows Azure Pack.

[xBitlocker](#) : Allows you to configure Bitlocker.

[xChrome](#) : Installs the Chrome browser

[xComputerManagement](#) : Allows configuration of a nodes's computer name and workgroup/domain

[xCredSSP](#) : xCredSSP resource, which enables, disables, and configures Credential Security Support Provider (CredSSP).

[xDatabase](#) : Allows configuration of databases

[xDHCPServer](#): Enables configuration of a DHCP Server.

[xDisk](#): Enables management of a disk drive

[xDismFeature](#): Enables or disables Windows optional features.

[xDnsServer](#): Enables configuration of a DNS Server

[xDscDiagnostics](#): Allows simple diagnostics of DSC Logs.

[xDscResourceDesigner](#): Allows easy creation and testing of DSC Resources

[xExchange](#): allows you to configure many different properties of Exchange 2013 servers, including individual server properties, databases and mount points, and Database Availability Groups.

[xFailOverCluster](#): Allows configuration of a Failover Cluster.

[xFirefox](#): Allows installation of the latest Firefox browser.

[xHyper-V](#): Allows configuration of the Hyper-V host.

[xInternetExplorerHomePage](#): This resource enables you to set one or more IE homepages using PowerShell DSC.

[xJEA](#): Defines and configures constrained PowerShell Endpoints to enable Just Enough Admin scenarios

[xMySQL](#) : allows you to setup a mysql server, database, user, and create a grant for the user. This resource requires WMF 5.0 Experimental July 2014.

[xNetworking](#): Allows configuration of a node's IP Address, DNS Address, and Firewall Rules.

[xOU](#): Allows creation and configuration of an Active Directory Organizational Unit (OU)

[xPendingReboot](#): Checks to see if a reboot is pending and allows DSC to predictably handle the condition

[xPhp](#): allows you to setup php in iis.

[xPowerShellExecutionPolicy](#): allows you to setup php in iis.

[xPSDesiredStateConfiguration](#): Contains improvements to built-in DSC Resources, as well as a resource for configuring a DSC "Pull Server."

[xReleaseManagement](#): Resources created to work with the Release Management features of Visual Studio

[xRemoteDesktopAdmin](#): Allows configuration of remote desktop and the related Windows firewall settings

[xRemoteDesktopSessionHost](#): Allows configuration of RDSH components.

[xSafeHarbor](#): This module contains the configurations that allow you to setup the SafeHarbor example. No new resources are included.

[xSCDPM](#) : Contains resources for installation of System Center Service Management Automation (DPM).

[xSCOM](#): Contains resources for installation of System Center Operations Manager (OM).

[xSCSMA](#) : Contains resources for resources for installation of System Center Service Management Automation (SMA).

[xSCSPF](#): Contains resources for resources for installation of System Center Service Provider Foundation (SPF).

[xSCSR](#): Contains resources for installation of System Center Service Reporting (SR).

[xSCVMM](#): Contains resources for installation of System Center Virtual Machine Manager (VMM).

[xSmbShare](#): Allow's configuration of a SMB Share.

[xSqlPs](#): Allows configuration of SQL Server

[xSqlServer](#): Resources for System Center-compliant setup of SQL Server

[xSystemSecurity](#): Allows configuration of User Account Control prompts and IE Enhanced Security Configuration.

[xTimeZone](#): Allows setting the system time zone in Windows using PowerShell DSC

[xWebAdministration](#): Allows configuration of IIS.

[xWindowsupdate](#) : Handles installation of a windows update or hotfix.

[xWindowsRestore](#) : Use PowerShell DSC to configure system restore, create or remove a restore point.

[xWinEventLog](#): Allows configuration of the Windows Event Logs.

[xWordPress](#): Resources and sample configurations to show end-to-end deployment of the common wordpress site. This resource requires WMF 5.0 Experimental July 2014.

[xFileShare](#): Allow configuration of a file share and share permission rules.

Desired State Configuration

- Push und Pull möglich
- Local Configuration Manager läuft als Engine auf allen Target-Nodes
- Prüfung und ggf. Korrektur der gewünschten Konfiguration in regelmäßigen Abständen möglich

Desired State Configuration

```
4 Configuration MyLCMConfig
5 {
6     Param([string]$ComputerName)
7     node ($ComputerName)
8     {
9         LocalConfigurationManager
10        {
11            ConfigurationMode = "ApplyAndAutoCorrect"          # Autokorrektur
12            ConfigurationModeFrequencyMins = 15                # nur überwachen: "ApplyAndMonitor"
13            RebootNodeIfNeeded = $True                         # Wie oft soll die Configuration angewendet werden?
14            RefreshMode = "Pull"                            # Kleiner geht nicht, 30min ist default
15            RefreshFrequencyMins = 30                      # "Push" ist default
16                                            # Download der Config vom Server alle x min
17                                            # 30min ist default, nur für Pull-Mode
18        }
19    }
20}
21}
22}
```

Demo: Desired State Config.

0 1 1 1
0 1 0 1 0 1
1 0 0 0 1 1
0 0 0 1 1 1
0 0 1 1 0 0 0
1 0 1 1 0 0 0 1 0
0 1 0 0 0 1 1 1 0
1 0 1 0 0 1 0 0 1 0
0 0 0 1 0 1 0 1 0 0
0 0 1 0 1 1 0 1 0 0 0
1 0 1 1 0 1 0 1 0 0 0
0 0 0 1 1 0 1 0 1 0 0 0
0 0 1 0 0 1 0 1 0 1 0 1
1 0 1 0 0 1 1 0 1 0 1 0 1
0 0 0 1 0 0 0 1 0 1 0 1 0 0
0 0 1 0 0 1 1 0 1 0 1 0 0 0
1 0 1 0 0 1 0 1 1 0 1 0 0 0
0 0 0 1 0 0 0 1 1 0 1 0 0 0
0 0 1 0 0 1 0 1 0 1 1 0 1 0 0
1 0 1 0 0 1 0 0 1 1 0 1 0 0 0
0 0 0 1 0 0 0 1 0 1 1 0 1 0 0 0
0 0 1 0 0 1 0 0 1 1 0 1 0 0 0
1 0 1 0 0 1 0 0 0 1 1 0 1 0 0 0
0 0 0 1 0 0 0 1 0 0 1 1 0 1 0 0 0
0 0 1 0 0 1 0 0 0 1 1 0 1 0 0 0
1 0 1 0 0 1 0 0 0 0 1 1 0 1 0 0 0
0 0 0 1 0 0 0 0 1 1 0 1 0 0 0
0 0 1 0 0 0 0 1 1 0 1 0 0 0
1 0 1 0 0 0 0 0 1 1 0 1 0 0 0
0 0 0 1 0 0 0 0 0 1 1 0 1 0 0 0



Was ist neu?

PowerShell 5.0



Get-Help about_Windows_PowerShell_5.0

- OneGet-Framework implementiert (später umbenannt)
- Unit-Tests mit Open-Source-Projekt „Pester“
- Neue Cmdlets (z.B. Compress-Archive, Expand-Archive, ConvertFrom-String, Set-Clipboard, Get-Clipboard, ...)
- Data Center Abstraction steuert Netzwerkhardware
- Definition eigener Klassen

Package Manager

- „PowerShell Package Manager“
- Integration des OpenSource-Projektes OneGet
- Package Manager für Windows-Systeme
- (eigentlich Package-Manager-Manager ☺)
- Ähnlich Apt-Get aus Linux-Welt

Package Manager

- Installation von Software aus Katalogen
- Bekannte Systeme wie NuGet und Chocolatey sind bereits nutzbar
- Bereits jetzt über 3200 Pakete (Stand: 10/2015)

PS C:\Users\Administrator> Get-Command -Module PackageManagement			
CommandType	Name	Version	Source
Cmdlet	Find-Package	1.0.0.0	PackageManagement
Cmdlet	Get-Package	1.0.0.0	PackageManagement
Cmdlet	Get-PackageProvider	1.0.0.0	PackageManagement
Cmdlet	Get-PackageSource	1.0.0.0	PackageManagement
Cmdlet	Install-Package	1.0.0.0	PackageManagement
Cmdlet	Register-PackageSource	1.0.0.0	PackageManagement
Cmdlet	Save-Package	1.0.0.0	PackageManagement
Cmdlet	Set-PackageSource	1.0.0.0	PackageManagement
Cmdlet	Uninstall-Package	1.0.0.0	PackageManagement
Cmdlet	Unregister-PackageSource	1.0.0.0	PackageManagement

Demo: Package Manager

Scripting mit der PowerShell



Scripting

Man nehme: Ein paar einfache Aufrufe...

```
1 $CPU = Get-WmiObject Win32_Processor -Computername localhost  
2 $RAM = Get-WmiObject Win32_OperatingSystem -Computername localhost  
3 $Disk = Get-WmiObject Win32_LogicalDisk -Computername localhost | Where-Object DeviceID -Like C*
```

... und führe Parameter ein.

Scripting

Ein Parameter „TargetComp“ ersetzt „localhost“...

```
1 Param(  
2     [string]$TargetComp  
3 )  
4  
5 $CPU = Get-WmiObject Win32_Processor -Computername $TargetComp  
6  
7 $RAM = Get-WmiObject Win32_OperatingSystem -Computername $TargetComp  
8  
9 $Disk = Get-WmiObject Win32_LogicalDisk -Computername $TargetComp | Where-Object DeviceID -Like C*  
10 |
```

Scripting

Parameter ist Pflicht, es dürfen auch mehrere Systeme übergeben werden:

```
1 Param(
2     [Parameter(Mandatory=$True)]
3     [string[]]$TargetComp
4 )
5
6 $CPU = Get-WmiObject Win32_Processor -Computername $TargetComp
7
8 $RAM = Get-WmiObject Win32_OperatingSystem -Computername $TargetComp
9
10 $Disk = Get-WmiObject Win32_LogicalDisk -Computername $TargetComp | Where-Object DeviceID -Like C*
```

Scripting

```
1 Param(  
2     #[Parameter(Mandatory=$True)]  
3     [string[]]$TargetComp = "localhost"  
4 )  
5  
6 $Proc = Get-WmiObject Win32_Processor -Computername $TargetComp  
7  
8 $RAM = Get-WmiObject Win32_OperatingSystem -Computername $TargetComp  
9  
10 $Disk = Get-WmiObject Win32_LogicalDisk -Computername $TargetComp # | where-Object DeviceID -Like C*  
11  
12 Write-Host "Prozessor:" -ForegroundColor Yellow  
13 $Proc | Format-Table @{n="Computer";e={$_.SystemName}},  
14                         @{n="Anzahl Kerne";e={$_.NumberOfLogicalProcessors}},  
15                         @{n="CPU(%)" ;e={$_.LoadPercentage}}  
16  
17 Write-Host "Arbeitsspeicher:" -ForegroundColor Yellow  
18 $RAM | Format-Table @{n="Computer";e={$_.PSComputerName}},  
19                         @{n="RAM ges. (GB)";e={[math]::Round($_.TotalVisibleMemorySize / 1MB,0)}},  
20                         @{n="RAM frei(GB)";e={[math]::Round($_.FreePhysicalMemory / 1MB,2)}}  
21  
22 Write-Host "Festplatte:" -ForegroundColor Yellow  
23 $Disk | Format-Table @{n="Computer";e={$_.PSComputerName}},  
24                         DeviceID,  
25                         @{n="Free(GB)";e={[math]::Round($_.FreeSpace / 1GB,2)}},  
26                         @{n="Free(%)" ;e={[math]::Round($_.FreeSpace / $_.Size * 100,1)}}}
```

Scripting

Jetzt noch alles in eine Funktion verpacken:

```
1 Function Get-HardwareInfo
2 {
3     Param(
4         #[Parameter(Mandatory=$True)]
5         [string[]]$TargetComp = "localhost"
6     )
7
8     $Proc = Get-WmiObject Win32_Processor -Computername $TargetComp
9
10    $RAM = Get-WmiObject Win32_OperatingSystem -Computername $TargetComp
11
12    $Disk = Get-WmiObject Win32_LogicalDisk -Computername $TargetComp # | Where-Object DeviceID -Like C*
13
14    Invoke-Command {...} # Hier ist die Ausgabe "versteckt"
15
16 }
17
18 Get-HardwareInfo # -TargetComp "SRV1","SRV2"
```

Scripting

Jetzt kann die Funktion genutzt werden (vorher 1x ausführen) :

```
PS C:\Users\Administrator> Get-HardwareInfo -TargetComp "SRV1", "SRV3"

Prozessor:

Computer Anzahl Kerne CPU(%)
-----
SRV1          4      0
SRV3          2      2

Arbeitsspeicher:

Computer RAM ges. (GB) RAM frei(GB)
-----
SRV1          8      5,27
SRV3          1      0,22

Festplatte:

Computer DeviceID Free(GB) Free(%)
-----
SRV1          C:    27,36   46
SRV1          D:     0        0
SRV3          C:    49,47  83,2
```

Demo: PowerShell Scripting



Scripting

Das ist noch lange nicht alles. Es fehlt bspw. noch:

- Fehlerbehandlung (z.B. Falscher Hostname, Host nicht erreichbar, ...)
- Hilfetexte
- Rückgabe in Objektform
- Weitere Parameter
- ...

Kontakt

 haikoh@ppedv.de

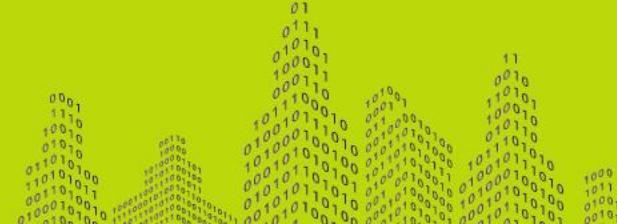
 <http://www.hertes.net>

 <https://about.me/haiko.hertes> (facebook, twitter, ...)

 <http://ppedv.de/msts/>



Technical Summit 2015
The Next Level



#msts15 

Vielen Dank

© 2014 Microsoft Corporation. All rights reserved. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.



Weiterführende Informationen

Entwickler:

www.techwiese.de - News, Ressourcen, Events und Support für Entwickler

www.msdn.de/newsletter - MSDN Flash – kostenloser Newsletter für Entwickler

IT Pros:

www.itprohub.de - News, Ressourcen, Events und Support für IT Profis

www.technet.de/flash - TechNet Flash - kostenloser Newsletter für IT Profis

Für Devs und IT Pros:

www.mva.ms - Kostenlose Online-Schulungen für Entwickler und IT Profis

www.ch9.ms - Videoplattform für Entwickler und IT Profis