

at Microsoft Virtual Academy

Windows PowerShell 3: Skripterstellung mit PowerShell

Teil 1

Frank Koch

Mai 2012

Kommandozeile (Shell) und Skriptsprache



```
C:\Windows\System32\WindowsPowerShell\v1.0\PowerShell.exe
PS C:\> Get-PSProvider

Name           Capabilities           Drives
-----
Alias           ShouldProcess          {Alias}
Environment    ShouldProcess          {Env}
FileSystem     Filter, ShouldProcess {C, D, WIN, Fav...}
Function       ShouldProcess          {Function}
Registry       ShouldProcess          {HKLM, HKCU}
Variable       ShouldProcess          {Variable}
Certificate    ShouldProcess          {cert}

PS C:\> Get-WmiObject Win32_ComputerSystem

Domain           : ntdev.corp.microsoft.com
Manufacturer     : Dell Inc.
Model           : Inspiron 9300
Name            : JPSVISTA1
PrimaryOwnerName : jsnover
TotalPhysicalMemory : 2146279424

PS C:\>
```

- Erhöhte Produktivität und Kontrolle
- Beschleunigte Automatisierung der Administration
- Leicht zu lernen und zu verwenden

PowerShell Grundlagen

- Interaktive Shell und Skriptsprache
 - Objektorientiert statt Text-Parsing
 - Basiert auf .NET
 - Kann auch WMI & COM Objekte Nutzen
 - Umfangreicher Befehlsumfang
 - 1:0 120+ 2.9: 200+ 3.0 2300+ Befehle
 - "Alte" Tools können weiterverwendet werden
- Administrationsgrundlage für alle weiteren Serverprodukte von Microsoft
 - Exchange, SQL, System Center...

Fragen

- Muss ich .NET Lernen bevor ich PowerShell nutzen kann?
 - Nein, Sie können bisherige Kenntnisse weiter nutzen!
- Muss ich alle meine Tools neuschreiben?
 - Nein, alle existierenden Tools funktionieren weiterhin!
- Muss ich eine neue Sprache lernen?
 - Nein, Sie können vorhandene Skriptkenntnisse weaternutzen!
- Warum Sie die neuen Möglichkeiten lernen sollten:
 - Die neue Sprache erleichtert extrem die Einbindung von .NET
 - .NET eröffnet eine unendliche Welt von Tools und Möglichkeiten
 - Lernen Sie in Ihrem eigenen Tempo, und nutzen Sie weiterhin bereits bekannte Tools und Kommandos
 - Online Hilfe ist voller Beispiele zum direkten Verwenden

Was heißt Objektorientiert?

- Die Kommandozeile kann direkt folgende "Objekte" verstehen und verarbeiten:

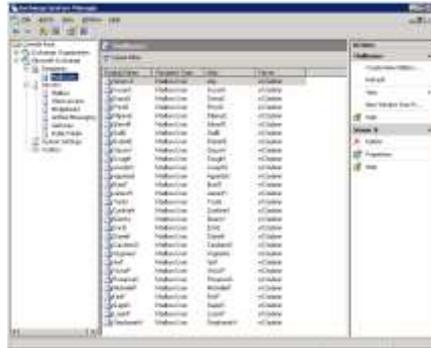
Eingabe	Ausgabe	Erklärung
„Hello World“	Hello World	Text wird als Textobjekt aufgefasst und direkt wieder ausgegeben
2 + 3	5	Zahlen werden direkt verarbeitet
2 + 3 + „4“	9	Fängt eine Zeile mit einer Zahl an, werden alle weiteren Objekte in Zahlen umgewandelt oder es gibt eine Fehlermeldung
„4“ + 2	42	Fängt eine Zeile mit Text an, werden alle übrigen Objekte als Text aufgefasst
„hello“ * 3	hellohellohello	Texte können addiert und multipliziert werden

Windows PowerShell ist mehr als ein Skripting-Tool

Microsoft Virtual Academy

MMC 3.0 als GUI zur PowerShell

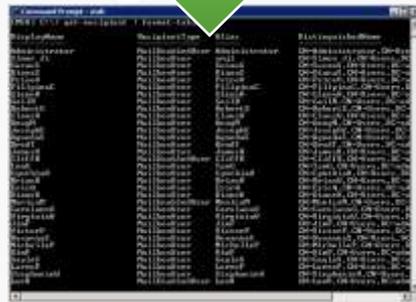
GUI



Layered Over



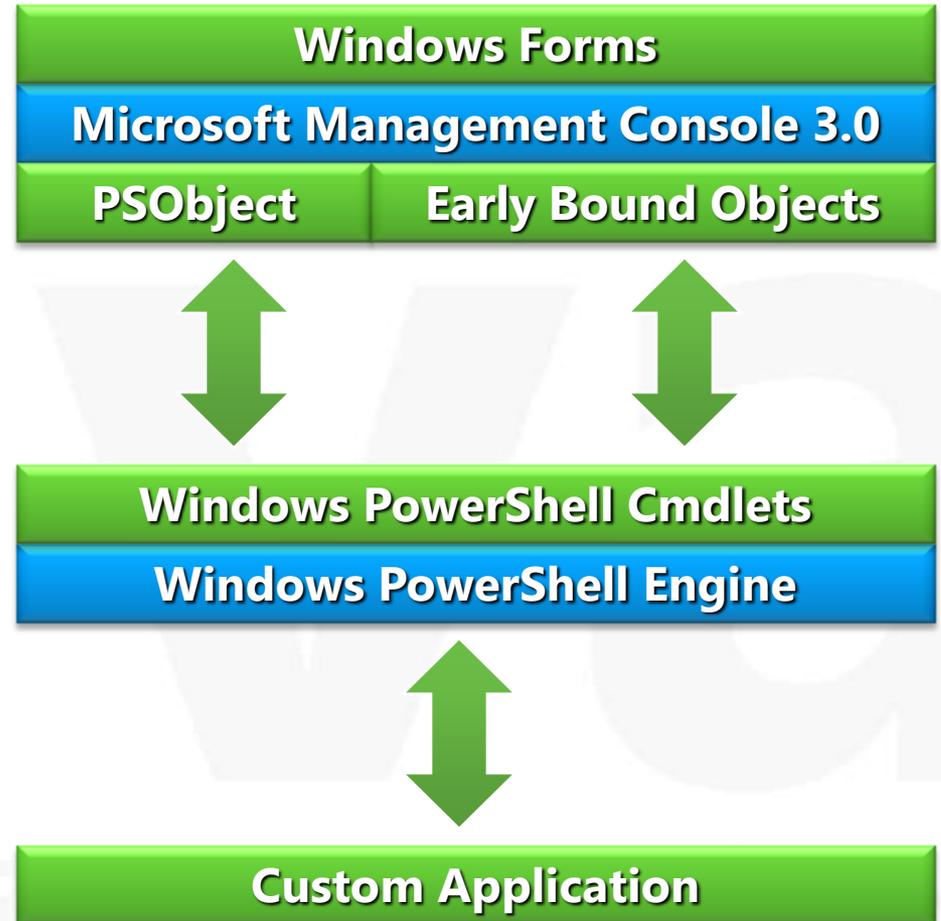
Command Line



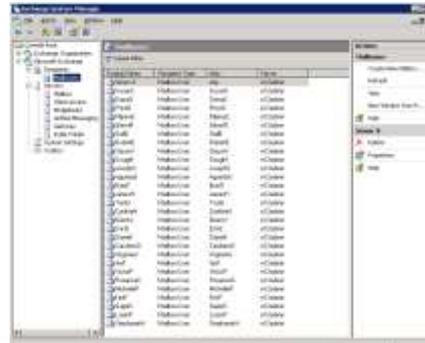
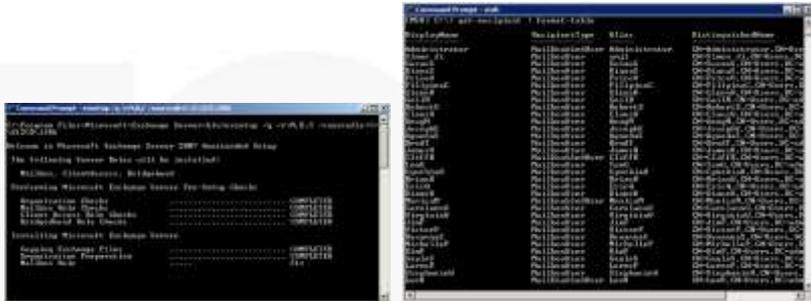
To Manage



Your Application



Beispiel: Exchange



WinForms

ADO.Net

WinForms

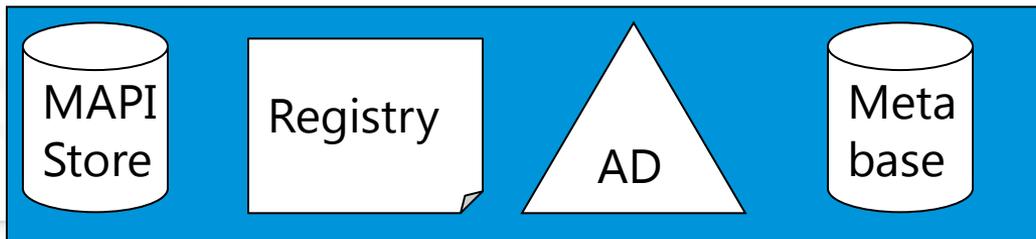
Early-bound objs

PowerShell Data Provider

PowerShell Engine

Exchange cmdlets

Configuration Data Access



Microsoft Produkte und PowerShell

- Exchange Server ab 2007
 - Fully build on PowerShell, fully scriptable
 - Add new users, mailboxes, mailstores...
 - GUI has only a subset of features
 - GUI "records" actions and present the PowerShell script at the end
- SC Virtual Machine Manager
 - Fully build on PowerShell, fully scriptable
 - Add new network cards, memory, processors...
- System Center Operations Manager
 - Server features scriptable via PowerShell

Microsoft Produkte und PowerShell

- SC Data Protection Manager ab 2007
 - Fully on PowerShell
- System Center Service Manager
 - PowerShell support when it ships
- Windows Server 2008
 - New Server Manager fully scriptable
- Windows Server Compute Cluster v2
- Part of Common Engineering Criteria
 - Version 2009+