



## Einführung in Windows PowerShell

Torsten Kroheck  
Staatlich geprüfter Informatiker  
<http://blog.suburbia.de>

**Microsoft**<sup>®</sup>  
**CERTIFIED**  
*Solution Developer*

**Microsoft**<sup>®</sup>  
**CERTIFIED**  
*Database Administrator*

# Agenda

- ▶ Einführung in Windows PowerShell
- ▶ Commandlets und Objekt-Pipelining.
- ▶ Nutzung von Aliasen
- ▶ Navigationsparadigma
- ▶ Sicherheitsfunktionen
- ▶ PowerShell und .NET
- ▶ COM-Objekte in PowerShell
  
- ▶ Skripte erstellen und konvertieren
  
- ▶ Tooldemo "Quest PowerGUI"
- ▶ PowerShell Projekte

Please note that all information is supplied "as is" and with no warranty



# Einführung in Windows PowerShell

- ▶ OT: Entwicklung in 17 Jahren...



- ▶ Quelle: <http://www.nerdcore.de/wp/2007/06/27/nevermind/>

# Einführung in Windows PowerShell

- ▶ Existierend: Command Prompt (cmd.exe)
- ▶ Herausforderung:
  - Interaktiv und Zusammensetzbar wie BASH/KSH
  - Programmierbar wie Perl/Ruby
  - Produktions-orientiert wie AS400CL / VMS DCL
- ▶ Vorbilder sind somit:
  - Ms-Dos Command Prompt (cmd.exe)
  - Unix Shells (Befehle awk / sed)
  - Perl / C#
  - .NET-Framework
  - Windows Scripting Host
  - Windows Management Instrumentation (WMI)

# Commandlets und Objekt-Pipelining.

- ▶ PS-Befehle heißen „Commandlets“ bzw. Cmdlets
- ▶ Commandlets bestehen aus 3 Teilen:
  - Verb (add, copy, format, get, new, out, set, ...)
  - Bindestrich (-)
  - Substantiv (Childitem, Command, Alias)
  - Parameterliste (-? , ...)
- ▶ PowerShell übergibt Objekte anstatt von Strings beim Pipelining
  - Also: `Get-xy | Where { xy } | Sort handles | Format-table`
  - Z.B.: `Get-Process | Where { $_.handles -gt 500 } | Sort handles | Format-table`

# Nutzung von Aliasen

- ▶ DIR ist Alias für Get-Childitems
- ▶ Get-Alias liefert die aktuelle Aliasliste
- ▶ Definition von Aliasen (temporär)
  - Geht nicht unter PowerShell: dir /q
  - Lösung: Löschen des Alias: Remove-Item Alias:dir
  - Erstellung eines neuen Alias: function dir {cmd /c dir \$args}  
Z.B.: dir /ad /q
- ▶ Definition im PS-Profile (permanent)
  - All Users:  
C:\WINDOWS\system32\windowspowershell\v1.0\profile.ps1
  - Single User: D:\Documents and Settings\%username%\My Documents\WindowsPowerShell\profile.ps1
  - Doppelte Aliase finden: gcm -type cmdlet,function,alias |group name |where {\$\_.count -gt 1}

# Navigationparadigma

- ▶ Navigation Provider: Datenspeicher werden generell als „Drive“ genommen
- ▶ „Drive“ ist ein Namensraum mit unterschiedlichen Informationsaxen (Item, ChildItem, Content, ItemProperty, ACL)
- ▶ Neu:
  - Cert: Zertifikatsspeicher
  - Env: Environment-Variablen
  - Function: Funktionen der PowerShell-Skriptsprache
  - HKCU / HKLM: Registry
  - Variable: Alle Variablen
- ▶ Get-PSDrive
- ▶ Definition eigener PS-Drives:
  - `New-PSDrive -name Devgroup -psprovider FileSystem -root c:\update`

# Sicherheitsfunktionen

## ▶ Sicherheitsfunktionen bei Commandlets:

- Whatif

- `Gps |where {$_.handles -ge 500} | stop-process -Whatif`

- Confirm

- `Stop-process -name Notepad -Confirm`

- Verbose

- `Stop-Process -name [a-x]*[q]*[r-t] -Verbose`

## ▶ Execution Policies (ändern durch Set-ExecutionPolicy)

- Restricted: keine Skripte (Standard)

- AllSigned: signierte Skripte, Skripte von nicht-vertrauten Quellen auf Anfrage

- Remote-Signed: Signatur nur für Skripte aus dem Internet (Outlook, Browser, Messenger)

- Unrestricted: Alle Skripte laufen

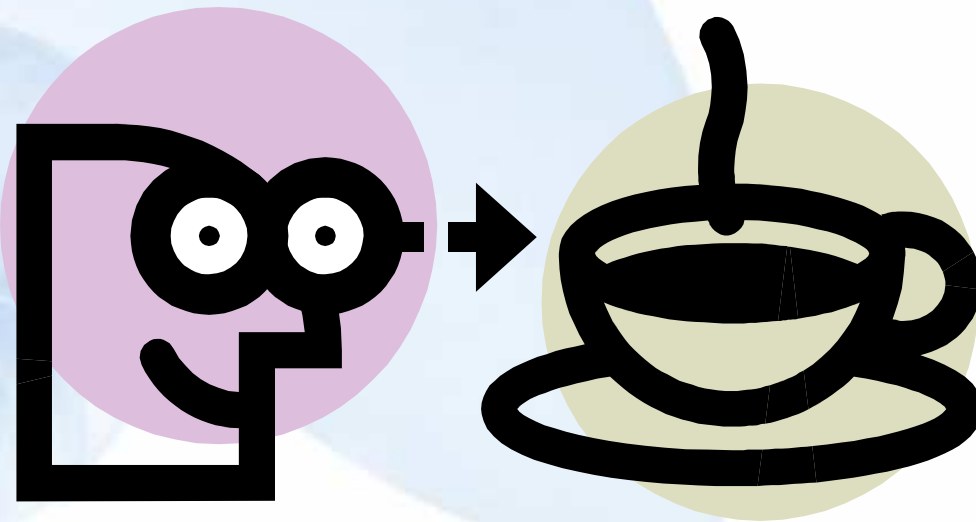
# PowerShell und .NET

- ▶ PowerShell basiert auf dem .NET 2.0 Framework
- ▶ Kritik: Keine neue Shell, sondern eine neue Skriptsprache
- ▶ Kritik: Durch „Secure-by-default“ ist es keine Alternative für Login-Scripts, da zu viele Stolperfallen.
- ▶ Gesamte .NET-BCL ist verfügbar
  - Aufruf Statischer Klassen via [Klasse]::Methode
    - Z.B. [System.Diagnostics.Eventlog]::WriteEntry(„PowerShell-Eintrag“, „PS-Eintrag erstellt“)
    - Get-EventLog Application -newest 10
  - Aufruf neuer Objekte mit New-Object
    - \$myVersion = new-object System.Version 1,2,3,4
  - Aufruf der PowerShell aus WindowsForms (Beispiel)

# Com-Objekte in PowerShell

- ▶ Der Befehl „New-Object“ mit Parameter „-ComObject“ erlaubt Zugriff auf COM.
  - `$ie = New-Object -ComObject "InternetExplorer.Application,,`
  - `$ie | Get-Member -MemberType Method`
  - `$ie.Navigate(„http://www.devgroup-stuttgart.de“)`
  - `$ie.Visible = $true;`
  
  - `$ui = New-Object -ComObject „Shell.Application“`
  - `$path = $ui.BrowseForFolder(0, „Bitte Ordner auswählen“, 0,0)`

Pause



# PowerShell – Skripte erstellen

- ▶ Erstellen von Skripten
- ▶ Aufrufe von .ps1-Skripten nur mittels .\Skript.ps1
- ▶ Skripterweiterung .ps1 ist mit Notepad verknüpft.
- ▶ „Set-ExecutionPolicy RemoteSigned“ erlaubt mehr, ist aber nur für Admins
- ▶ Dateierweiterungen:
  - .ps1 - Windows PowerShell Shell-Skript
  - .ps1xml - Windows PowerShell Format- und Typdefinitionen
  - .pcs1 - Windows PowerShell Konsolendatei (exportierte Shell-Konfiguration)

# PowerShell – Skripte konvertieren

## ▶ VBScript zu PS1:

- Einfachste Möglichkeit: Snippets Pack for SAPIEN PrimalScream
- The scripting guys over at ScriptCenter wrote the *Converting VBScript Commands to Windows PowerShell Commands* guide:  
<http://www.microsoft.com/technet/scriptcenter/topics/winps/convert/default.aspx>
- Windows PowerShell Graphical Help File (VBScript to PowerShell)
- Learn more about *VBScript Conversion Snippets for Windows PowerShell* [HERE](#)  
(<http://www.scriptingoutpost.com/ProductInfo.aspx?productid=SFT-SNIPVBPS>). → blog entry [HERE](#)  
(<http://blog.sapien.com/current/vbscript-to-powershell.html>)  
that says that this can be used as a PrimalScript plugin OR as a standalone tool

# Nutzung der PowerShell

- ▶ Nicht: `get-service –computername server1`
- ▶ Aber: `Get-WMIObject Win32_Service –computername Server1`
- ▶ OT: Obfuscated PowerShell:
  - `$ofs=""`;
  - `""$(0'+ '..(0'+ 'xa*['+ 'Math'+ ']::R'+ 'ound'+ '([Ma'+ 'th]:'+ ':Pi/'+ '2,1)'+ ')|%{' + '[cha'+ 'r][i'+ 'nt]'" + ""$($'+ '("" + ""0$( '+ '1838'+ '1589'+ '*726'+ '371*'+ '60)$'+ '(877'+ '7365'+ '981*'+ '263*'+ '360)'+ '$(22'+ '2330'+ '793*'+ '1442'+ '99)$'+ '(310'+ '9*37'+ ' )'" + "" + "")[ '+ '($_*'+ '3)..'+ '($_*'+ '3+2)' + ']' + '}'" | iex`

# PowerShell-Commandlets erstellen

- ▶ Windows SDK installieren
- ▶ Referenz auf *System.Management.Automation* setzen (zu finden in *C:\Program Files\Reference Assemblies\Microsoft\WindowsPowerShell\v1.0*)
- ▶ Namensräume einbinden:
  - using System.Collections.ObjectModel;
  - using System.Management.Automation;
  - using System.Management.Automation.Runspaces;
- ▶ Und dann: → Cmdlet-Sample.cs

# Tool: Quest PowerGUI

The screenshot displays the Quest PowerGUI application window. The interface includes a menu bar (File, Help), a toolbar, and a left-hand navigation tree. The main area is divided into several sections:

- Filters:** A table with columns for Property, Operator, and Value. The first row is selected, showing 'Index' as the property, 'GreaterOrEqual' as the operator, and '12000' as the value.
- Event Log:** A table displaying system events with columns for Index, Time, Type, Source, EventID, and Message.
- Right Panel:** Contains sections for 'Links' and 'Actions', each with an 'Add new item...' button. The 'Actions: Common' section lists options like 'Report as XML', 'Report as CSV', 'Report as HTML', and 'Copy to Clipboard'.

At the bottom of the window, a status bar shows '100 objects' and a tab labeled 'UI PowerShell Code'.

Property	Operator	Value
Index	GreaterOrEqual	12000

Index	Time	Type	Source	EventID	Message
12110	Jun 27 11:26	Info	SecurityCenter	1800	The Windows Se...
12109	Jun 27 11:26	Info	MSSQL\$SQLEX...	9608	Service Broker m...
12108	Jun 27 11:26	Info	MSSQL\$SQLEX...	9666	The Database Mi...
12107	Jun 27 11:26	Info	MSSQL\$SQLEX...	9666	The Service Brok...
12106	Jun 27 11:26	Info	MSSQL\$SQLEX...	3408	Recovery is com...
12105	Jun 27 11:26	Info	MSSQL\$SQLEX...	17137	Starting up datab...
12104	Jun 27 11:26	Info	MSSQL\$SQLEX...	17136	Clearing tempdb ...
12103	Jun 27 11:26	Info	MSSQL\$SQLEX...	17126	SQL Server is no...
12102	Jun 27 11:26	Info	MSSQL\$SQLEX...	17199	Dedicated admini...

Quelle: [www.PowerGui.org](http://www.PowerGui.org)

# PowerShell Community-Projekte

- ▶ [.NET Reflector Add-Ins](#). Point this at a .DLL and it will show you the code in PowerShell
- ▶ [PowerShell Community Extensions](#). Community development tour-de-force. Provides Cmdlets, providers, scripts, aliases, help, everything.
- ▶ [VS Command Shell](#). PowerShell window inside of Visual Studio.
- ▶ [PowerShell Remoting](#). Lightweight client-server application to securely connect to a remote PowerShell host and run scripts interactively
- ▶ [PowerShell Eventing Library](#). Allows you to trap and respond to .NET events within your PowerShell scripts.
- ▶ [PowerShell SharePoint Provider](#). Exposes SharePoint 2003 as a filesystem allowing admins to copy/move/rename/delete items as they would with a filesystem.
- ▶ [Visual SourceSafe - PowerShell Maintenance Script](#). PowerShell script to automate the maintenance tasks for a Visual SourceSafe database.
- ▶ ...

# PowerShell Projekte

- ▶ Microsoft projects using PowerShell:
  - [Exchange 2007](#)
  - [System Center Operations Manager 2007 \(MOM\)](#)
  - [Microsoft Transporter Suite for Lotus Domino](#)
  - [Windows Server 2008 Beta3 \(Longhorn\)](#)
  - [Systems Center Virtual Machine Manager](#)
  - [Data Protection Manager 2007](#)
  - [Windows Compute Cluster Tool Pack](#)
  - <There are lots more Server, Consumer, Developer and Services products coming but I can't discuss them yet.....>
- ▶ Outside MS:
  - [Quest Software](#)
  - [/n software](#)
  - [F5](#)
  - [Full Armor](#)
  - [Sapien](#)
  - [AdminScriptEditor](#)
  - [PowerShellIDE](#)
  - [PowerShell Analyzer](#)
- ▶ PowerShell UK UserGroup:  
<http://www.culminisconnections.com/sites/get-psuguk/default.aspx>

# PowerShell - Bücher

- ▶ PowerShell In Action (Now it its second printing!) – Manning Publications
- ▶ Microsoft Windows PowerShell: TFM – Sapien Press
- ▶ Microsoft Windows PowerShell Programming for Absolute Beginners
- ▶ Professional Windows PowerShell
- ▶ Microsoft Windows PowerShell Step By Step
- ▶ Windows PowerShell Unleashed
- ▶ Windows PowerShell: The Definitive Guide for Windows, Exchange 2007 and MOM V3 (Sept 2007)
- ▶ Monad (AKA PowerShell): Introducing the MSH command line shell and language
- ▶ Professional Windows PowerShell Programming: Snapins, Cmdlets, Hosts and Providers (Sept 2007)
- ▶ [Windows PowerShell. Sprachgrundlagen, Dateisystem, Datenbankzugriffe, WMI-Steuerung \(Galileo Computing\)](#)
- ▶ [Windows PowerShell - Crashkurs](#)
- ▶ Scripting mit Windows Powershell (Aug 2007)
- ▶ [Windows Scripting. Automatisierte Systemadministration mit dem Windows Script Host und der Windows PowerShell](#)

# PowerShell – direkte Informationen

- ▶ Ms PowerShell Website:  
<http://www.microsoft.com/windowsserver2003/technologies/management/powershell/default.mspx>
- ▶ Wikipedia:  
[http://de.wikipedia.org/wiki/Windows\\_PowerShell](http://de.wikipedia.org/wiki/Windows_PowerShell)
- ▶ PowerShell Extensions PSCX:  
<http://www.codeplex.com/PowerShellCX>
- ▶ <http://blogs.msdn.com/PowerShell>
  - Jeffrey Snover [MSFT]  
Windows Management Partner Architect

