

## Scripting

Ein Script ist nichts anderes als eine Textdatei, die ein oder mehrere Cmdlets enthält. Im einfachsten Fall ein Einzeiler, den man nicht ständig eintippen möchte.

Natürlich wird es schnell komplexer. Wenn man schon einmal dabei ist, kann man das Ganze mit Kontrollstrukturen anreichern, also zum Beispiel Bedingungen einfügen, in einer Schleife mehrere Objekte erstellen oder bearbeiten, ein Protokoll schreiben und Mechanismen für die Fehlerbehandlung hinzufügen.

### Ausführungsrichtlinie (execution policy)

Zunächst muß festgestellt werden, ob und unter welchen Bedingungen die Scriptausführung erlaubt ist. Die wichtigsten Einstellungen sind:

RemoteSigned	Scripte, die als aus dem Internet stammend gekennzeichnet sind, (rechte Maustaste –Dateieigenschaften) müssen vertrauenswürdig signiert sein. <b>Standardeinstellung bei Windows Server.</b>
Restricted	Scriptausführung ist nicht erlaubt. <b>Standard in Windows Client OS</b>
Unrestricted	Scriptausführung ist erlaubt, Warnungsmeldung bei Skripten aus dem Internet.
Bypass	Scriptausführung ist erlaubt, <b>keine</b> Warnungsmeldung bei Skripten aus dem Internet.
AllSigned	Alle Skripte müssen signiert sein.

Die Ausführungsrichtlinie kann mit `Get-ExecutionPolicy` ermittelt werden.

Geändert wird die Ausführungsrichtlinie mit `Set-ExecutionPolicy`

z.B.: `Set-ExecutionPolicy RemoteSigned`

Sinnvoll ist es, die Einstellung per Gruppenrichtlinie vorzunehmen:

Computerkonfiguration - Administrative Vorlagen – Windows-Komponenten – WindowsPowerShell

Einstellungen per Gruppenrichtlinie können lokal **nicht** geändert werden.

### Parameterübergabe

Es gibt verschiedene Wege, Daten an das Skript zu übergeben. Diese werden in den meisten Fällen sinnvollerweise in Variablen gespeichert. Während der Entwicklung oder zu Testzwecken kann die Zuweisung direkt am Anfang des Skriptes erfolgen.

z.B.: `$pfad="C:\Daten\Werte.csv"`

Je nach Einsatzscenario könnten die Werte auch abgefragt,

`$wert = Read-Host "Gib den Wert ein"`

aus einer Datei eingelesen

`$user = Import-Csv "c:\Daten\user.csv"`

oder als Parameter hinter dem Skriptaufruf übergeben werden.

`.\script.ps1 -Pfad ="C:\Daten\Werte.csv" -Wert 10`

## Skriptausführung

- A) Windows-Explorer – rechte Maustaste auf Skriptdatei > "Mit PowerShell ausführen"  
Problem: Keine Parameterübergabe möglich aber Read-Host oder Parameter im Skript  
Gegen das automatische Schließen des Fensters hilft `Pause` am Ende des Skriptes.
- B) Aufruf aus der PowerShell Commandline.  
Aus dem Skriptverzeichnis heraus: `PS C:\Scripte> .\script.ps1`  
Man beachte `".\"` vor dem Skriptnamen.  
Oder von einem beliebigen Ort: `PS C:\> Powershell.exe c:\scripte\script.ps1`
- C) Aus der Entwicklungsumgebung (ISE), z.B. in der Testphase
- D) Es gibt viele weitere Wege: z.B.: Logonscript, Registry (Run, RunOnce), als Scheduled Task  
`Invoke-Command -Filepath <Pfad>`