

Powershell

Einführung

Inhaltsverzeichnis

1	Was ist ein Skript?	1
2	Eigenschaften von Powershell	2
3	PowerShell-Versionen.....	3
4	Die wichtigsten cmdlets.....	4
5	Übungen	5

1 Was ist ein Skript?

Betrachten wir zwei Beispiele [bucd]:

- Ein Programm durchsucht bestimmte Verzeichnisse auf der Festplatte nach geänderten Dokumenten, die gefundenen Dateien werden komprimiert und auf einem entfernten Computer archiviert. Es sind keine Benutzereingriffe nötig. Darüber hinaus kann veranlasst werden, dass das Skript zu einer bestimmten Zeit ausgeführt wird.
- Ein Programm kontrolliert aufgrund einer Artikelliste den Lagerbestand und findet heraus, welche Artikel bestellt werden müssen. Das Programm erstellt automatisch eine «Bestellliste».

Das erste Programm erledigt Aufgaben, die typische Betriebssystemfunktionen umfassen. Es ist „betriebssystemnahe“ und geeignet für ein Skript. Für jede Skriptsprache (Perl, Shell, PHP usw.) gibt es einen entsprechenden Interpreter. Linux-Shell, DOS-Batch, VBS und PowerShell sind betriebssystemspezifische Skriptsprachen. PowerShell Code ist nicht lauffähig auf einem anderen Betriebssystem. Das Skript ist damit nicht portierbar. Als Vorteil bietet uns PowerShell den Zugriff auf das Windows-Objektmodell, was dem Code eine grosse Mächtigkeit in wenigen Zeilen ermöglicht.

Das zweite Programm greift meist auf Datenbanken zu. Evtl. werden weitere Abfragen auf lokale und entfernte Daten hinzugenommen, um z.B. saisonale Schwankungen bei den Kundenbedürfnissen zu berücksichtigen. Bei solchen komplexen Anwendungsaufgaben werden bevorzugt höhere, universelle Programmiersprachen verwendet, wie Java, C# usw.

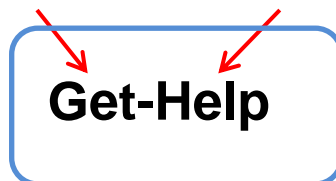
Aufgrund der unterschiedlichen Anforderungen und Rahmenbedingungen werden unterschiedliche Programmiersprachen zum Einsatz gebracht [bucd]:

- Das 1. Beispiel steht typischerweise für ein Skript: Die Übersetzung in einen Maschinenbefehl übernimmt ein sogenannter Interpreter, der jede Anweisung im Skript einzeln analysiert, übersetzt und ausgeführt. Dieser Interpreter ist bei jeder Ausführung des Skripts nötig. Es entsteht keine ausführbare Datei. Funktioniert ein Befehl nicht, bleibt das Skript in dieser Zeile stehen. Ein Benutzer kann in der Konsole einzelne Befehle absetzen und das Ergebnis auswerten. Das enge Hin und Her mit der Maschine wird als interaktiv bezeichnet.
- Bevor das Programm im 2. Beispiel vom Prozessor ausgeführt werden kann, muss es als Ganzes übersetzt werden. Für diese Zwischenschritte werden Compiler, Linker usw. eingesetzt. Am Ende entsteht z.B. eine „.exe“-Datei als Compilat, die vom Prozessor ausgeführt werden kann. Aufgrund der Zwischenschritte dauert es länger bis der Benutzer zur ausführbaren Datei kommt. Dieser maschinennahe, binäre Code wird ohne Programmierumgebung ausgeführt, was in der Regel aber schneller geht als bei einem Skript.

Ein Skript ist ein Programm, das als Textdatei gespeichert wurde. Bei jeder Durchführung wird das Skript von einem Interpreter übersetzt und ausgeführt. [bucd]

2 Eigenschaften von Powershell

Befehle bestehen immer aus Verb und einem Nomen:



Get-Help

Parameter erhalten ein -



Get-Help -detailed

Ältere bekannte Befehle bleiben bestehen:

Help statt Get-Help –detailed (Win cmd)
man statt Get-Help –detailed (Unix)

Genannt werden die Befehle in Powershell:



cmdlet

Powershell arbeitet mit Objekten (Nicht mit Texten, wie der Vorgänger!)

Jedes Objekt verfügt über Eigenschaften und Methoden:

Get-ChildItem | get-member

..liefert sämtliche Eigenschaften und Methoden eines Objekts.

Objekte und Listen von Objekten können von einem cmdlet zum nächsten übergeben werden. Dazu wird die Pipe „|“ verwendet. Die Ausführungsrichtung ist dabei immer von links nach rechts.

3 PowerShell-Versionen

Die PowerShell 5.0 ist als Teil von Windows 10 am 29.07.2015 erschienen. Sie ist als Erweiterung für Windows Server 2008 R2 (mit Service Pack 1) und Windows Server 2012/2012 R2 enthalten, ebenso für Windows 7 und Windows 8.1. [Schwi]

Die Versionen unterscheiden sich in ihrer Mächtigkeit [Schwi]

Version	Betriebssystem	Anzahl der Commandlets und Funktionen
PowerShell 5.0	Windows 10	1'404
PowerShell 4.0	Windows Server 2012 R2	1'376
PowerShell 4.0	Windows 8.1	1'132
PowerShell 4.0	Windows 7	573
PowerShell 3.0	Windows 8	945
PowerShell 3.0	Windows 7	561
PowerShell 2.0	Windows 7	273
PowerShell 1.0	Alle	163

Diese Zahlen können wie folgt ermittelt werden [Schwi]:

(Get-Command) | group commandtype

4 Die wichtigsten cmdlets

Get-Help (update-help)

Mit Get-help erhalten Sie Informationen zu den Cmdlets:

```
PS C:\WINDOWS\system32> get-help

THEMA
    Hilfesystem zu Windows PowerShell

KURZBESCHREIBUNG
    Zeigt Hilfe zu windows PowerShell-Cmdlets und -Konzepten an.

DETAILBESCHREIBUNG
    In der windows PowerShell-Hilfe werden Cmdlets, Funktionen, Skripts und Module von windows
    Elemente der Windows PowerShell-Sprache.
```

Get-Member

Mit Get-Member erhalten Sie eine Auflistung aller Methoden und Eigenschaften eines Objektes. Übergeben wird das Objekt mit der Pipe „|“

```
PS C:\WINDOWS\system32> get-help |Get-Member

TypeName: System.String

Name      MemberType      Definition
----      -
Clone     Method          System.Object Clone(), System.Object ICloneable.Clone(), int CompareTo(System.Object value), int Contains(string value)
CompareTo Method          void CopyTo(int sourceIndex, char[] dest)
Contains  Method          bool EndsWith(string value), bool Equals(System.Object obj), bool Equals(string value)
CopyTo    Method          bool Equals(System.Object obj), bool Equals(string value)
EndsWith  Method          bool Equals(System.Object obj), bool Equals(string value)
Equals    Method          bool Equals(System.Object obj), bool Equals(string value)
```

Get-Command

Damit erhalten Sie eine Auflistung der zur Verfügung stehenden Cmdlets. Wenn Sie die Befehle nur eines Bereiches brauchen, können Sie das gewünschte Modul angeben:

Bsp.: PS C:\> Get-Command -Module Microsoft.PowerShell.Security, PSScheduledJob

```
PS C:\WINDOWS\system32> Get-Command

CommandType Name                                     Version Source
-----
Alias       Add-ProvisionedAppxPackage              3.0     Dism
Alias       Apply-WindowsUnattend                   3.0     Dism
Alias       Disable-PhysicalDiskIndication           2.0.0.0 Storage
Alias       Disable-StorageDiagnosticLog             2.0.0.0 Storage
Alias       Enable-PhysicalDiskIndication            2.0.0.0 Storage
Alias       Enable-StorageDiagnosticLog              2.0.0.0 Storage
Alias       Flush-Volume                            2.0.0.0 Storage
Alias       Get-DiskSNV                             2.0.0.0 Storage
Alias       Get-PhysicalDiskSNV                     2.0.0.0 Storage
```

Select-Object

Das Select-Object gibt Ihnen ein Objekt oder eine Liste von Objekten zurück.

```
PS C:\WINDOWS\system32> get-date

Freitag, 1. September 2017 08:56:21

PS C:\WINDOWS\system32> get-date | Select-Object *

DisplayHint : DateTime
DateTime    : Freitag, 1. September 2017 08:56:25
Date        : 01.09.2017 00:00:00
Day         : 1
DayOfWeek   : Friday
DayOfYear   : 244
Hour        : 8
Kind        : Local
Millisecond  : 820
Minute      : 56
```

5 Übungen

1. Aktualisieren Sie die Hilfe von Powershell mit dem cmdlet Update-Help.
2. Zeigen Sie die Hilfe für das cmdlet Get-Service an. Vergleichen Sie die Ausgaben, wenn Sie das cmdlet ohne Parameter oder jeweils mit einem der Parameters -Detailed, -Full, oder -Examples
3. Finden Sie heraus, welche Cmdlets den Parameter -Verb verwenden. Lesen Sie in der Hilfe zu einem der angezeigten Cmdlets nach, was der Parameter bedeutet.
4. Suchen Sie in der Hilfe nach Beispielen für das Cmdlet Get-Process!
5. Finden Sie heraus, welche Befehle das Verb *Import* verwenden.
6. Ermitteln Sie welche Eigenschaften und Dienste das Cmdlet Get-Service besitzt!
7. Ermitteln Sie
 - a) die Syntax des Cmdlets Get-Module
 - b) Beispiele zum Einsatz des Cmdlets Get-PSDrive
 - c) was der Parameter -Format des Cmdlets Get-Date bedeutet.

8. Rufen Sie die vollständigen Hilfetexte zu den Cmdlets Get-History, Get-Process und Get-PSProvider auf und lesen Sie sie durch
9. Welche Eigenschaften stehen dem Cmdlet Get-Module zur Verfügung?
10. Mit welchem Befehl lässt sich ein Dienst auf Ihrem Windows-System starten? Es sind auch Beispiele für den PowerShell-Befehl gewünscht.
Tipp: Beginnen Sie Ihre Suche mit den folgenden Befehlen:

```
Get-Help *start  
Get-Help *start*
```
11. Gesucht ist eine Liste der laufenden Prozesse (get-process), die nach der CPU-Nutzung sortiert (sort-object) ist. Zuerst sollen jene Prozesse stehen, die die CPU am meisten belasten. Es genügt, wenn Sie sich auf 10 Prozesse beschränken (select-object).
12. Geben Sie alle Dienste aus und sortieren Sie diese nach der Eigenschaft status. Zudem sollen nur der Name und der Status ausgegeben werden

In diesem Dossier verwendete Quellen

- [bucd] Didier Buclin: „Modul 122: Abläufe mit Skripts/Makros automatisieren“, Compendio Verlag
- [schwi] Holger Schwichtenberg: „Windows PowerShell 5.0“, Hanser Verlag

Historie

Dokument erstellt	R. Müller	31.08.2017
Schreibfehler korrigiert	R. Müller	24.09.2019
Aufgabe 6 umformuliert	R. Müller	22.10.2019