

Windows PowerShell Cmdlets

1 Einführung

Commandlet (kurz Cmdlet) werden die PowerShell-Befehle genannt.

Ein Cmdlet hat folgende **allgemeine Form**:

Verb-Substantiv [-Parameter1 Wert1] [-Parameter2 Wert2] ...

Eine Liste der verfügbaren Commandlets erhalten Sie über das Cmdlet **get-command**:

PS> get-command	Alle verfügbaren Befehle
PS> get-command -noun service	Alle Befehle mit dem Sustainiv service
PS> get-command -verb get	Alle verfügbaren Befehle mit dem Verb get
PS> get-command get-*	Wildcards sind erlaubt
PS> get-command -name get-date	Bestimmter Befehl

Hilfe zu einem bestimmten Cmdlet (hier **get-process**) erhalten Sie mit Hilfe des Cmdlets **get-help** wie folgt:

PS> get-help get-process	
PS> get-gelp get-process -examples	Beispiele anzeigen

2 Einige Cmdlets

Auf den Aufruf von Cmdlets mit dem Verb **get** erfolgt üblicherweise eine Ausgabe.

Beispiele:

```
PS>get-date  
  
PS>get-process  
  
PS>get-host
```

Über Cmdlets mit dem Verb **set** lassen sich Änderungen am System vornehmen.

Beispiele: `set-date, set-acl`

Es gibt noch weitere Verben, wie:

3 Pipeline

Mit der Pipeline lassen sich Befehle kombinieren.

Beispiel Prozess stoppen:

Notepad (Editor) starten	
<code>Get-process -name notepad</code>	Prozess anzeigen
<code>Get-process -name notepad stop-process</code>	Prozess stoppen

Beispiel Ausgabe formatieren:

```
get-date | format-list
```

4 Objekt als Rückgabewert

Diese Cmdlets mit dem Verb **get** geben immer ein **Objekt** oder eine **Liste von Objekten** zurück. Ein Objekt ist eine Datenstruktur, mit Eigenschaften (Werte) und Methoden (Funktionen).

Mit dem Cmdlet **get-member** lässt sich die Objektstruktur anzeigen:

```
PS> get-process | get-member  
  
PS> get-date | get-member
```

Mit dem Cmdlet **select-object** lassen sich alle Eigenschaften eines Objekts anzeigen:

```
PS> get-date | select-object *
```

Einzelne Eigenschaften lassen sich wie in der Objektorientierung üblich mit **"objekt.eigenschaft"** anzeigen:

```
PS> (get-date).DayOfWeek                      Ausgabe Wochentag
```

5 Übung 1 Cmdlets

1) Was gibt das Cmdlet **get-ChildItem** aus?

2) Welcher DOS-Befehl entspricht dem Cmdlet **get-ChildItem**?

3) Wie können Sie rekursiv alle Verzeichnisse und Dateien ausgeben?

4) Welche Cmdlets weisen das Verb **use** auf?

5) Geben Sie die Meldungen aus dem Ereignisprotokoll (Typ Application) von heute aus.
Hinweis: Heute entspricht "**\$ (get-date) .date**"

6) Ermitteln Sie das heutige Datum

7) Geben Sie den aktuellen Jahrestag aus!

8) Listen Sie die Cmdlets auf, welche mit der Verwaltung von Diensten zu tun haben (Dienst heisst auf Englisch "Service").

9) Geben Sie die IP-Konfiguration der aktiven Schnittstelle aus (wahrscheinlich LAN-Adapter).
Hinweis: Grenzen Sie die Ausgabe mit den vorhandenen Parametern ein!

10) Geben Sie alle Informationen zu den lokalen Disks aus!

11) Geben Sie nur die IPv4-Adresse (192.168.xxx.xxx) der aktiven Schnittstelle aus

12) Welche Eigenschaften und Methoden weist der Output des Cmdlets **get-hotfix** auf?

13) Geben Sie alle Eigenschaften des letzten Hotfixes (unterster in der Liste) aus!

6 Filtern mit "Where-Object"

Mit dem Befehl **Where-Object**, lassen sich Ausgaben filtern.

Beispiel, alle Services, welche den Status "Running" aufweisen ausgeben ("\$_" stellt das aktuelle Objekt dar!):

```
get-service | Where-Object {$_.status -eq "running"}
```

Beispiel, alle Services mit dem Starttyp "Automatic", welche nicht gestartet sind, starten:

```
get-service | Where-Object {$_.starttype -eq "automatic"  
-and $_.status -ne "running"} | start-service
```

Beispiel: Die Eigenschaften "EventId" und "Message" aus dem Ereignisprotokoll "System" von heute ausgeben:

```
get-eventlog -LogName system -After (get-date).date |  
Select-Object eventId, message
```

Die wichtigsten **Vergleichsoperatoren** sind:

-eq	gleich
-ne	Nicht gleich
-gt	grösser
-ge	Grosser oder gleich
-lt	kleiner
-le	Kleiner oder gleich
-like	Vergleich mit Wildcards

Die wichtigsten **logischen Operatoren** sind:

-not	nicht
-and	und
-or	oder
-xor	exklusiv oder

7 Übung 2: Cmdlet-Ausgaben filtern

- 1) Listen Sie alle Dienste auf, welche nicht gestartet sind!

- 2) Listen Sie alle Eigenschaften der Prozesse auf, deren Eigenschaftsnamen mit "64" endet

- 3) Listen Sie alle Prozesse auf, die mehr als 500 Handles geöffnet haben!

- 4) Geben Sie alle Prozesse aus, welche mehr als 100s CPU-Zeit verbraucht haben!

- 5) Geben Sie Id, Prozessname und CPU-Verbrauch aller Prozesse, welche mehr als 100s CPU verbraucht haben, aus!

- 6) Geben Sie die Prozessnamen aller Prozesse aus, welche mehr als eine Sekunde CPU verbraucht haben und aus dem Verzeichnis "Programm Files (x86)" gestartet wurden

- 7) Geben Sie Name und Status der Drucker aus, welche über "Fileprint" angeschlossen sind!

- 8) Geben Sie die Systemmeldungen aus dem Ereignisprotokoll von heute, vom Typ Error und Warning aus!

- 9) Erstellen Sie eine Datei **test.txt** mit Inhalt (im Windows-Explorer) und geben Sie alle Eigenschaften dieser Datei aus!

- 10) Wie erhalten Sie die Dateierweiterung der Datei **test.txt**?

- 11) Geben Sie Namen und Grösse aller Bilddateien (jpg, png und gif) ihres persönlichen Laufwerks aus!

- 12) Kopieren Sie die Datei test.txt nach test1.txt!
