

Windows PowerShell

Ausgaben formatieren und umleiten

1 Theorie

Dieser Teil behandelt verschiedene Möglichkeiten zur Formatierung und Umleitung von Ausgaben.

Format-List	<p>Ausgabe als detaillierte Liste. Die Objekte werden dabei untereinander dargestellt. Diese Formatierung eignet sich, wenn viele Eigenschaften weniger Objekte auszugeben ist. Beispiel:</p> <pre>Get-Process -name "firefox" Select-Object * Format-List</pre>
Format-Table <small>(Standardausgabe bei vielen Cmdlets)</small>	<p>Ausgabe als Tabelle. Eignet sich für viele Objekte mit einer überhaubaren Anzahl von Objekten. Beispiel:</p> <pre>Get-Process Select-Object Id, Name, Cpu , VirtualMemorySize Format-Table</pre>
Out-GridView	<p>Zeigt die Ausgabe in einem separaten Fenster, in einer grafischen Tabelle an, welche zusätzliche Filtermöglichkeiten bietet. Beispiel:</p> <pre>Get-Process Out-GridView</pre>
Out-File (>)	<p>Ausgabe in eine Datei umleiten (>) oder bestehende Datei erweitern (>>). Beispiel:</p> <pre>Get-Process > prozesse.txt</pre> <pre>Get-Process Out-File prozesse.txt</pre>
Out-Null	<p>Ausgabe unterdrücken. Beispiel:</p> <pre>Get-Process Out-Null</pre>
Out-Host -Paging	<p>Seitenweise Ausgabe. Beispiel:</p> <pre>Get-Process Out-Host -Paging</pre>

Export-Csv	<p>Ausgabe in einer CSV-Datei ausgeben. <i>Beispiel:</i></p> <pre>get-process Export-Csv -Delimiter ";" prozesse.csv</pre>
ConvertTo-Html	<p>Ausgabe in HTML-Format. <i>Beispiel:</i></p> <pre>get-process select-object id, name, cpu ConvertTo-Html -Title "Prozessliste" > prozesse.html</pre>
Sort-Object	<p>Sortiert Ausgaben nach einer beliebigen Spalte. <i>Beispiel:</i></p> <pre>Get-Process Sort-Object -Property CPU</pre> <p>Absteigend sortieren:</p> <pre>Get-Process Sort-Object -Property CPU -Descending</pre>
Group-Object	<p>Objekte nach einer Eigenschaft gruppieren. <i>Beispiel:</i></p> <pre>Get-Service Group-Object -Property status</pre>
Measure-Object	<p>Statistische Werte ausgeben. <i>Beispiel:</i></p> <pre>Get-Process Measure-Object -Property CPU -Min -Max -Sum- -Ave</pre>
Foreach-Object	<p>Eine Liste von Objekten in einer Schleife abarbeiten. <i>Beispiele:</i></p> <pre>get-process ForEach-Object { write-host \$_.id \$_ .name} 1..100 ForEach-Object {\$_}</pre>

2 Übungen

- 1) Erstellen Sie eine Tabelle als Textdatei alle Dienste. Die Tabelle soll Name und Status anzeigen, nach Status sortiert und gruppiert (**Format-Table -GroupBy**) sein.

- 2) Geben Sie alle Alias (Get-Alias) als HTML-Tabelle aus, die den Namen und die Definition des Alias beinhalten soll.

- 3) Geben Sie die Anzahl Ereignisse der Ereignisanzeige "Anwendung" vom heutigen Tag aus!
Hinweis: Count-Eigenschaft von Measure-Object!

- 4) Geben Sie Name, Status und Starttyp aller Dienste sortiert und gruppiert nach Status aus!

- 5) Exportieren Sie alle Services, welche den Starttyp "Manuell" aufweisen in ein CSV.

- 6) Geben Sie alle Computerinformationen (**Get-Computerinfo**) in der Datei **c:/tmp/config.txt** aus! Es kann sein, dass ein Grossteil der Eigenschaften keinen Wert aufweisen. Finden Sie eine Lösung (Google hilft)!

- 7) Geben Sie die Computerinformationen, welche mit OS starten in der Datei **c:/tmp/OSconfig.txt** aus!

- 8) Geben Sie Name und Kommandotyp aller PS-Befehle, sortiert nach Name in einer grafischen Tabelle, in einem separaten Fenster aus!

- 9) Geben Sie für das Subnetz 192.168.102.0/24 alle IP-Adressen aus!

- 10) Geben Sie die Daten aller Tage (1. Januar bis 31. Dezember) des aktuellen Jahres aus!
