

## Allgemeine PowerShell-Befehle

1. Navigation durch die Windows Registry wie durch ein Dateisystem:

```
cd hkcu:
```

2. Rekursive Suche nach einem Textfragment in Dateien:

```
dir -r | select string "Suchwort"
```

3. Ausgabe der fünf speicherhungrigsten Prozesse:

```
ps | sort -p ws | select -last 5
```

4. Neustart eines Dienstes (stoppen und wieder starten) am Beispiel [DHCP](#):

```
Restart-Service DHCP
```

5. Alle Elemente eines Ordners ausgeben:

```
Get-ChildItem - Force
```

6. Elemente aus Verzeichnissen oder Ordnern rekursiv ausgeben:

```
Get-ChildItem -Force c:\directory -  
Recurse
```

7. Alle Dateien in einem Verzeichnis ohne einzelne Rückfragen löschen:

```
Remove-Item C:\tobedeleted -Recurse
```

8. Den aktuellen Computer neu starten:

```
(Get-WmiObject -Class  
Win32_OperatingSystem -ComputerName  
.) .Win32Shutdown(2)
```

## Per PowerShell Informationen sammeln

9. Informationen über Hersteller und Modell des Computers ausgeben:

```
Get-WmiObject -Class Win32_ComputerSystem
```

10. Informationen über das [BIOS](#) des aktuellen Computers ausgeben:

```
Get-WmiObject -Class Win32_BIOS -  
ComputerName .
```

11. Installierte Hotfixes (QFEs, Windows Updates) ausgeben:

```
Get-WmiObject -Class  
Win32_QuickFixEngineering -ComputerName .
```

12. Benutzernamen des aktuell angemeldeten Benutzers ausgeben:

```
Get-WmiObject -Class Win32_ComputerSystem  
-Property UserName -ComputerName .
```

13. Alle Namen der installierten Anwendungen auf dem aktuellen Computer ausgeben:

```
Get-WmiObject -Class Win32_Product -  
ComputerName . | Format-Wide -Column 1
```

14. Alle [IP-Adressen](#) des aktuellen Computers ausgeben:

```
Get-WmiObject -Class  
Win32_NetworkAdapterConfiguration -Filter  
IPEnabled=TRUE -ComputerName . | Format-  
Table -Property IPAddress
```

15. Umfassenderen Bericht über die IP-Konfiguration des aktuellen Computers ausgeben:

```
Get-WmiObject -Class  
Win32_NetworkAdapterConfiguration -Filter  
IPEnabled=TRUE -ComputerName . | Select-  
Object -Property [a-z]* -ExcludeProperty  
IPX*,WINS*
```

16. Netzwerkadapter des vorliegenden Computers mit aktiviertem DHCP ausgeben:

```
Get-WmiObject -Class  
Win32_NetworkAdapterConfiguration -Filter  
"DHCPEnabled=true" -ComputerName .
```

17. DHCP auf allen Netzwerkadaptern des aktuellen Computers aktivieren:

```
Get-WmiObject -Class  
Win32_NetworkAdapterConfiguration -Filter
```

```
IPEnabled=true -ComputerName . | ForEach-Object -Process {$_.EnableDHCP() }
```

## Softwareverwaltung über die PowerShell

18. MSI-Paket auf einem anderen Computer installieren:

```
(Get-WMIObject -ComputerName TARGETMACHINE -List | Where-Object -FilterScript {$_.Name -eq "Win32_Product"}) .Install(\\MACHINEWHEREMSIRESIDES\path\package.msi)
```

19. Eine installierte Anwendung durch ein MSI-basiertes Upgrade-Paket aktualisieren:

```
(Get-WmiObject -Class Win32_Product -ComputerName . -Filter "Name='name_of_app_to_be_upgraded'") .Upgrade(\\MACHINEWHEREMSIRESIDES\path\upgrade_package.msi)
```

20. MSI-Paket vom aktuellen Computer entfernen:

```
(Get-WmiObject -Class Win32_Product -Filter "Name='product_to_remove'" -ComputerName . ).Uninstall()
```

## PowerShell-Befehle zur VM-Verwaltung

21. Einen anderen Computer nach einer Minute herunterfahren:

```
Start-Sleep 60; Restart-Computer -Force -ComputerName TARGETMACHINE
```

22. Einen Drucker hinzufügen:

```
(New-Object -ComObject WScript.Network).AddWindowsPrinterConnection("\\printerserver\hplaser3")
```

23. Einen Drucker entfernen:

```
(New-Object -ComObject  
WScript.Network).RemovePrinterConnection(  
"\\printerserver\hplaser3 ")
```

24. Eine Remote-Sitzung starten (Remote-Verwaltung muss aktiviert sein):

```
enter-pssession TARGETMACHINE
```

25. PowerShell verwenden, um ein Script auf einem anderen Server auszuführen:

```
invoke-command -computername machine1,  
machine2 -filepath c:\Script\script.ps1
```