Powershell CMDLETS Cheatsheet

Prozesse schließen

Schließt einen Prozess nach Namen:

```
Stop-Process -Name "ProcessName" -Force
```

Schließt einen Prozess nach PID:

```
Stop-Process -ID PID -Force
```

Basic Networking PowerShell cmdlets

Auslesen der IP Configuration (ipconfig mit PowerShell)

```
Get-NetIPConfiguration
```

Auflisten aller Netzwerk-Adapter mit der Powershell

```
Get-NetAdapter
```

Einen spezifischen Netzwerk-Adapter in der Powershell anhand des Namens erhalten

```
Get-NetAdapter -Name *Ethernet*
```

Mehr Informationen zu Netzwerk-Adaptern in der Powershell anzeigen wie VLAN ID, Speed, Connection status

```
Get-NetAdapter | ft Name, Status, Linkspeed, VlanID
```

Treiber-Informationen in der Powershell anzeigen

```
Get-NetAdapter | ft Name, DriverName, DriverVersion, DriverInformation,
DriverFileName
```

Adapter Hardware Informationen in der Powershell

Das kann zum Beispiel wirklich hilfreich sein, wenn man den PCI slot des NIC kennen muss.

```
Get-NetAdapterHardwareInfo
```

Ein- und ausschalten eines Netzwerk Adapters in der Powershell

```
Disable-NetAdapter -Name "Wireless Network Connection"
Enable-NetAdapter -Name "Wireless Network Connection"
```

Umbenennen eines Netzwerk-Adapters in der Powershell

```
Rename-NetAdapter -Name "Wireless Network Connection" -NewName "Wireless"
```

IP Konfiguration mit der PowerShell

Erhalten der IP und DNS Address-Informationen

```
Get-NetAdapter -Name "Local Area Connection" | Get-NetIPAddress
```

Nur die IP Adresse erhalten

```
(Get-NetAdapter -Name "Local Area Connection" | Get-NetIPAddress).IPv4Address
```

Erhalten der DNS Server Adress-Informationen

```
Get-NetAdapter -Name "Local Area Connection" | Get-DnsClientServerAddress
```

Setzen der IP Adresse

```
New-NetIPAddress -InterfaceAlias "Wireless" -IPv4Address 10.0.1.95 -PrefixLength
"24" -DefaultGateway 10.0.1.1
```

oder, wenn du eine existierende Adresse ändern willst:

```
Set-NetIPAddress -InterfaceAlias "Wireless" -IPv4Address 192.168.12.25 -
PrefixLength "24"
```

Entfernen der IP Adresse

```
Get-NetAdapter -Name "Wireless" | Remove-NetIPAddress
```

DNS

Setzen des DNS Server

```
Set-DnsClientServerAddress -InterfaceAlias "Wireless" -ServerAddresses
"10.10.20.1","10.10.20.2"
```

Setzem des Interface auf DHCP

```
Set-NetIPInterface -InterfaceAlias "Wireless" -Dhcp Enabled
```

Verwalten des DNS Cache mit PowerShell

Anzeigen des DNS Cache:

```
Get-DnsClientCache
```

Leeren des DNS Cache:

```
Clear-DnsClientCache
```

NSlookup mit der PowerShell

```
Resolve-DnsName www.bing.com
```

Informationen von einem speziellen DNS Server:

```
Resolve-DnsName www.bing.com -Type MX -Server 8.8.8.8
```

DNS Recurds nach Typ:

```
Resolve-DnsName -Name www.bing.com -Type A
```

DNS-Only Abfragen:

```
Resolve-DnsName -Name www.bing.com -DnsOnly
```

In diesem Beispiel wird ein Name aufgelöst, der nur DNS verwendet. LLMNR- und NetBIOS-Abfragen werden nicht ausgegeben.

Abrufen der Netzwerkschnittstellenkonfiguration

```
Get-DnsClient
```

Zurücksetzen des DNS-Clients, um die von DHCP angegebenen Standard-DNS-Serveradressen zu verwenden

```
Get-DnsClient | Set-DnsClientServerAddress -ResetServerAddresses
```

Ping mit PowerShell

Für einen simplen ping command mit PowerShell, kann man das Test-Connection cmdlet nutzen:

```
Test-Connection bing.com
```

.. aber es gibt natürlich auch einen fortgeschrittenen Weg um eine Verbindung mit PowerShell zu testen:

```
Test-NetConnection -ComputerName www.bing.com
```

... und natürlich auch mit mehr Details aus der Test-NetConnection

```
Test-NetConnection -ComputerName www.bing.com -InformationLevel Detailed
```

Man kann auch einen Ping an mehrere IP Adressen mit der PowerShell senden:

```
1..99 | % { Test-NetConnection -ComputerName x.x.x.$_ } | FT -AutoSize
```

Tracert mit PowerShell

```
Test-NetConnection www.bing.com –TraceRoute
```

Port-Scans mit der PowerShell

Nutze die PowerShell um auf offene Ports zu testen:

```
Test-NetConnection -ComputerName www.bing.com -Port 80
Test-NetConnection -ComputerName www.bing.com -CommonTCPPort HTTP
```

Route mit PowerShell

Ändern von Einträgen in der Routingtabelle

Das Cmdlet Set-NetRoute ändert Einträge in der IP-Routingtabelle. Geben Sie Routen an, die mithilfe des Parameters DestinationPrefix oder des NextHop-Parameters geändert werden sollen. Sie können Routen auch mithilfe des Cmdlets Get-NetRoute angeben. Wenn Sie nicht angeben, welche Routen geändert werden sollen, ändern die Cmdlets alle Routen auf dem Computer.

IP-Routing ist der Prozess der Weiterleitung eines Pakets basierend auf der Ziel-IP-Adresse. Das Routing erfolgt auf TCP/IP-Hosts und an IP-Routern. Der sendende Host oder Router bestimmt, wo das Paket weitergeleitet werden soll. Um zu bestimmen, wo ein Paket weitergeleitet werden soll, konsultiert der Host oder Router eine Routingtabelle, die im Arbeitsspeicher gespeichert ist. Wenn TCP/IP gestartet wird, werden Einträge in der Routingtabelle erstellt. Sie können Einträge entweder manuell oder automatisch hinzufügen.

Ändern der Metric:

```
Set-NetRoute -RouteMetric 257
```

Ändern der Lifetime

```
$TimeSpan = New-TimeSpan -Days 1
Set-NetRoute -DestinationPrefix "192.168.0.0/24" -PreferredLifetime $TimeSpan
```

Auslesen von Einträgen in der Routingtabelle

Ruft die IP-Routinginformationen aus der IP-Routingtabelle ab:

```
Get-NetRoute | Format-List -Property *
```

Auslesen aller IPv6 Routes:

```
Get-NetRoute -AddressFamily IPv6
```

```
Get-NetRoute -Protocol Local -DestinationPrefix 192.168*
```

Abrufen des nächsten Hops für die Standardroute:

```
Get-NetRoute -DestinationPrefix "0.0.0.0/0" | Select-Object -ExpandProperty
"NextHop"
```

```
Get-NetRoute -InterfaceAlias Wi-Fi
```

Hinzufügen einer IP-Route zur Routingtabelle

```
New-NetRoute –DestinationPrefix "10.0.0.0/24" –InterfaceAlias "Ethernet" –NextHop
192.168.192.1
```

Entfernen von Routen aus der Routingtabelle

Alle Routen entfernen:

```
Remove-NetRoute
```

Entfernen von Routen mit bestimmten nächsten Hops

```
Remove-NetRoute -NextHop "192.168.0.1"
```

Netstat mit PowerShell

Das Get-NetTCPConnection-Cmdlet ruft aktuelle TCP-Verbindungen ab. Verwenden Sie dieses Cmdlet, um TCP-Verbindungseigenschaften wie lokale oder Remote-IP-Adresse, lokalen oder Remoteport und Verbindungsstatus anzuzeigen.

```
Get-NetTCPConnection
```

```
Get-NetTCPConnection –State Established
```

```
Get-NetTCPConnection -AppliedSetting Internet
```

NIC Teaming PowerShell commands

Erstellen eines neuen NIC Teaming (Network Interface Adapter Team)

```
New-NetLbfoTeam -Name NICTEAM01 -TeamMembers Ethernet, Ethernet2 -TeamingMode
SwitchIndependent -TeamNicName NICTEAM01 -LoadBalancingAlgorithm Dynamic
```

SMB PowerShell commands

Auslesen der SMB Client Konfiguration

```
Get-SmbClientConfiguration
```

Auflisten der SMB Verbindungen

```
Get-SmbConnection
```

Auflisten der SMB Mutlichannel Verbindungen

```
Get-SmbMutlichannelConnection
```

Dateien über SMB

Auflisten von geöffneten Dateien über SMB

```
Get-SmbOpenFile
```

Informationen zu einer speziellen Datei:

```
Get-SmbOpenFile -FileId 4415226383569 | Select-Object -Property *
```

Geöffnete Dateien nach Typ:

```
Get-SmbOpenFile | Where-Object -Property ShareRelativePath -Match ".DOCX"
```

Abrufen von Informationen zu einer Datei, die für einen SMB-Client geöffnet wurde

```
Get-SmbOpenFile -SessionId 4415226380393
```

Schließen einer geöffneten Datei nach Datei-ID:

```
Close-SmbOpenFile -FileId 4415226383589
```

Schließen von Dateien basierend auf der Session ID:

```
Close-SmbOpenFile -SessionId 4415226380393
```

Schließen geöffneter Dateien, die einer Dateinamenerweiterung entsprechen

```
Get-SmbOpenFile | Where-Object -Property ShareRelativePath -Match ".DOCX" | Close-SmbOpenFile -Force
```

SMB Direct (RDMA) Adapter

Auflisten der SMB Direct (RDMA) Adapter

```
Get-NetAdapterRdma
```

Hyper-V Networking mit der PowerShell

Erhalten und setzen der Network Adapter VMQ settings

Auflisten der Network Adapter VMQ settings:

```
Get-NetAdapterVmq
```

Aktivieren/Deaktivieren von VMQ:

```
# Disable VMQ
Set-NetAdapterVmq -Enabled $false
# Enable VMQ
Set-NetAdapterVmq -Enabled $true
```

Auflisten der VM Network Adapter

```
Get-VMNetworkAdapter -VMName Server01
```

Auflisten der VM Network Adapter IP Adressen

```
(Get-VMNetworkAdapter -VMName NanoConHost01).IPAddresses
```

Auflisten der VM Network Adapter Mac Adressen

```
(Get-VMNetworkAdapter -VMName NanoConHost01).MacAddress
```