



Windows Server 2012 R2  
Core-Server

Windows Server 2012 R2 Core Befehle

# Windows Server 2012 R2 Core Befehle

## Inhalt

Windows Server Vorbereitung .....	2
Wissenswertes vor und nach der Core Installation.....	2
Wichtige Befehle bevor wir den Server konfigurieren .....	3
Display-Auflösung ändern (Optional) .....	6
Statische IP-Adresse vergeben .....	6
DNS-Server Adresse konfigurieren .....	6
WINS Server Adresse konfigurieren .....	7
Computername ändern und zur Domain hinzufügen .....	8
Windows Updates konfigurieren.....	9
Windows Server Core aktivieren .....	10
Pagefile.sys konfigurieren .....	11
Festplattenpartition erstellen und formatieren.....	12
Neue Festplatte dem Server hinzufügen.....	14
Rollen und Features installieren.....	16
CMD.....	16
PowerShell.....	16
DSC - Desired State Configuration.....	17

## Windows Server Vorbereitung

Um dieses HowTo schreiben zu können, wurde das Ganze in einer Virtuellen Umgebung mit Oracle VM VirtualBox nachgestellt.

- ✓ Microsoft Windows Server 2012 R2 Standard Core Installation

## Wissenswertes vor und nach der Core Installation

Zum einen sollte man sich bewusst sein, das man auf einen Windows Server Core nicht alle Rollen und Features installieren kann. Microsoft hat hier eine Liste zusammengestellt, was man alles installieren kann/darf. <https://technet.microsoft.com/de-de/library/hh831786.aspx>

Es ist seit der Version Windows Server 2012 auch möglich, nach der Server Core Installation, eine Grafische Oberfläche zu installieren. Um sich die spätere Administration ein wenig zu erleichtern. Dieses werde ich aber, in diesem Howto nicht ansprechen.

Die Befehle für das nachträglich installieren der GUI, könnt Ihr in den oben aufgeführten Link nachlesen.

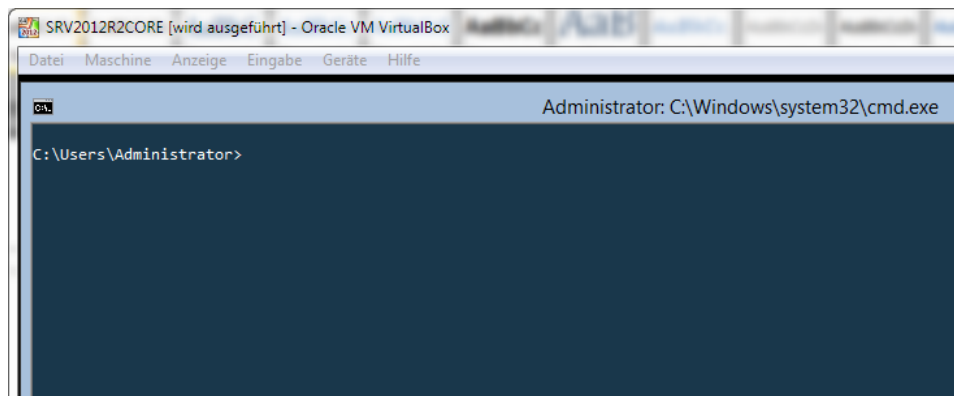
In diesen Howto kommen sowohl CMD-Befehle wie auch PowerShell-Befehle vor, habe versucht mit so wenig Aufwand wie möglich zum Ziel zu kommen. In der Standard CMD-Konsole nach der ersten Anmeldung, kann man mit den Befehlen **cmd** und **powershell** zwischen den beiden Konsolen hin und her wechseln.

Manchmal habe ich die Befehle absichtlich ausgeschrieben und manchmal findet man die Befehle in der abgekürzten Version vor.

Um die Lesbarkeit in Skripten oder Dokumentationen zu verbessern, würde ich jeden ans Herz legen die Befehle komplett auszuschreiben, es ist zwar immer mehr Schreibaufwand, aber ein paar Monate später wisst ihr immer noch was Ihr einst getan habt. ☺

## Wichtige Befehle bevor wir den Server konfigurieren

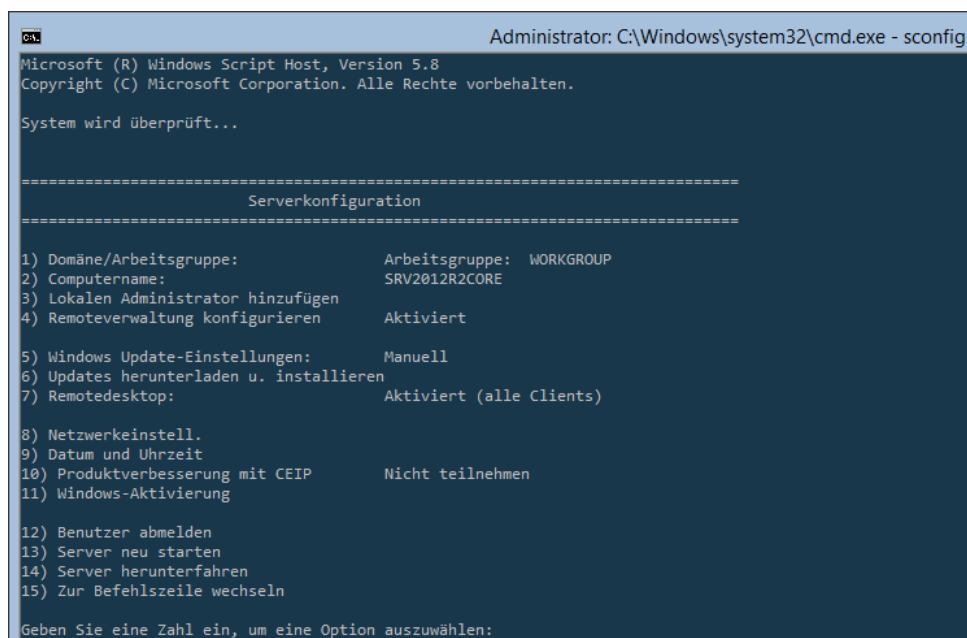
Nach der ersten Anmeldung sieht man nur ein CMD-Konsole und sonst nichts, das ist am Anfang sehr verwirrend.



Aber um hier mit dem administrieren beginnen zu können, gibt es ein Kommando das man sich sehr gut merken sollte.

### **sconfig**

Dieses Kommando ändert erst Mal alles. ☺ Hier wird ein VBS-Skript aus dem Verzeichnis **C:\Windows\System32\de-DE\sconfig.vbs** ausgeführt.



Wenn man möchte dass dieses Skript jedes Mal nach dem Anmelden zu sehen ist, muss man dieses in der Registry hinterlegen.

Dazu wechselt man mit der Eingabe **15** zurück in die CMD-Konsole und öffnet den Registry Editor.

### **regedit**

```

12) Benutzer abmelden
13) Server neu starten
14) Server herunterfahren
15) Zur Befehlszeile wechseln
Geben Sie eine Zahl ein, um eine Option auszuwählen: 15
C:\Users\Administrator>regedit_
  
```

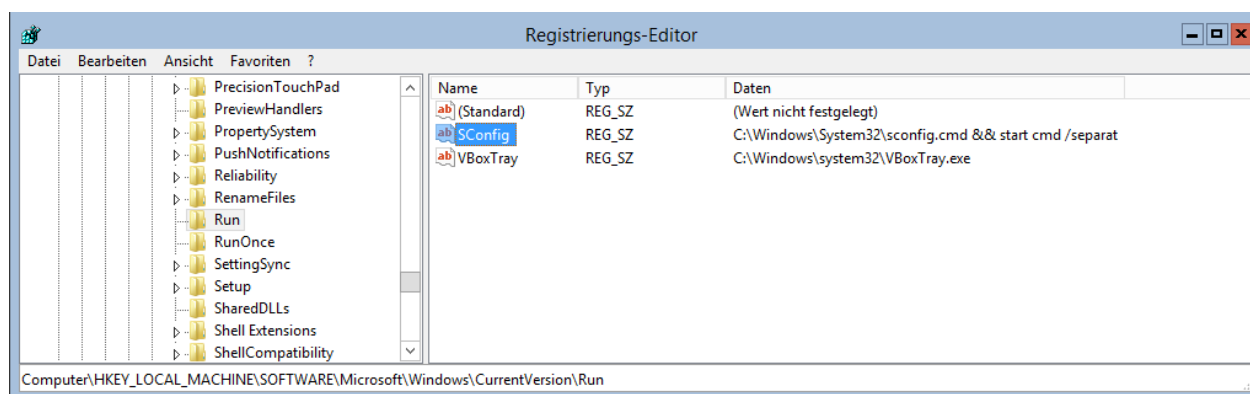
Hierzu navigiert man zu den Schlüssel wie unten aufgeführt, und legt einen neuen REG\_SZ Key an.

**Key:** *HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run*

**Type:** *REG\_SZ*

**Entry:** *SConfig*

**Value:** *C:\Windows\system32\sconfig.cmd && start cmd /separat*

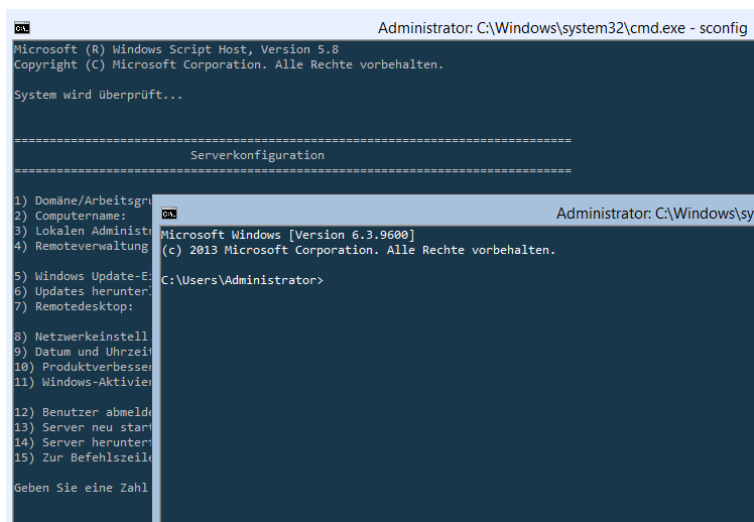


Nachdem man sich mit **logoff** vom System abgemeldet und erneut angemeldet hat. Sieht man immer das Hauptmenü von **SConfig**.

Man muss aber bedenken, wenn man Befehle auf der Kommandozeile eingeben möchte, muss **SConfig** mit **15** verlassen werden.

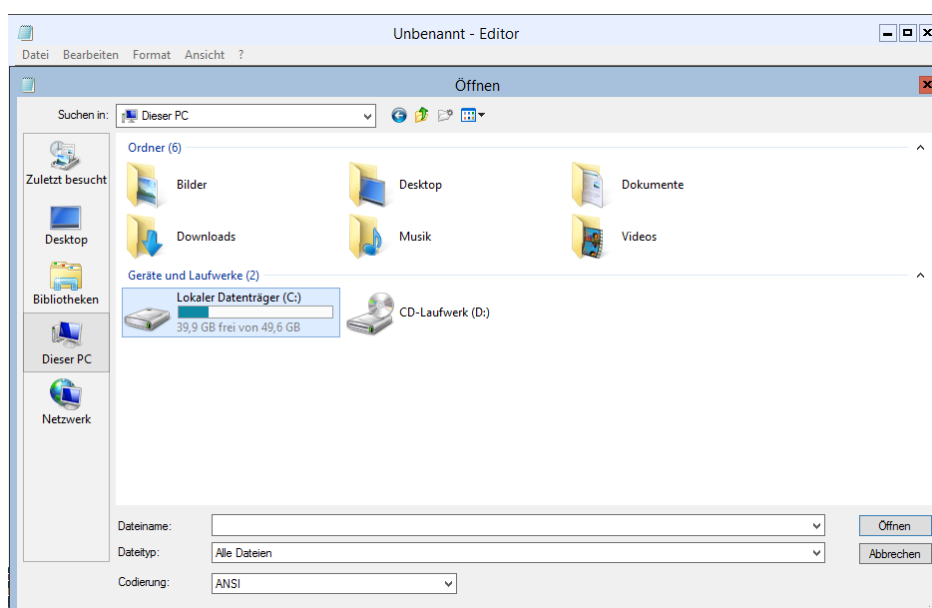
Was man aber ebenfalls machen kann um z.B. ein zweites CMD-Konsolenfenster zu haben ist, **SConfig** kurz mit **15** verlassen, ein zweites CMD-Konsolenfenster öffnen und im Hintergrund wieder **SConfig** starten. Somit müsste man nicht immer mit **15 - SConfig** verlassen und bei Bedarf wieder neu aufrufen. Müsst ihr aber selbst entscheiden wie Ihr gerne arbeiten wollt. 😊

### start cmd /separate



So hier noch ein paar Standardbefehle die immer mit an Board sind. Wenn Ihr z.B. einen Art Explorersersatz sucht, könnt ihr natürlich über den **DIR** oder **Get-ChildItem** Befehl in den Verzeichnissen navigieren/suchen oder Ihr missbraucht den Windows Editor - Notepad dafür.

**msinfo32, systeminfo (natürlich alle DOS Kommandos, siehe help), set, tasklist, taskkill, taskmgr, sc, net, regedit, notepad, timedate.cpl, intl.cpl**



## Display-Auflösung ändern (Optional)

So nun legen wir aber endlich los, bei Bedarf kann man die Display Auflösung ändern, dies ist natürlich nur relevant, wenn man mit Virtuellen Maschinen arbeitet.

CMD:

- ↳ `setres /i`
- ↳ `setres -w 1152 -h 864 -f`

PowerShell:

- ↳ `Get-DisplayResolution`
- ↳ `Set-DisplayResolution -Width 1152 -Height 864 -Force`

## Statische IP-Adresse vergeben

CMD:

- ↳ `netsh interface ipv4 show interface` oder `ipconfig /all`
- ↳ `netsh interface ipv4 set address name="<IDX-Wert>" source=static address=192.168.0.xxx mask=255.255.255.0 gateway=192.168.0.254`
- ↳ `netsh int ipv4 set addr "Ethernet" static 192.168.0.xxx 255.255.255.0 192.168.0.254`

PowerShell:

- ↳ `Get-NetAdapter`
- ↳ `Get-NetIPInterface -InterfaceIndex 12 -AddressFamily IPv4 (DHCP-Einstellung überprüfen)`
- ↳ `Set-NetIPInterface -InterfaceIndex 12 -AddressFamily IPv4 -Dhcp Disabled -PassThru`
- ↳ `New-NetIPAddress -AddressFamily IPv4 -InterfaceIndex 12 -IPAddress 192.168.178.102 -PrefixLength 24 -DefaultGateway 192.168.178.1`
- ↳ `Get-NetIPAddress -InterfaceIndex 12 -AddressFamily IPv4`

## DNS-Server Adresse konfigurieren

CMD:

- ↳ `netsh interface ipv4 show dnsserver name="Ethernet"`
- ↳ `netsh interface ipv4 set dnsserver name="Ethernet" source=static address=192.168.0.100 register=primary`
- ↳ `netsh int ipv4 set dns "Ethernet" static 192.168.0.100 primary`

oder

- ↳ **netsh interface ipv4 add dnsserver name="<IDX-Wert>" address=192.168.0.100 index=1**
- ↳ **netsh int ipv4 add dns <IDX-Wert> 192.168.0.101 index=2**

PowerShell:

- ↳ **Set-DnsClientServerAddress -InterfaceIndex 12 -ServerAddresses 192.168.0.100 -PassThru**
- ↳ **Set-DnsClientServerAddress -InterfaceIndex 12 -ServerAddresses 192.168.0.100,192.168.0.101 -PassThru**

DNS-Server Adresse löschen

CMD:

- ↳ **netsh interface ipv4 delete dnsserver name="Ethernet" address=192.168.0.100**
- ↳ **netsh int ipv4 del dns "Ethernet" all**

PowerShell:

- ↳ **Set-DnsClientServerAddress -InterfaceIndex 12 -ResetServerAddresses**

## WINS Server Adresse konfigurieren

- ↳ **netsh interface ipv4 set winsserver name="Ethernet" static address=192.168.0.100**

oder

- ↳ **netsh int ipv4 add winsserver name="<IDX-Wert>" address=192.168.0.100 index=1**
- ↳ **netsh int ipv4 add wins <IDX-Wert> 192.168.0.101 index=2**

WINS-Server Adresse löschen

- ↳ **netsh int ipv4 delete winsserver name="Ethernet" address=192.168.0.100**
- ↳ **netsh int ipv4 del wins "Ethernet" all**



## Computername ändern und zur Domain hinzufügen

Hier gibt es wieder mehrere Ansätze, entweder man nimmt die Einstellung über **SConfig** vor oder setzt folgende Befehle auf der CMD oder PowerShell Konsole ab.

### Computername ändern:

CMD:

- ↳ `netdom renamecomputer %COMPUTERNAME% /newname:NEUECOMPUTERNAME /force`
- ↳ `netdom renamecomputer %COMPUTERNAME% /newname:NEUECOMPUTERNAME /userd:<Domain\Administrator> /passwordd:* /force`
- ↳ `shutdown -r -t 0`

PowerShell:

- ↳ `Rename-Computer -ComputerName $env:COMPUTERNAME -NewName NEUECOMPUTERNAME -Force -Restart`

### Zur Domäne hinzufügen/entfernen

CMD:

- ↳ `netdom join %COMPUTERNAME% /domain:<Domain> /userd:<Domain\Administrator> /passwordd:* /reboot`
- ↳ `netdom remove %COMPUTERNAME% /domain:<Domain> /userd:<Domain\Administrator> /passwordd:* /reboot`

PowerShell:

- ↳ `Add-Computer -ComputerName $env:COMPUTERNAME -DomainName <Domain> -Credential <Domain\Administrator> -Force -Restart`
- ↳ `Remove-Computer -UnJoinDomainCredential <Domain\Administrator> -Passthru -Verbose -Restart`
- ↳ `Remove-Computer -ComputerName $env:COMPUTERNAME -UnjoinDomainCredential <Domain\Administrator> -WorkgroupName WORKGROUP -Restart -Force`

### Computername ändern und zur Domäne hinzufügen

PowerShell:

- ↳ `Add-Computer -ComputerName $env:COMPUTERNAME -Domain <Domain> -NewName NEUECOMPUTERNAME -Credential <Domain\Administrator> -Force -Restart`

## Windows Updates konfigurieren

Windows Updates Konfigurations-Status überprüfen:

- ↳ `%SystemRoot%\System32\cscript.exe %SystemRoot%\System32\SCRegEdit.wsf /AU`
- ↳ `%SystemRoot%\System32\cscript.exe %SystemRoot%\System32\SCRegEdit.wsf /AU /v`

Windows Updates aktivieren:

- ↳ `%SystemRoot%\System32\cscript.exe %SystemRoot%\System32\SCRegEdit.wsf /AU 4`
- ↳ `net stop wuauclt && net start wuauclt`
- ↳ `wuauclt /detectnow`

Windows Patch Manuel installieren

- ↳ `wusa.exe C:\Verzeichnis\`

Windows Updates deaktivieren

- ↳ `%SystemRoot%\System32\cscript.exe %SystemRoot%\System32\SCRegEdit.wsf /AU 1`
- ↳ `net stop wuauclt`

Windows Updates überprüfen

- ↳ `wmic qfe list`

Windows Updates bei Bedarf wieder deinstallieren

- ↳ `expand /f:* C:\Verzeichnis\`
- ↳ Man navigiert in das Verzeichnis C:\temp und öffnet die update.xml Datei, hier ersetzt man den Eintrag Install durch Remove und speichert die Datei wieder ab.
- ↳ `pkgmgr /n:C:\temp\`

## Windows Server Core aktivieren

- ↳ Server Core Lizenz Testzeitraum verlängern - **start /w %SystemRoot%\System32\slmgr.vbs /rearm**
- ↳ Server Core Lizenz deinstallieren - **start /w %SystemRoot%\System32\slmgr.vbs /upk**
- ↳ Server Core Lizenz aus der Registry löschen - **start /w %SystemRoot%\System32\slmgr.vbs /cpky**
- ↳ Server Core Echtlizenz einspielen - **start /w %SystemRoot%\System32\slmgr.vbs /ipk  
XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX**
- ↳ Server Core Online aktivieren - **start /w %SystemRoot%\System32\slmgr.vbs /ato**
- ↳ Ablaufdatum der Lizenz anzeigen lassen - **start /w %SystemRoot%\System32\slmgr.vbs /xpr**
- ↳ Server Core Lizenzinformationen abrufen - **start /w %SystemRoot%\System32\slmgr.vbs /dli oder /dlv oder /dlv all**

Falls zum Zeitpunkt der Aktivierung keine Internetverbindung besteht, kann durch einen gebührenfreien Anruf (Deutschland Tel.: 0800-284 828 3) die Lizenz aktiviert werden.

- ↳ Die Installations-ID anzeigen lassen - **start /w %SystemRoot%\System32\slmgr.vbs /dti =  
Installations-ID**
- ↳ Server manuell aktiviert - **start /w %SystemRoot%\System32\slmgr.vbs /atp ActivationID  
Installations-ID**

oder mit

- ↳ **DISM /online /Get-CurrentEdition**

-- Aktuelle Edition : ServerStandard

- ↳ **DISM /online /Get-TargetEditions** (Upgradeoptionen)

-- Zieledition : ServerDatacenter

- ↳ **DISM /online /Set-Edition:ServerStandard /ProductKey:XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX /AcceptEula**
- ↳ **DISM /online /Set-Edition:ServerDatacenter /ProductKey:XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX /AcceptEula**

## Pagefile.sys konfigurieren

Bei der Auslagerungsdatei sollte man baldmöglichst die Konfiguration anpassen, bei einer meiner ersten Server Core Installationen, hatten wir einen Hyper-V Server mit 128 GB Arbeitsspeicher aber nur einer 160 GB Festplatte, nach der Installation des Servers und ein paar Konfigurationsanpassungen wurde uns vom Monitoring gemeldet das die Festplatte vollgelaufen ist, es hat ein wenig gedauert bis wir auf die PageFile.sys Datei gekommen sind. 😊

**Bitte die unten aufgeführten Befehle immer in der CMD-Konsole ausführen, unter PowerShell haben manche Befehle nicht sauber funktioniert und es kamen seltsame Fehler.**

Pagefile Informationen anzeigen lassen

CMD:

- ↳ **wmic pagefile list /format**
- ↳ **wmic pagefileset list /format**
- ↳ **wmic computersystem list /format**

Pagefile Datei konfigurieren

- ↳ **wmic computersystem set AutomaticManagedPagefile=False**
- ↳ **wmic pagefileset create name="E:\\pagefile.sys"**
- ↳ **wmic pagefileset where name="E:\\pagefile.sys" set InitialSize=2047,MaximumSize=2048**
- ↳ **wmic pagefileset where name="C:\\pagefile.sys" delete**
- ↳ **shutdown -r -t 0**

PowerShell:

Wer gerne mit PowerShell das ganze konfigurieren möchte, hier eine gute Webseite dazu.

<https://positivedb.wordpress.com/2014/02/04/powershell-pagefile-configuration>

## Festplattenpartition erstellen und formatieren

Für den Test habe ich einen Server Core installiert mit einer 50 GB Festplatte, der C:\ Partition habe ich bei der Grundinstallation 35 GB zugewiesen, somit bleiben ungefähr 14,5 GB übrig. Diese möchte ich im Anschluss über die Kommandozeile konfigurieren.

```
C:\Users\Administrator>diskpart
Microsoft DiskPart-Version 6.3.9600
Copyright (C) 1999-2013 Microsoft Corporation.
Auf Computer: SRV2012R2CORE

DISKPART> list disk

Datenträger ### Status Größe Frei Dyn GPT
-----
Datenträger 0 Online 50 GB 14 GB
```

Um mit Festplatten auf der Kommandozeile arbeiten zu können, benutzt man das Programm **diskpart**.

```
DISKPART> list volume

Volume ### Bst Bezeichnung DS Typ Größe Status Info
-----
Volume 0 D System-reserviert CD 0 B Kein Medium
Volume 1 System-reserviert NTFS Partition 350 MB Fehlerfrei System
Volume 2 C NTFS Partition 34 GB Fehlerfrei Startpartition

DISKPART>
```

Um mir alle Geräte anzeigen zu lassen die am System angeschlossen sind, verwende ich den Befehl

### list volume

```
DISKPART> select volume 0
Volume 0 ist jetzt das gewählte Volume.

DISKPART> remove letter d
Der Laufwerksbuchstabe oder der Bereitstellungspunkt wurde erfolgreich entfernt.

DISKPART> assign letter e
Der Laufwerksbuchstabe oder der Bereitstellungspunkt wurde zugewiesen.

DISKPART> list volume

Volume ### Bst Bezeichnung DS Typ Größe Status Info
-----
* Volume 0 E System-reserviert CD 0 B Kein Medium
Volume 1 System-reserviert NTFS Partition 350 MB Fehlerfrei System
Volume 2 C NTFS Partition 34 GB Fehlerfrei Startpartition

DISKPART>
```

Wenn man mit Geräten/Festplatten/Partitionen arbeiten möchte, müssen diese vorab selektiert werden, diese werden durch einen kleinen vorangestellten Stern hervorgehoben.

In diesen Beispiel möchte ich gerne dem CD/DVD Laufwerk einen neuen Laufwerksbuchstaben zuweisen.

- ↳ **select volume 0**
- ↳ **remove letter d**
- ↳ **assign letter e**

```
DISKPART> select disk 0
Datenträger 0 ist jetzt der gewählte Datenträger.
DISKPART> list partition

Partition ## Typ           Größe  Offset
-----
Partition 1 Primär   350 MB 1024 KB
Partition 2 Primär   34 GB  351 MB

DISKPART> create partition primary
Die angegebene Partition wurde erfolgreich erstellt.
DISKPART> list partition

Partition ## Typ           Größe  Offset
-----
Partition 1 Primär   350 MB 1024 KB
Partition 2 Primär   34 GB  351 MB
* Partition 3 Primär   14 GB  35 GB

DISKPART> select partition 3
Partition 3 ist jetzt die gewählte Partition.
DISKPART> format fs=ntfs quick
100 Prozent bearbeitet
DiskPart hat das Volume erfolgreich formatiert.
DISKPART> assign letter d
Der Laufwerksbuchstabe oder der Bereitstellungspunkt wurde zugewiesen.
DISKPART> list volume

Volume ## Bst  Bezeichnung  DS  Typ           Größe  Status  Info
-----
Volume 0  E    System-res  NTFS Partition  350 MB Fehlerfre System
Volume 1  C    System-res  NTFS Partition  34 GB  Fehlerfre Startpar
* Volume 3  D    System-res  NTFS Partition  14 GB  Fehlerfre

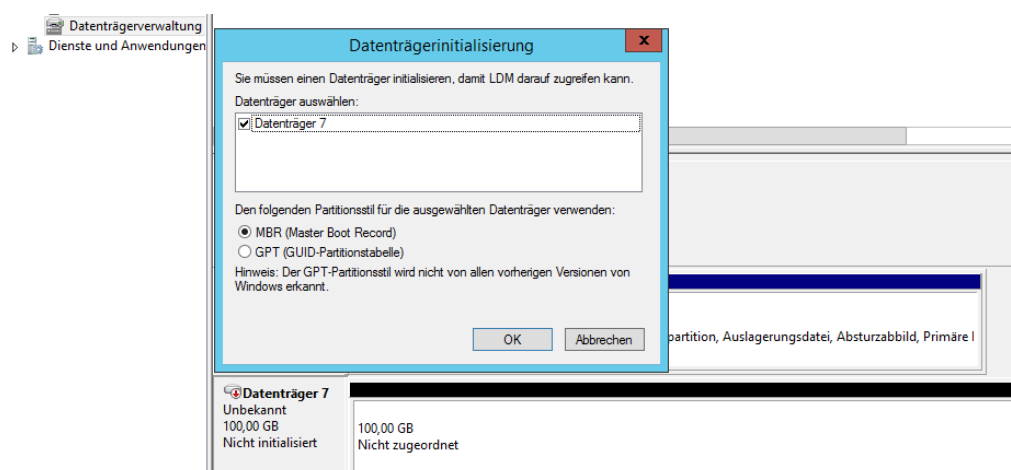
DISKPART>
```

Um nun eine neue Partition erstellen zu können zeige ich mir alle Festplatten an, danach wählt man die Festplatte aus und lässt sich alle Partitionen anzeigen, mit **create** erstellen wir eine neue Primäre Partition, wählen diese zur Sicherheit nochmal aus und formatieren diese, am Schluss setzen mir noch einen Laufwerksbuchstaben.

- ↳ **list disk**
- ↳ **select disk 0**
- ↳ **list partition**
- ↳ **create partition primary**
- ↳ **list partition**
- ↳ **select partition 3**
- ↳ **format fs=ntfs quick**
- ↳ **assign letter=d**
- ↳ **exit**

## Neue Festplatte dem Server hinzufügen

Wenn mal der Speicherplatz auf dem Core Server ausgeht, besonders in Virtuellen Maschinen, dann kann man jederzeit eine weitere Festplatte dem Server hinzufügen. Das würde in der Grafischen Oberfläche wie folgt aussehen. Um diese Festplatte zu konfigurieren kann man wieder die CMD Konsole wie auch PowerShell Konsole dafür einsetzen.

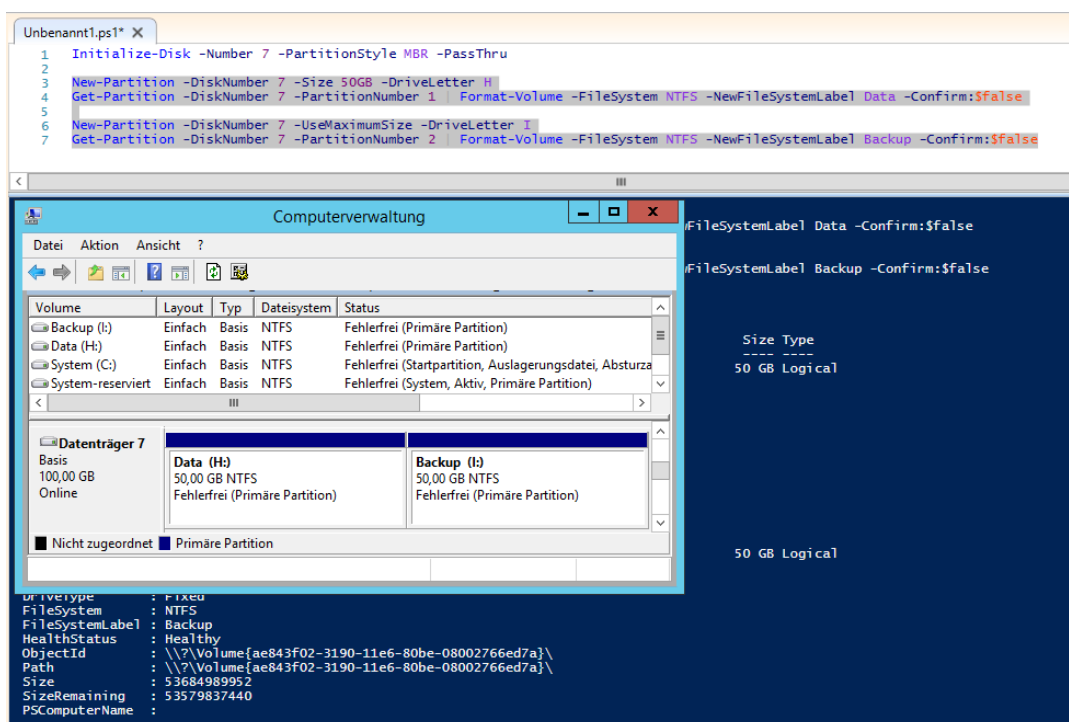


CMD:

- ↳ **list disk (Festplatten Nummer herausfinden)**
- ↳ **select disk x**
- ↳ **clean - (Dieser Befehl ist nur Notwendig, wenn man nochmal von vorne beginnen muss)**
- ↳ **convert mbr/gpt (Wenn man diesen Befehl absetzt wird die Festplatte initialisiert)**
  
- ↳ **create partition primary size=51200 (50GB)**
- ↳ **create partition primary (Rest ca.50 GB)**
  
- ↳ **list partition**
- ↳ **select partition 1**
- ↳ **format fs=ntfs label="Data" quick**
- ↳ **assign letter=h**
  
- ↳ **select partition 2**
- ↳ **format fs=ntfs label="Backup" quick**
- ↳ **assign letter=i**

PowerShell:

- ↳ **Get-Disk (Festplatten Nummer herausfinden)**
- ↳ **Get-Disk -Number 7**
- ↳ **Get-Partition -DiskNumber 7**
  
- ↳ **Get-Disk -Number 7 | Clear-Disk -RemoveData (Get-Disk -Number 7 | Clear-Disk -RemoveData -Confirm:\$false) - (Dieser Befehl ist nur Notwendig, wenn man nochmal von vorne beginnen muss)**
- ↳ **Initialize-Disk -Number 7 -PartitionStyle MBR/GPT -PassThru (Wenn man diesen Befehl absetzt wird die Festplatte initialisiert)**
  
- ↳ **New-Partition -DiskNumber 7 -Size 50GB -DriveLetter H**
- ↳ **Get-Partition -DiskNumber 7 -PartitionNumber 1 | Format-Volume -FileSystem NTFS -NewFileSystemLabel Data -Confirm:\$false**
  
- ↳ **New-Partition -DiskNumber 7 -UseMaximumSize -DriveLetter I**
- ↳ **Get-Partition -DiskNumber 7 -PartitionNumber 2 | Format-Volume -FileSystem NTFS -NewFileSystemLabel Backup -Confirm:\$false**





## Rollen und Features installieren

Um am Server Core Rollen und Features installieren zu können, gibt es auch wieder sehr viele Möglichkeiten, zum einen kann man Remote über einen zweiten Server mit GUI die Rollen und Features auf einen Server Core installieren oder natürlich direkt auf dem Server Core selbst mit **DISM** oder **Install-WindowsFeature** oder **DSC (Desired State Configuration)**. Einen großen Nachteil hat aber **DISM**, es benötigt sehr viel Recherche und Schreiarbeit, bis man die Serverrolle und alle passenden Feature zusammen geschrieben hat.

Siehe weiter unten, nach dem 6. Feature habe ich aufgehört, ansonsten würde nur die Installation vom IIS Webserver die ganze Seite ausfüllen. Für die Installation würde ich auf PowerShell setzen, da es hier die Schalter **-IncludeAllSubFeature -IncludeManagementTools** gibt, diese sparen sehr viel Tipparbeit.

### CMD

- ↳ **DISM /Online /Get-Features /Format:Table**
- ↳ **DISM /Online /Get-FeatureInfo /FeatureName:IIS-WebServerRole**
- ↳ **DISM /Online /Enable-Feature /?**
- ↳ **DISM /Online /Enable-Feature /FeatureName:IIS-WebServerRole /FeatureName:IIS-WebServer /FeatureName:IIS-CommonHttpFeatures /FeatureName:IIS-Security /FeatureName:IIS-RequestFiltering /FeatureName:IIS-StaticContent /FeatureName:IIS-DefaultDocument /FeatureName:usw. /LimitAccess /All**

### PowerShell

- ↳ **Get-WindowsFeature**
- ↳ **Get-Help Install-WindowsFeature -Examples**
- ↳ **Install-WindowsFeature -Name Web-Server -IncludeAllSubFeature -IncludeManagementTools -Source D:\sources\sxs**
- ↳ **Get-WindowsFeature | Where-Object {{\$\_.Name -like "Web-\*"} -and (\$\_.InstallState -eq "Installed")}**

## DSC - Desired State Configuration

Peter Kriegel hat zu diesem Thema ein sehr gutes Video verfasst, dies kann man sich bei YouTube ansehen.

↳ <https://www.youtube.com/watch?v=phSA--WWB1Y>

Meine Beispieldatei würde ungefähr wie folgt aussehen

```

DSC_ConfigFile_IIS.ps1 X
1  ## ----- Powershell Desired State Configuration ----- ##
2  ## create *.mof File = SERVERNAME -OutputPath $OutputPath -ComputerName SERVERNAME
3  ## start *.mof File = Start-DscConfiguration -Path $OutputPath -Wait -Verbose
4  ## ----- ##
5
6  ## ----- Default Script Variables ----- ##
7
8  $Computername = $env:COMPUTERNAME
9  $SoftwareSourcePath = "\\ads01.htdom.local\dsc$"
10 $OutputPath = "C:\PowershellScripts\DSC"
11 $LogPath = "C:\Logs"
12
13 ## ----- End Default Script Variables ----- ##
14
15 ## ----- Start Desired State Configuration - IIS01 ----- ##
16
17 Configuration IIS01
18 {
19     Node $Computername
20     {
21         # Create local Logs Folder
22         File CreateDirectoryLogs {
23             Ensure = "Present"
24             Type = "Directory"
25             DestinationPath = (Join-Path "$env:SystemDrive\" 'Logs')
26         }
27     }
28
29     # Install Windows Feature - NET-Framework-3.5
30     WindowsFeature NET-Framework-3.5
31     {
32         Ensure = "Present"
33         Name = "NET-Framework-Core"
34         Source = ($SoftwareSourcePath + "\sxs")
35         LogPath = ($LogPath + "\NET-Framework-3.5.Log")
36     }
37
38     # Install Windows Feature - Web-Server with some features
39     Foreach ($Feature in @("Web-Server", "Web-WebServer", "Web-Common-Http", "Web-Http-Errors", "Web-Default-Doc", "Web-Static-Content", "Web-Dir-Br
40     {
41         WindowsFeature "$Feature$Number"
42         {
43             Ensure = "Present"
44             Name = $Feature
45             LogPath = ($LogPath + "\Web-Server.Log")
46         }
47     }
48 }
49 }
50 }
51 }
52
53 ## ----- End Desired State Configuration - IIS01 ----- ##
  
```

Danach würde man das oben gezeigte Skript über die ISE laufen lassen und im Anschluss eine \*.mof Datei mit folgenden Befehl erstellen.

↳ **IIS01 -OutputPath \$OutputPath -ComputerName IIS01 (Gleiche Name wie Configuration)**

Diese \*.mof Datei würde man dann auf den passenden Webserver ausführen um den IIS mit sämtlichen Features zu installieren.

↳ **Start-DscConfiguration -Path \$OutputPath -Wait -Verbose**