

Windows

- Bootmanager
 - Bootmanager Reihenfolge festlegen
- CMD
 - Icons aus der Taskleiste entfernen
 - Namen und Geschwindigkeit einer Seriellen Schnittstelle ermitteln
 - Netzlaufwerk Ã¼ber CMD einbinden
 - Systemsteuerung Ã¼ber Kommandozeile Ã¶ffnen
 - Windows Bootstick - Installierbare Versionen einsehen
 - Windows Lizenz mit der Kommandozeile auslesen
- Einstellungen
 - Windows 8 / 10 / 11 StartmenÃ¼ sperren
 - Windows Scrollrichtung invertieren
 - Fingerabdrucksensor in Windows 10 / 11 aktivieren
- Kommandozeile
 - Benutzer / Computer SID auslesen
 - Partitionen mit Diskpart erstellen
- Powershell
 - Ausgaben mit PowerShell filtern
 - CSV mit Powershell einlesen
 - IPv4 Adresse mit PowerShell konfigurieren
 - Ping Ergebnis in CSV schreiben
 - Powershell Befehle Ã¼bersicht

- Powershell Portscanner
 - Uhrzeit und Datum formatiert in PowerShell ausgeben
 - Windows Lizenz aus Bios mit PowerShell auslesen
 - Windows Treiber über PowerShell installieren
 - Eingabesprache mit der PowerShell auf Deutsch stellen
 - Windows Serverrollen über die PowerShell installieren
 - Änderungsdatum mit PowerShell ändern
 - PowerShell Skript mit dem Windows Taskplaner automatisiert ausführen
 - Active Directory Computer mit Angabe des Betriebssystems mit der PowerShell ausgeben
- Tricks
 - Windows 10 Gottmodus aktivieren
 - Windows 10 / 11 im Auditmodus starten
 - Windows Netzwerkadapter einer VLAN ID zuweisen
- Windows 10
 - Websuche im Startmenü deaktivieren
 - Windows 11 Hardware Anforderungen umgehen
 - Bitlocker Status einsehen
- Windows 11
 - Windows 11 Installation - TPM Anforderung umgehen
- Windows Deployment Service
 - Treiber in WDS Startabbild integrieren
 - Migration auf einen neuen WDS-Server
- Windows Server
 - NTP Zeitserver Synchronisation auf einem Windows Server konfigurieren
 - Active Directory Kennwort Änderung über Web-GUI
 - Ordnerumleitung mit einer Gruppenrichtlinie konfigurieren
- Lizenzen
 - Aktivierungsserver von Microsoft

- Programme
 - Microsoft PowerToys installieren

Bootmanager

Bootmanager Reihenfolge festlegen

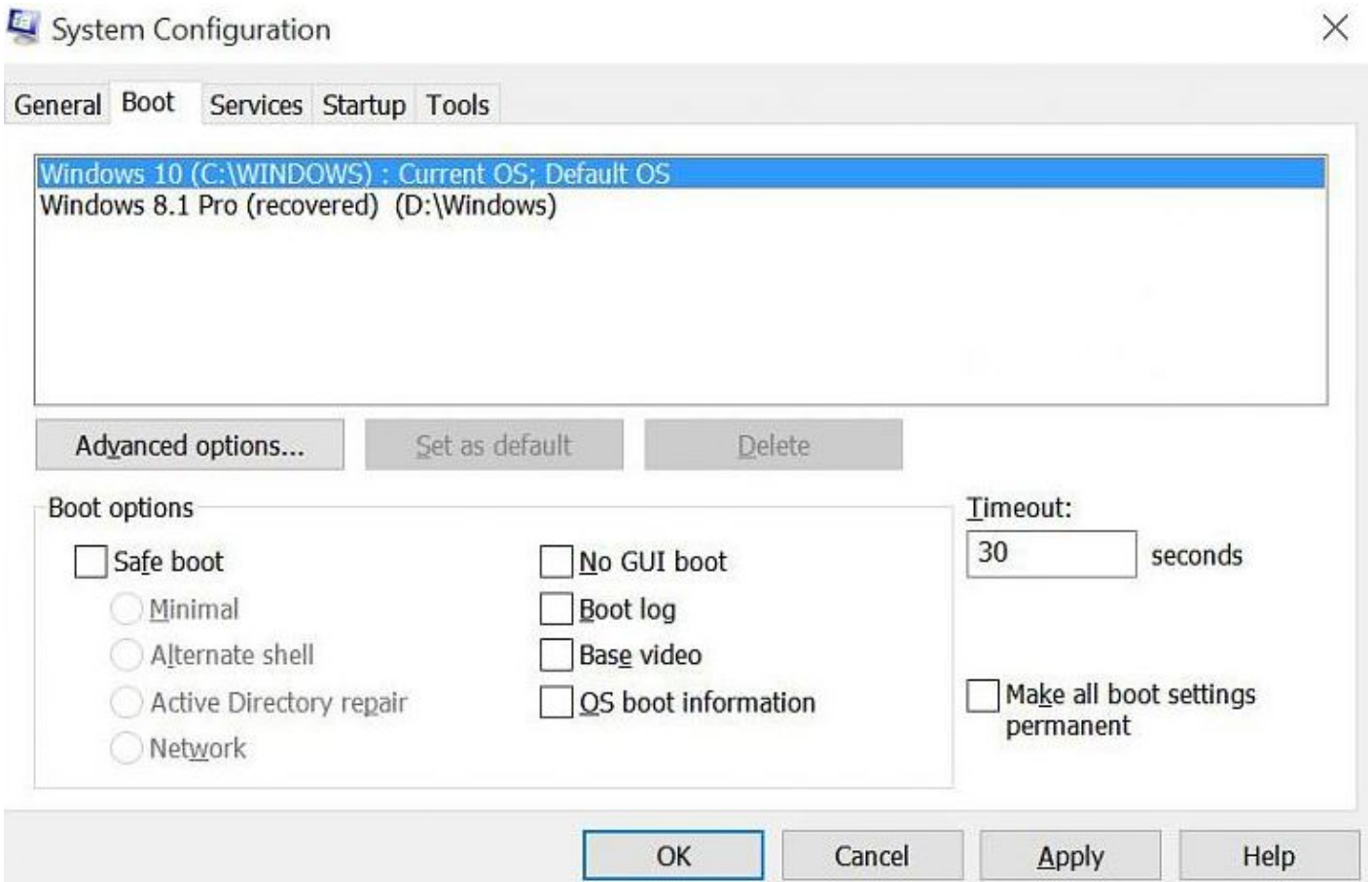
Einleitung

Manchmal haben wir auf unserem **Windows Computer** mehrere Betriebssysteme parallel installiert (Sprichwort: Dual Boot). Dies können mehrere Instanzen von Windows sein, als auch andere Betriebssysteme wie Linux oder Mac OS. In diesem Beitrag wollen wir einstellen, welches Betriebssystem Windows Standardgemäß startet.

Boot Reihenfolge einstellen

Um das entsprechende **Standard Betriebssystem** einzustellen, müssen wir im ersten Schritt die **Systemkonfiguration** öffnen. Um dies zu tun, öffnen wir das **Ausführen Fenster** in Windows und geben dort `msconfig` ein.

Unter dem Reiter **Boot** sehen wir alle derzeit installierten Betriebssysteme auf diesem Rechner. Wir müssen lediglich jetzt nur noch unser **Wunsch Startbetriebssystem** auswählen, und dann auf den Button *Als Standard festlegen / Set as default* klicken.



Sobald wir jetzt unseren Rechner neu starten, startet der Computer mit dem festgelegten Timeout.

Verzögerung festlegen

Im weiteren können wir auch eine **Verzögerung** festlegen. Diese Verzögerung ist der Zeitraum bis der Computer automatisch das ausgewählte Betriebssystem startet. Je höher diese Zahl ist, desto mehr Zeit haben wir das Betriebssystem auszuwählen welches gestartet werden soll.

Wir müssen lediglich dazu eine **Administrative Eingabeaufforderung** öffnen. Dort geben wir den nachstehenden Befehl ein und ersetzen **<Sekunden>** durch den Zeitraum, welchen wir festlegen möchten.

```
bcdedit /timeout <sekunden>
```

CMD

Icons aus der Taskleiste entfernen

Einleitung

Mit diesem Skript kannst du schnell alle Icons aus der Taskleiste entfernen. Du fügst dieses einfach in die Konsole ein, und dann werden die Icons gelöscht. Du solltest danach dich am besten einmal abmelden und dann wieder anmelden.

Code

```
DEL /F /S /Q /A "%AppData%\Microsoft\Internet Explorer\Quick Launch\User Pinned\TaskBar\*
```

Namen und Geschwindigkeit einer Seriellen Schnittstelle ermitteln

Mit diesem Skript kannst du den Namen und die Geschwindigkeit einer Seriellen Schnittstelle ermitteln.

Kopiere dir den Code in eine Text Datei und speichere diese in eine Datei mit der Endung **.bat**.

Im nächsten Schritt machst du einen Doppelklick drauf, du erhältst dann die Ausgabe mit der Geschwindigkeit.

```
echo off
for /f "tokens=3" %%a in (
    'REG QUERY HKLM\HARDWARE\DEVICEMAP\SERIALCOMM'
) do set "COMPORT=%%a"
echo COM-Port: %COMPORT%

for /f "tokens=2" %%a in (
    'MODE %COMPORT% ^| FIND /I "Baud"'
) do set "SPEED=%%a"
echo Geschwindigkeit: %SPEED% Bauds
pause
```

Netzlaufwerk Ã¼ber CMD einbinden

Einleitung

Wir werden in diesem Beispiel ein Netzlaufwerk Ã¼ber die CMD verbinden. Damit erscheint das Netzlaufwerk als "*weiterer DatentrÃ¤ger*" im Explorer. Dort kÃ¶nnen dann wie gewohnt von DatentrÃ¤gern, Daten abgelegt werden. Ein Zugriff von mehreren Personen auf eine Datei ist dann auch kein Problem mehr.

Anwendung

Wenn wir das Netzlaufwerk verbinden wollen, Ã¶ffnen wir die CMD. Dort verwenden wir folgenden Befehl um das Netzlaufwerk zu verbinden:

```
net use <laufwerks-buchstabe>: <unc-pfad>
```

Wenn wir diesen Befehl abgesetzt haben, erscheint das Netzlaufwerk nun im Explorer. Wir kÃ¶nnen den DatentrÃ¤ger Ã¶ffnen und sehen auch dort abgelegte Daten.

Laufwerk nicht permanent

Wenn wir nicht wollen das Windows bei einer Neuansmeldung das Netzlaufwerk behÃ¤lt, setzen wir den Parameter **persistent:no** hinten an den Befehl.

```
net use <laufwerks-buchstabe>: <unc-pfad> /persistent:no
```

Laufwerk permanent mit Login Informationen

Wir kÃ¶nnen dem Befehl auch mitteilen das das Netzlaufwerk permanent angebunden sein soll. Dazu kÃ¶nnen wir auch noch Login Informationen mitgeben damit der Nutzer sich automatisch auf dem Netzlaufwerk anmeldet wenn der Nutzer sich auf dem Computer anmeldet.

```
net use <laufwerks-buchstabe>: <unc-pfad> <password> /user:<benutzer> /savecred /persistent:yes
```

Angebundenes Laufwerk entfernen

Wenn wir jetzt ein Laufwerk wieder entfernen wollen, verwenden wir den Befehl **net use /delete**. Dann wird der Eintrag aus dem Explorer gelöscht.

```
net use <laufwerks-buchstabe>: /delete
```

Systemsteuerung über Kommandozeile öffnen

Einleitung

In diesem Beitrag gehe ich kurz darauf ein, wie wir mit der **Kommandozeile** die **Systemsteuerung** öffnen können.

Systemsteuerung öffnen

Um die **Systemsteuerung** zu öffnen, müssen wir im ersten Schritt die **Kommandozeile** öffnen. Dies machen wir, indem wir die Tastenkombination **[Windows-Taste] + R** und geben dort `cmd` ein. Wenn wir dies mit der **[Enter-Taste]** bestätigen, dann öffnet sich die **Kommandozeile**.

Dort müssen wir nur folgenden Befehl eingeben, damit die **Systemsteuerung** im Anschluss geöffnet wird.

```
start control
```

Windows Bootstick - Installierbare Versionen einsehen

Einleitung

Immer wieder trifft man auf alte USB Sticks die einen Windows Installer enthalten, aber man weiß nicht um welche Version es sich handelt. Man könnte jetzt den PC von diesem Stick aus booten und installieren. Aber um diesen Vorgang zu erleichtern gibt es ein Skript welches die Installierbaren Versionen in einem Terminal Fenster ausgibt.

Anwendung

Wir laden folgenden Code in eine Batch Datei ein. Den Namen können wir frei wählen. Ich verwende den Namen **ReadWim.bat**. Im nächsten Schritt öffnen wir das **Windows Ausführen Fenster** [Windows Taste + R]. Dort geben wir folgendes ein: shell:sendto. (Ordner: C:\Users\<Kontoname>\AppData\Roaming\Microsoft\Windows\SendTo) Es öffnet sich ein Ordner. Dort erstellst du eine Verknüpfung. Unter Eigenschaften legst du einen Pfad fest in der die Batch Datei liegt.

Wichtig: Unter den Eigenschaften der Verknüpfungen musst du unter **Verknüpfung => Erweitert** die Checkbox **Als Administrator ausführen** aktivieren!

```
%comspec% /c C:\skripte\ReadWim.bat
```

Im nächsten Schritt müssen wir einen Registry Eintrag hinterlegen. Das machen wir indem wir eine Datei erstellen mit dem Namen **addReg.reg**. Dort fügen wir folgenden Code ein

```
Windows Registry Editor Version 5.00
```

```
[HKEY_CLASSES_ROOT\Drive\shellex\ContextMenuHandlers\Sendto]
```

```
@="{7BA4C740-9E81-11CF-99D3-00AA004AE837}"
```

Diese Datei speicherst du ab, und machst dann einen Doppelklick drauf.

Code

```
@echo off
set workdir=%~dp0
whoami /groups | find "S-1-16-12288" > nul
if not errorlevel 1 (
    %windir%\System32\more +16 "%~f0" > "%workdir%%~n0-temp.ps1"
    powershell -NoProfile -ExecutionPolicy Bypass -File "%workdir%%~n0-temp.ps1" %*
    del %workdir%%~n0-temp.ps1
    exit /b
)
color 0C
echo.
echo Das Attribut -Als Administrator ausfuehren- wurde nicht gesetzt!
pause
goto :eof

$Details = $true
$Details = $false

function Ausgabe([string[]] $Pfade, [bool] $Details) {
    ForEach ($Pfad in $Pfade) {
        if (Test-Path -Path $Pfad) {
            Write-Host `n$Pfad -ForegroundColor Yellow -BackgroundColor Black
        }
    }
    if ($Details) {
        Write-Host "`n$Pfad nicht gefunden." -ForegroundColor Red -BackgroundColor Black
        continue
    }
}

$ISO = $Null
if (Test-Path -Path $Pfad -Include *.iso) {
    $ISO = $Pfad
    Mount-DiskImage $Pfad | Out-Null
    $Pfad = (Get-DiskImage $Pfad | Get-Volume).DriveLetter + ":\\"
}

if ($Details) {
```

```

if (Test-Path -Path $Pfad -Include *.swm,*.esd,*.wim) {
    $Index = (Get-WindowedImage -ImagePath $Pfad).ImageIndex
    ForEach ($Nnummer in $Index) {
        Dism.exe /Get-ImageInfo /Imagefile:$Pfad /index:$Nnummer
    }
}
else {
    $Wims = Get-ChildItem -Path $Pfad -include *.wim,*.esd,*.swm -Recurse
    ForEach ($Pfad in $Wims) {
        $Index = (Get-WindowedImage -ImagePath $Pfad).ImageIndex
        ForEach ($Nnummer in $Index) {
            Dism.exe /Get-ImageInfo /Imagefile:$Pfad /index:$Nnummer
        }
    }
}
else {
    if (Test-Path -Path $Pfad -Include *.swm,*.esd,*.wim) {
        Dism.exe /Get-ImageInfo /Imagefile:$Pfad
    }
    else {
        foreach ($wim in (Get-ChildItem -path $Pfad -Include *.wim,*.esd,*.swm -Recurse)) {
            Dism.exe /Get-ImageInfo /Imagefile:$wim
        }
    }
}

If ($ISO) {
    Dismount-DiskImage $ISO | Out-Null
}

$Dateien = @()
ForEach ($arg in $Args) {
    if ($arg -eq "-v" -or $arg -eq "/v" -or $arg -eq "--verbose") {
        $Details = $true
    }
    else {
        $Dateien += $arg
    }
}

```

```
}
```

```
}
```

Ausgabe \$Dateien \$Details

Write-Host "`r`n`r`n(Enter zum Beenden)" -NoNewline

Read-Host

Quellenangabe: Skript & Anleitung stammt von <https://heise.de>. Diese Anleitung ist hier nur eine gekürzte Fassung.

Original-Artikel: <https://www.heise.de/ratgeber/Anleitung-Mit-Skript-Windows-Setup-Medium-analysieren-6550233.html>

Windows Lizenz mit der Kommandozeile auslesen

Einleitung

Bei einer Neuinstallation eines Rechners, kann es mal dazu kommen, dass der Lizenzschlüssel der im **BIOS** hinterlegt ist, nicht in das Betriebssystem eingefügt wird. Diesen Schlüssel können wir mit der **PowerShell** oder der **Kommandozeile** auslesen. In diesem Beitrag gehen wir drauf ein, wie wir den Lizenzschlüssel mit der **Kommandozeile** auslesen.

Lizenzschlüssel auslesen

Um jetzt den Lizenzschlüssel aus dem **BIOS** auszulesen, müssen wir im ersten Schritt die **Kommandozeile** mit **administrativen Berechtigungen** öffnen. Sobald wir das getan haben, müssen wir nur den nachfolgenden Befehl eingeben. Wir erhalten dann in der Ausgabe den Lizenzkey, welchen wir dann in das Betriebssystem einpflegen können.

```
wmic path softwarelicensingervice get OA3xOriginalProductKey
```

Einstellungen

Windows 8 / 10 / 11

Startmenü sperren

Einleitung

Falls du in Windows 8 / 10 / 11 das Startmenü sperren möchtest, das bedeutet das keine Änderungen mehr gemacht werden können, kannst du dies über Gruppenrichtlinien oder einen Registry Eintrag bewerkstelligen.

Dies ist praktisch wenn z.B. ein Standardisierter Desktop / Startmenü eingerichtet werden soll und niemand was Ändern soll damit es für jeden die selbe Anordnung hat.

Durchführung

Im ersten Schritt musst du dir ein Startmenülayout überlegen. Du pinnst die benötigten Dateien, Ordner und Programme an das Startmenü. Dieses lässt sich dann später nicht mehr verändern solange diese Regel aktiv ist.

Dateien und Ordner anheften

Wenn du Dateien, Ordner oder eigene Verknüpfungen anheften möchtest, wirst du feststellen das du den Punkt **"An Start anheften"** nicht finden kannst.

Dafür müssen wir den Speicherort der Dateien verändern, so dass diese dann als *installierte Programme* auftauchen. Diese kannst du dann anheften.

Du kopierst die Dateien dann an folgenden Pfad:

```
C:\Users\%USERNAME%\AppData\Roaming\Microsoft\Windows\Start Menu
```

Sobald diese dort abgelegt sind, findest du unter diesem Benutzer die Dateien und Ordner in der Programmliste. Diese kannst du dann jetzt wie gewohnt an Start anheften.

Anordnung über Gruppenrichtlinien sperren

Wenn du jetzt das Startmenü über die Gruppenrichtlinien sperren möchtest, musst du zuerst das Layout als **XML Datei** exportieren. Dafür öffnest du die PowerShell und gibst folgenden Befehl ein:

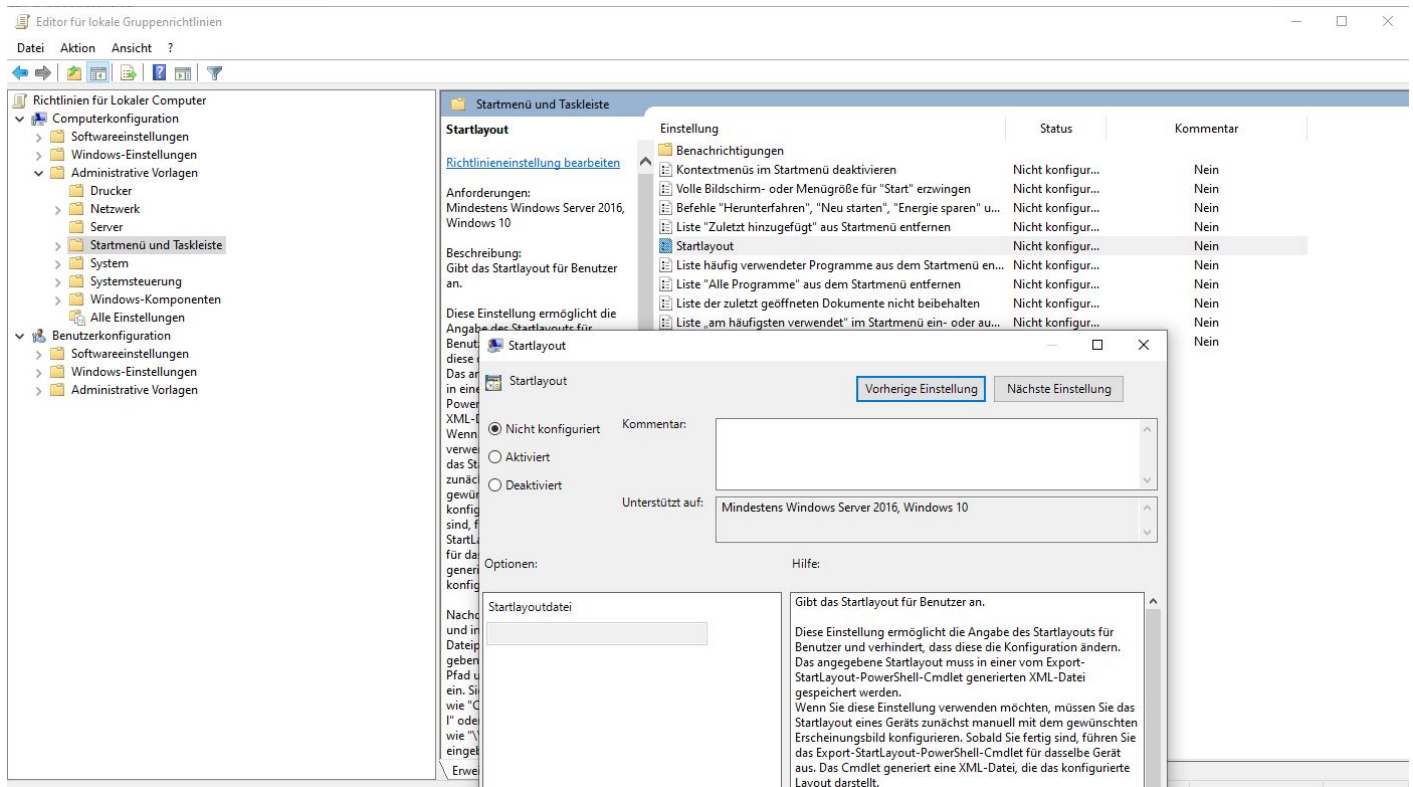
```
Export-StartLayout -As XML -Path ($env:HOMEPath + "\Desktop\startlayout.xml")
```

Die Datei die jetzt auf deinem Desktop liegt, kannst du lokal ablegen oder über einen UNC Pfad auf einem Server. Wichtig ist das der Benutzer Leserechte auf diesen Pfad hat.

Du kannst dann unter folgendem Pfad die Gruppenrichtlinie erstellen:

Administrative Vorlagen / Startmenü

Dort aktivierst du dann die Regel und gibst dort den Pfad zur XML Datei an. Sobald die Regel aktiv ist, kann das Startmenü nicht mehr verändert werden.



Anordnung über Registry Eintrag sperren

Das selbe Ergebnis kannst du auch mit dem Registrierungseditor erzielen. Hier musst du im Vergleich zur Gruppenrichtlinie keine XML Datei exportieren. Mithilfe der XML kannst du das Startlayout aber wieder importieren. Dafür kannst du das nachstehende PowerShell Skript nutzen.

```
Import-StartLayout -LayoutPath <pfad>
```

Hier können wir unterscheiden ob wir das Layout Geräteweise oder Benutzerspezifisch sperren wollen.

Zuerst öffnen wir den Registrierungseditor indem wir die Tastenkombination **[Windows-Taste] + [R]** drücken. Dort geben wir dann das Wort **regedit** ein. Es öffnet sich dann der

Registrierungseditor.

Wenn wir das Layout **Geräteweit** sperren wollen, navigieren wir zu folgendem Pfad:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\
```

Wenn wir das Layout **Benutzerspezifisch** sperren wollen, navigieren wir zu folgendem Pfad:

```
HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\Explorer
```

Falls ein Ordner nicht existiert kannst du einen erstellen durch einen Rechtsklick auf den letzten Ordner und erstellen einen neuen **Schlüssel** mit dem entsprechenden Namen.

Dort erstellen wir einen **DWORD-Eintrag (32-bit)** mit dem Namen **LockedStartLayout**. Und ändern den Wert auf **1**. Wenn du dich dann einmal abmeldest, und dann wieder anmeldest sind die Änderungen übernommen und du kannst keine Änderungen mehr vornehmen.

Wenn du am Layout wieder etwas verändern möchtest, setzt du den Wert einfach auf **0**.

Windows Scrollrichtung invertieren

Einleitung

In diesem Beitrag gehe ich kurz darauf ein, wie wir unter **Windows 10 / Windows 11** die **Scrollrichtung** des **Mausrads** einstellen können. Über diese Funktion werden sich insbesondere viele *Mac Benutzer* freuen!

Scrollrichtung anpassen

Um die **Scrollrichtung** anzupassen, müssen wir einen Eintrag in der **Registry** ändern. Dazu müssen wir im ersten Schritt den **Registrierungseditor** öffnen. Dies können wir erledigen, indem wir "Ausführen" öffnen, und dort dann `regedit` eingeben.

In dem **Registrierungseditor** müssen wir folgendem Pfad folgen:

```
HKEY_LOCAL_MACHINE / SYSTEM / CurrentControlSet / Enum / HID
```

Dort müssen wir die einzelnen Unterordner nach dem Eintrag **FlipFlopWheel** suchen, und den Wert auf **1** ändern. Nachdem wir jetzt den Rechner neu gestartet haben, verhält sich das Mausrad jetzt entsprechend anders.

Fingerabdrucksensor in Windows 10 / 11 aktivieren

Einleitung

In diesem Beitrag erläutere ich kurz, wie wir den **Login per Fingerabdrucksensor** aktivieren können. Bei einigen Geräten, die sich in einer **Active Directory Domäne** befinden, kann das Problem auftauchen, dass die "*Registrierung des Fingers*" nicht möglich ist.

Die Fehlermeldung kann wie folgt aussehen:

Anmeldung per Fingerabdruck ist derzeit nicht verfügbar!

Fingerabdrucksensor aktivieren

Gruppenrichtlinien erstellen

Im ersten Schritt müssen wir sicherstellen, dass wir den **Login per Fingerabdruck** über die **Gruppenrichtlinien (GPO)** aktiviert haben. Dazu müssen wir 3 Richtlinien eingestellt haben.

Computerkonfiguration / Richtlinien / Administrative Vorlagen / Windows-Komponenten / Biometrie /
Verwendung von Biometrie zulassen => Aktivieren

Computerkonfiguration / Richtlinien / Administrative Vorlagen / Windows-Komponenten / Biometrie /
Benutzeranmeldung mithilfe von Biometrie zulassen => Aktivieren

Computerkonfiguration / Richtlinien / Administrative Vorlagen / Windows-Komponenten / Biometrie /
Domänenbenutzeranmeldung mithilfe von Biometrie zulassen => Aktivieren

Da bei der "*Finger Registrierung*" noch eine **PIN** erstellt werden muss, wird noch eine **vierte Gruppenrichtlinie** benötigt.

Computerkonfiguration / Richtlinien / Administrative Vorlagen / System / Anmelden / PIN-Anmeldung aktivieren
=> Aktivieren

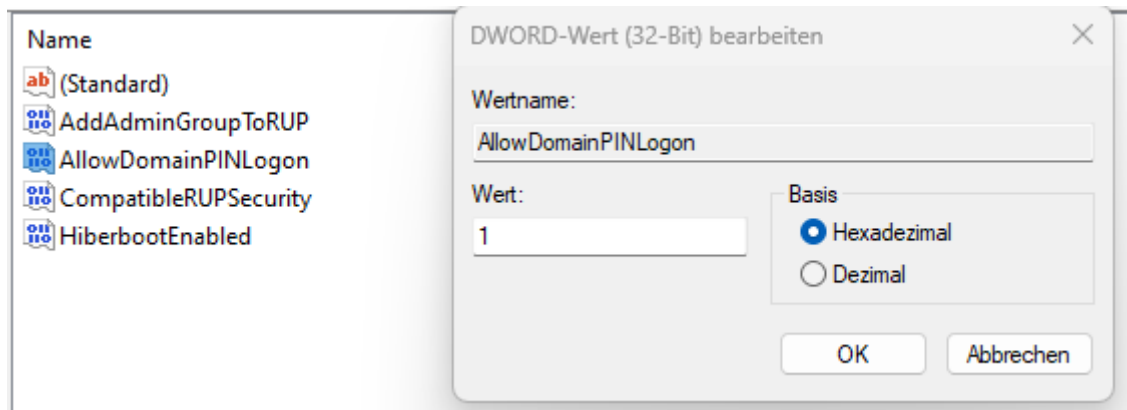
Registryeintrag hinzufügen

Wenn nach einer erfolgreichen **Gruppenrichtlinien-Aktualisierung** die Option immer noch deaktiviert ist, müssen wir in der Registry einen neuen Eintrag erstellen. Dazu navigieren wir zu folgendem Pfad:

```
Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\System
```

Dort erstellen wir einen neuen **DWORD-Wert (32 Bit)**.

Diesen Eintrag benennen wir **AllowDomainPINLogon** und ändern den Wert auf **Hexadezimal 1**.



Als letzten Schritt müssen wir jetzt nur noch den PC neu starten. Im Anschluss kann der Fingerabdruck hinterlegt werden, um sich damit in Zukunft anzumelden.

Kommandozeile

Benutzer / Computer SID auslesen

Wenn du die **SID** auslesen möchtest, öffnest du in Windows die **Befehlszeile** als **Administrator**. Dort gibst du folgenden Befehl ein:

```
wmic useraccount get sid
```

Du erhältst dann eine Reihe von Zahlen. Der Teilstring, der identisch ist, ist die **Computer-SID**. Die restlichen 3 Zahlen sind die **Benutzer-SID**.

Wenn wir den dazugehörigen Benutzernamen angezeigt bekommen wollen, verwenden wir den nächsten Befehl.

```
wmic useraccount get name,sid
```

Partitionen mit Diskpart erstellen

Einleitung

Wenn wir Festplatten oder andere Partitionen verÄndern oder erstellen wollen kÄnnen wir in Windows das Tool **Diskpart** nutzen. Dies ermÄglicht uns, Festplatten und andere Speichermedien zu partitionieren, lÄschen und zu bearbeiten.

Anwendung

Zuerst mÄssen wir unseren USB Stick oder die Festplatte mit unserem PC / Computer verbinden. Wenn dies erledigt ist, mÄssen wir jetzt die **Kommandozeile** Äffnen. Die **Kommandozeile** kÄnnen wir Äber **AusfÄhren** und dann durch das eingeben von **cmd**. Wir kÄnnen aber auch Äber **Start** die Eingabeaufforderung Äffnen. Dort geben wir den folgenden Befehl ein um **diskpart** zu starten:

```
diskpart
```

Wenn wir jetzt den folgenden Befehl eingeben, erhalten wir alle angeschlossenen Speichermedien:

```
list disk
```

```
Microsoft DiskPart-Version 10.0.22000.653

Copyright (C) Microsoft Corporation.
Auf Computer: NOTEBOOK-PHILLI

DISKPART> list disk

  DatentrÄger ###  Status              GrÖÙe   Frei    Dyn  GPT
  -----
  DatentrÄger 0     Online              931 GB  2048 KB      *
  DatentrÄger 1     Online             7651 MB      0 B
```

Mit dem Befehl **select disk** wÄhlen wir das Speichermedium aus welches wir verÄndern wollen. Wir erkennen das Speichermedium an der GrÄÙÄÙe. Bei uns ist das **DatentrÄger 1**.

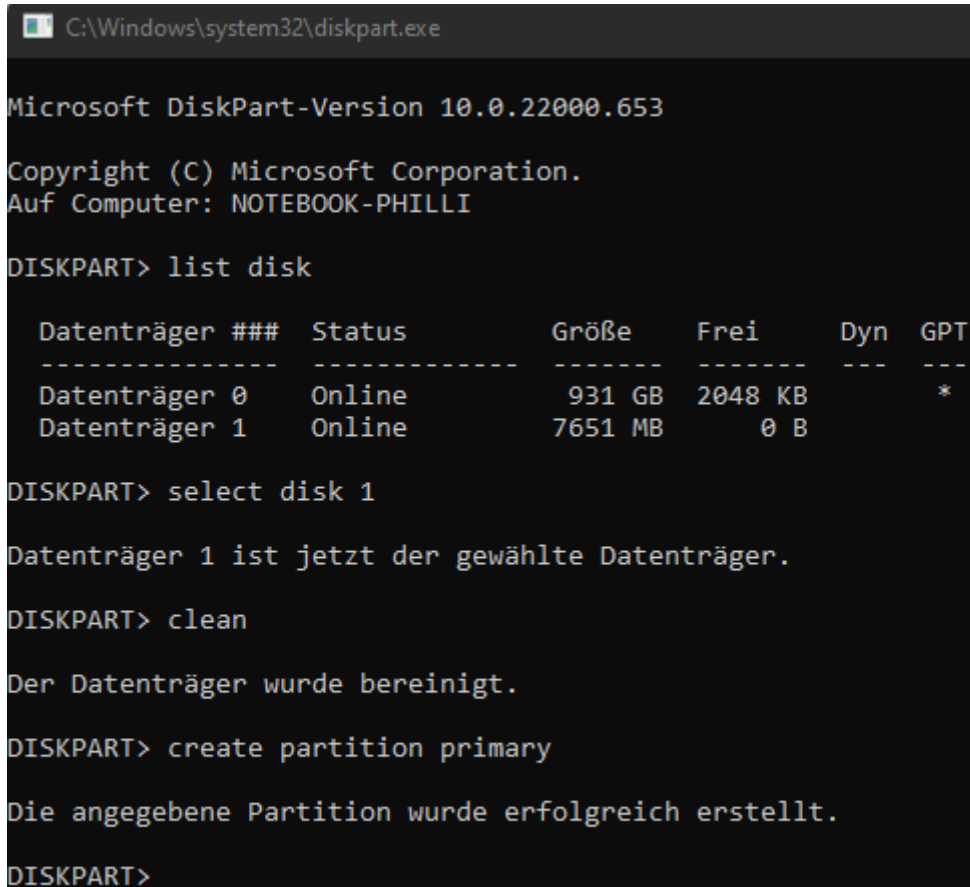
```
select disk <nummer>
```

Jetzt löschen wir den Inhalt der Partition.

```
clean
```

Im nächsten Schritt erstellen wir die primäre Partition. Diese benötigt Windows um den Stick zu starten. Das Format wird später festgelegt. Wir erstellen jetzt lediglich die Partition.

```
create partition primary
```



```
C:\Windows\system32\diskpart.exe

Microsoft DiskPart-Version 10.0.22000.653

Copyright (C) Microsoft Corporation.
Auf Computer: NOTEBOOK-PHILLI

DISKPART> list disk

   Datenträger ###  Status              Größe   Frei    Dyn  GPT
   -----
   Datenträger 0    Online              931 GB  2048 KB
   Datenträger 1    Online             7651 MB    0 B

DISKPART> select disk 1

Datenträger 1 ist jetzt der gewählte Datenträger.

DISKPART> clean

Der Datenträger wurde bereinigt.

DISKPART> create partition primary

Die angegebene Partition wurde erfolgreich erstellt.

DISKPART>
```

Jetzt formatieren wir den Stick und weisen dem ein Partitionsformat zu.

```
format fs=fat32 quick
```

Wir können auch eine Sektorgröße mitgeben. Dazu fügen wir den Parameter **A=<größe>** hinzu. Wenn dieser nicht angegeben wird, verwendet er automatisch die möglichst große Größe anhand der Kapazität des Speichermediums.

```
format fs=fat32 a=<größe> quick
```

Dateisysteme	
Dateisystem	Dateisystemeinheitengröße
FAT	512, 1024, 2048, 4096, 8192, 16k, 32, 64k
FAT32	512, 1024, 2048, 4096, 8192, 16k, 32, 64k
NFTS	512, 1024, 2048, 4096, 8192, 16k, 32k, 64k, 128k, 256k, 512k, 1M, 2M
exFAT	512, 1024, 2048, 4096, 8192, 16k, 32k, 64k, 128k, 256k, 512k, 1M, 2M, 4M, 8M, 16M, 32M
ReFS	4096, 64K
UDF	

Als letztes verwenden wir den Befehl **assign** damit Windows das Laufwerk einbindet und einen Buchstaben zuweist.

```
assign
```

Um jetzt die Partitionierung zu beenden setzen wir den Befehl **exit** ab, dann können wir dem Speichermedium Daten hinzufügen oder entfernen.

```
exit
```

Powershell

Ausgaben mit PowerShell filtern

Einleitung

Wenn du z.B. große Log Dateien durchforsten möchtest, ist das filtern nach bestimmten Wörtern oder regulären Ausdrücken sehr hilfreich. Dies kannst du mit der PowerShell realisieren.

Anwendung

Du musst im ersten Schritt die Datei einlesen, dann können diese Daten weiter verarbeitet werden.

```
Get-Content "C:\temp\log.txt"
```

Diese übergeben wir mit der Pipeline Variable an die Zeilenweise Ausgabe der Datei.

```
Get-Content "C:\temp\log.txt" | Out-String -Stream
```

Im letzten Schritt filterst du mithilfe von `Select-String` die Ausgaben. Dafür verwenden wir ein paar Argumente die alle verwendet werden **können**.

Zuerst gibt es das Argument **-Pattern**. Mit diesem gibst du den Text an nachdem gesucht werden soll. Es können hierbei auch reguläre Ausdrücke verwendet werden.

Als zweites Argument gibt es noch **-SimpleMatch**. Damit teilst du der PowerShell mit das es sich **nicht** um einen regulären Ausdruck handelt.

Als letztes gibt es noch **-CaseSensitive**. Hier wird dann auch auf Groß- und Kleinschreibung geachtet.

Wir suchen in unserem Beispiel nach allen Einträgen die das Wort "Hardware" enthalten.

Ein Wert nach dem gesucht werden soll

```
Get-Content "C:\temp\log.txt" | Out-String -Stream | Select-String -Pattern "Hardware" -SimpleMatch
```

Mehrere Werte nach dem gesucht werden soll

```
Get-Content "C:\temp\log.txt" | Out-String -Stream | Select-String -Pattern "Hardware","Software","Netzwerk" - SimpleMatch
```

Du erhältst nun gefilterte Ausgaben von PowerShell.

CSV mit Powershell einlesen

Einleitung

Du kannst mit der Powershell CSV-Dateien einlesen. Dies kannst du verwenden um Listen von Daten einzulesen und diese z.B. als Parameter für Funktionen zu verwenden. So können automatisierte oder vereinfachte Prozesse gestaltet werden.

Code

Zuerst müssen wir eine CSV Datei erstellen. Im Beispiel erstellen wir eine Liste aus Rechnern die später eingelesen und angepingt werden sollen. Über die Überschriften können die Daten dann später angesprochen werden.

	A	B	C	D	
1	ID	PC-Name	IPv4-Adresse	Abteilung	
2	1	PC-1	192.168.75.1	Einkauf	
3	2	PC-2	192.168.75.2	Verkauf	
4	3	PC-3	192.168.75.3	Verkauf	
5	4	PC-4	192.168.75.4	Marketing	
6	5	PC-5	192.168.75.5	Marketing	
7	6	PC-6	192.168.75.6	IT	
8	7	PC-7	192.168.75.7	IT	
9	8	PC-8	192.168.75.8	IT	
10	9	PC-9	192.168.75.9	Geschäftsleitung	
11	10	PC-10	192.168.75.10	Geschäftsleitung	
12					
13					

Im nächsten Schritt erstellen wir in Powershell ein **Array** in der die CSV Daten gespeichert werden. Dort wird dann über die Powershell-Funktion **Import-CSV** die Datei eingelesen. Dort geben wir als Parameter den Pfad und das Trennzeichen mit. Die Daten werden dann als **mehrdimensionales Array** in das **Array** übergeben. In der folgenden Schleife wird dann das **Array** durchgearbeitet. Über die Überschriften können dann die Daten angesprochen werden.

```
# Import der CSV Daten in das Array
$Daten = @()
$Daten = Import-Csv -Path "C:\Temp\computer.csv" -Delimiter ";"

# Schleife die Daten der CSV nacheinander durcharbeitet
foreach($d In $Daten)
```

```
{
# Code der ausgeführt wird
if(Test-Connection -ComputerName $d.'IPv4-Adresse')
{
    Write-Host("Der Computer mit dem Namen " + $d.'PC-Name' + " ist erreichbar!") -ForegroundColor Green
} else {
    Write-Host("Der Computer mit dem Namen " + $d.'PC-Name' + " ist nicht erreichbar!") -ForegroundColor
Red
}
}
```

IPv4 Adresse mit PowerShell konfigurieren

Einleitung

Mit der Hilfe von der PowerShell kannst du deiner Schnittstelle eine IP-Adresse zuweisen. Dies kann nützlich sein wenn du z.B. öfters die IP-Adresse wechseln musst.

Hier erfährst du wie du ein Skript schreibst in der du deine IP-Adresse schneller wechseln kannst als andere die über die GUI arbeiten.

Anwendung

Schnittstellen Namen ermitteln

Im ersten Schritt müssen wir den Namen oder den Index der Schnittstelle ermitteln. Dies machst du indem du dein PowerShell Fenster öffnest und folgenden Befehl in die Konsole eingibst.

```
Get-NetAdapter
```

Dort werden jetzt Schnittstellen mit der Interface ID oder dem Interface Namen angezeigt. Dort suchst du das entsprechende Interface heraus, der du deine Statische IP-Adresse geben möchtest.

Du suchst dir dort aus, ob du mit dem **Interface Namen** oder der **Interface ID** arbeiten möchtest.

IP-Adresse Ändern

Hier werden beide Wege beschrieben wie du die IP-Adresse veränderst. Abhängig davon ob du den Namen oder die ID verwenden willst.

Interface Namen

```
New-NetIPAddress -IPAddress <ipv4-adresse> -AddressFamily IPv4 -InterfaceAlias <interface-name> -  
DefaultGateway <gateway> -PrefixLength <prefix-länge>
```

Interface ID

```
New-NetIPAddress -IPAddress <ipv4-adresse> -AddressFamily IPv4 -InterfaceIndex <interface-id> -  
DefaultGateway <gateway> -PrefixLength <prefix-lnge>
```

Wenn du jetzt den Befehl `ipconfig` in die CMD eingibst, siehst du das du die neue IP-Adresse gesetzt hast.

Ping Ergebnis in CSV schreiben

Einleitung

Mit diesem kleinen Skript kann man in eine CSV Datei schreiben lassen, ob ein Ping erfolgreich durchgegangen ist oder nicht. Es wird dann immer die Uhrzeit und das Datum dazu geschrieben um nachzuverfolgen wann das genau war.

Du kannst in dem Skript festlegen in welchen Abschnitten die Pings durchgeführt werden, wie viele Durchläufe durchgeführt werden sollen, und unter welchem Pfad die Log Datei liegen soll und wie diese heißt.

Die Datei wird bei jeder Durchführung gelöscht und neu erstellt damit keine Altlasten zur Verwirrung sorgen können.

Code

```
$path = "C:\logs"
$filename = "result.csv"
$destAddress = "srv-windows1"

Write-Host("=====") -ForegroundColor Green
Write-Host("Starte Connection Tester") -ForegroundColor Green
Write-Host("Ziel: " + $zielAdresse) -ForegroundColor Green
Write-Host("Entwickelt von: Phillip Unzen") -ForegroundColor Green
Write-Host("=====") -ForegroundColor Green
Write-Host(" ")

if(Test-Path -Path ($path + "\" + $filename))
{
    Remove-Item -Path ($path + "\" + $filename)
    New-Item -Path $path -Name $filename -ItemType File
} else {
    New-Item -Path $path -Name $filename -ItemType File
```

```
}

for($i = 0; $i -le 10000; $i++)
{
    if(Test-Connection -ComputerName $destAddress -Count 1)
    {
        Write-Host("Ping erfolgreich!") -ForegroundColor Green
        $str = (Get-Date).ToString("dd.MM.yyyy") + ";" + (Get-Date).ToString("HH:mm:ss") + ";" + "Erfolgreich"

    } else {
        Write-Host("Keine Verbindung") -ForegroundColor Red
        $str = (Get-Date).ToString("dd.MM.yyyy") + ";" + (Get-Date).ToString("HH:mm:ss") + ";" + "Fehler"
    }

    $str | Out-File -FilePath ($path + "\" + $filename) -Encoding utf8 -Append

    Start-Sleep 2
}
```

Powershell Befehle

Übersicht

Einleitung

Hier findest du einige hilfreiche Befehle die für das Powershell Skripten wichtig sein könnten. Kann als Nachschlagewerk manchmal ganz praktisch sein. Die Liste ist nicht sortiert, da nach und nach neue Einträge hinzugefügt werden.

Übersicht

Befehl	Funktion
Start-Sleep <sekunden>	Startet einen Timeout. Code wird fortgeführt wenn die Sekunden abgelaufen sind.
Test-Path -path <Pfad mit Datei / Pfad des Ordners>	Überprüft ob eine Datei oder ein Ordner vorhanden ist und gibt true oder false zurück.
Remove-Item -path <Pfad mit Datei / Pfad des Ordners>	Löscht Datei/en oder Ordner.
New-Item -path <Pfad> -name <Name> -ItemType <Typ> (Directory File)	Legt neue Ordner oder Dateien an.
Get-Date	Aktuelles Datum mit Uhrzeit erhalten
Get-Random	Zufallszahl ausgeben

(Liste wird weitergeführt...)

Powershell Portscanner

Einleitung

Hier findest du einen Portscanner der komplett in PowerShell geschrieben ist. Das heißt, du kannst diesen auf jedem Windows-Rechner ohne weitere Software Installationen ausführen.

Dieser ist dafür, dass er in PowerShell geschrieben ist, leider nicht so schnell wie die üblichen Portscanner. Trotzdem funktioniert er gut!

Du musst bei dem Skript nur die IP-Adresse des zu scannenden Hosts angeben, und den Port Bereich. Das Skript beschränkt sich auf den **Well Known Ports** Bereich (1 - 1023).

Code

```
$ipv4Address = "192.168.1.35"

for($i = 1; $i -lt 1024; $i++)
{
    if(Test-NetConnection $ipv4Address -InformationLevel Quiet -Port $i)
    {
        Write-Host("Port $i ist offen") -ForegroundColor Green
    }
}
```

Uhrzeit und Datum formatiert in PowerShell ausgeben

Einleitung

In PowerShell kannst du das aktuelle Datum oder die aktuelle Uhrzeit zurÃ¼ckbekommen um mit dieser weiter zu arbeiten oder andere Dinge auszugeben.

Hier findest du die Code Beispiele wie du zum Beispiel nur das Datum, oder nur die Uhrzeit erhalten kannst.

Code

Uhrzeit und Datum erhalten

```
Get-Date
```

Uhrzeit erhalten (24-Stunden Format)

```
(Get-Date).ToString("HH:mm:ss")
```

Uhrzeit erhalten (12-Stunden Format)

```
(Get-Date).ToString("hh:mm:ss")
```

Uhrzeit erhalten (ohne führende Nullen)

```
(Get-Date).ToString("h:m:s")
```

Tage / Monate zu einem Datum hinzufügen und dieses ausgeben

```
(Get-Date).AddDays(5).AddHours(3).ToString("dd.MM.yyyy_HH:mm")
```


Windows Lizenz aus Bios mit PowerShell auslesen

Einleitung

Bei manch einer Neuinstallation von Windows, kann es dazu kommen, dass der Lizenz Key aus dem BIOS nicht erkannt und eingegeführt wird. Um den Lizenzkey dann in dem Betriebssystem einzugeben, brauchen wir diesen erst einmal. Dafür gibt es diverse Tools, oder wir nutzen die PowerShell, um mit Bordmitteln diesen Lizenzkey auszulesen.

Windows Key auslesen

Um den Lizenzkey aus dem BIOS auszulesen, müssen wir im ersten Schritt die **PowerShell** mit **administrativen Berechtigungen** öffnen. Dort angekommen, müssen wir nur den nachstehenden Code eingeben. Sobald wir den Befehl mit einem **Enter** abschicken, wird uns der Lizenzkey in der Ausgabe angezeigt.

```
(Get-CimInstance -Query "select * from SoftwareLicensingService").OA3xOriginalProductKey
```

Windows Treiber &ber PowerShell installieren

Einleitung

Sobald wir einen Windows-Client oder Server installieren, müssen wir Treiber installieren. In diesem Beitrag gehe ich kurz darauf ein, wie wir die Treiber &ber die **PowerShell** installieren können.

Treiber installieren

Im ersten Schritt müssen wir mit `cd` in den Ordner wechseln, in dem die Treiber liegen.

```
cd D:
```

Dort angekommen, geben wir den folgenden Befehl ein. Mit diesem Befehl wird der Ordner **rekursiv** gescannt und die benötigten Treiber daraus installiert.

```
Get-ChildItem -Filter *.inf -Recurse | Select-Object FullName | ForEach-Object {pnputil -a $_.FullName}
```

Sobald das Skript durchgelaufen ist, haben wir alle **Treiber** installiert.

Eingabesprache mit der PowerShell auf Deutsch stellen

Einleitung

In diesem Beitrag erläutere ich kurz, wie wir mit der **PowerShell** die Eingabesprache unseres **Windows Systems** auf Deutsch stellen können. Dazu ist lediglich nur ein Befehl nötig.

Eingabesprache umstellen

Um jetzt die **Eingabesprache** auf **Deutsch** zu stellen, müssen wir im ersten Schritt die **PowerShell** öffnen. Sobald das Fenster auf ist, geben wir nur den folgenden Befehl ein. Dann erfolgt die Eingabe auf **deutscher Sprache**.

```
Set-WinUserLanguageList -LanguageList de-DE
```

Windows Serverrollen & über die PowerShell installieren

Einleitung

In diesem Beitrag erkläre ich kurz, wie wir über die **PowerShell Windows Serverrollen** installieren können.

Verfügbare Rollen anzeigen lassen

Damit wir erstmal wissen, welche Rollen wir installieren können, müssen wir uns diese Rollen erstmal anzeigen lassen. Dazu verwenden wir den **PowerShell Befehl** `Get-WindowsFeature`.

```
Get-WindowsFeature
```

Wir erhalten dann eine Liste mit allen **Rollen**, die wir auf unserem Server jetzt installieren können. Aus dieser Liste suchen wir eine Rolle heraus. Für unser Beispiel installieren wir die **DNS Rolle**, damit wir einen **DNS-Server** installiert haben.

Rolle auf dem Server installieren

Um jetzt die entsprechende **Rolle** auf dem Server zu installieren, verwenden wir jetzt den **PowerShell Befehl** `Install-WindowsFeature`.

```
Install-WindowsFeature DNS
```

Wir können dazu auch die **Management-Tools** mitinstallieren. Damit erhalten wir dann in der Administrationsoberfläche des **Windows Servers** die Programme, um den **DNS-Server** zu verwalten. Dafür müssen wir lediglich den Parameter `-IncludeManagementTools` zu dem Befehl hinzufügen.

```
Install-WindowsFeature DNS -IncludeManagementTools
```

Nachdem der Fortschrittsbalken durchgelaufen ist, ist die Installation abgeschlossen.

Änderungsdatum mit PowerShell Ändern

Einleitung

Mit der Hilfe von der **PowerShell** können wir das **Änderungsdatum** von Dateien ganz einfach Ändern.

Datum und Uhrzeit Ändern

Um das **Datum** und die **Uhrzeit** jetzt zu Ändern, müssen wir lediglich nur einen Befehl absetzen. In diesem müssen wir nur noch die Parameter für den **Pfad** und den **Zeitstempel (Datum und Uhrzeit)** anpassen. Dann wird durch das Ausführen des Skriptes das Datum und die Uhrzeit verändert.

```
$files = gci -path "PFAD" -Recurse
foreach($f in $files)
{
    $f.LastWriteTime = Get-Date "2016-01-01 00:00:01"
}
```

PowerShell Skript mit dem Windows Taskplaner automatisiert ausführen

Einleitung

Bei sich wiederholenden Prozessen lohnt es sich ein **PowerShell Skript** zu entwickeln, damit solche Prozesse schneller abgearbeitet werden können. Um die Ausführung von Skripten zu automatisieren, kann man den **Taskplaner** von **Windows** nutzen. Mit diesem ist es möglich, Skripte und Programme automatisch zu beliebigen Ereignissen auszuführen.

Wie man ein PowerShell Skript automatisiert ausführen lässt, wird in dieser Anleitung kurz erklärt.

PowerShell Skript ausführen

Im ersten Schritt müssen wir das PowerShell Skript erstellen. Sobald wir dies erledigt haben, müssen wir das Skript auf dem Rechner ablegen oder auf einen Netzwerkpfad, wo der Benutzer des ausführenden Computers Zugriff hat.

Aufgabe erstellen

Jetzt erstellen wir die Aufgabe, indem wir die **Aufgabenplanung** öffnen, und öffnen mit einem Rechtsklick das *Kontextmenü*. In dem *Kontextmenü* wählen wir den Punkt **Einfache Aufgabe erstellen** aus. Dort vergeben wir einen gewünschten Namen und klicken uns weiter durch das Setup.

Jetzt fügen wir eine Aktion hinzu. Da wir ein PowerShell Skript ausführen wollen, öffnen wir die PowerShell Engine und geben das Skript nur als Parameter mit.

Die Maske wird dann wie folgt ausgefüllt:

Aktion: Programm starten

Programm/Skript: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

Argumente hinzufügen: -NoProfile -NoLogo -NonInteractive -ExecutionPolicy Bypass -File DATEI.ps1

Aktion bearbeiten

Geben Sie die Aktion an, die von der Aufgabe ausgeführt werden soll.

Aktion: Programm starten

Einstellungen

Programm/Skript:
C:\Windows\System32\WindowsPowerShell\v1.0\power Durchsuchen...

Argumente hinzufügen (optional): -NoProfile -NoLogo -No

Starten in (optional):

OK Abbrechen

Als letzten Schritt müssen wir, sobald das Setup fertig ist, die **Eigenschaften** der Aufgabe öffnen. Dort müssen wir lediglich nur einen Haken bei "*Mit höchsten Privilegien ausführen*" setzen. Danach drücken wir auf Ok und unsere Aufgabe wurde erstellt.

Wichtig: Unter **Trigger** stellen wir ein, wann das Skript automatisiert ausgeführt wird. Häufig wird hier eine bestimmte Uhrzeit in einem festem Intervall eingestellt.

Active Directory Computer mit Angabe des Betriebssystems mit der PowerShell ausgeben

Einleitung

In diesem Beitrag erkläre ich kurz, wie wir auf unserem Domänencontroller mit der PowerShell alle Computer mit Angabe des installierten Betriebssystems ausgeben können. Dadurch können wir schnell eine Auswertung darüber ziehen, welche Rechner bei uns in der Domäne installiert sind, und welches Betriebssystem einsetzen.

Computer ausgeben

Um die Ausgabe zu erhalten, müssen wir im ersten Schritt die PowerShell-Konsole öffnen. Entweder die *ISE (Integrated Script Environment)* oder direkt die *Kommandozeile*. Dort fügen wir den folgenden Code ein, und ändern den Pfad zu der entsprechenden Datei:

```
Get-ADComputer -Filter * -SearchBase "OU=<Ordnungseinheit>,DC=<DOMAIN>,DC=<LOCAL>" -Properties * |  
Select Name, OperatingSystem | Out-File -FilePath <Pfad>
```

Als Ausgabe erhalten wir eine Liste mit den Computern mit der Angabe des Namens und des verwendeten Betriebssystems.

Tricks

Windows 10 Gottmodus aktivieren

Alle Bereiche der Systemsteuerung

Wer Hard- und Software, angeschlossene Geräte, Updates und Programme konfigurieren will, nutzt für gewöhnlich die Windows-Systemsteuerung. Mit ihren Ober- und Unterkategorien und hierarchischen Ordnerstrukturen erweist sie sich jedoch als reichlich unübersichtlich. Der Windows God Mode vereinfacht den Zugriff auf 250 Einstellungen der Systemsteuerung, listet alles in einer praktischen Liste auf und erstellt für den Sofort-Zugriff eine zentrale Verknüpfung auf dem Desktop.

Wir führen einen Rechtsklick auf eine freie Stelle des Desktops aus und erstellen einen neuen Ordner.

Nun benennen wir den Ordnernamen um.

```
GodMode.{ED7BA470-8E54-465E-825C-99712043E01C}
```

Der Systemsteuerungs-Ordner erscheint nun mit allen Systemsteuerungsfunktionen auf dem Desktop.

Darüber hinaus gibt es noch die bekannten Panels & Verwaltungstools, die hier im Folgenden nur zur Übersicht mit aufgeführt sind:

Verwaltung

Die Computerverwaltung: â€" compmgmt.msc

Geräte-Manager: â€" devmgmt.msc

Datenträgerverwaltung: â€" diskmgmt.msc

Dienste: â€" services.msc

Drucker ändern: â€" control printers

Netzwerkverbindungen: â€" ncpa.cpl

Programme und Features: â€" appwiz.cpl

Windows Systemsteuerung: â€" control

Systemeigenschaften: â€" sysdm.cpl

Konsolen:

certmgr.msc â€” Zertifikate
fsmgmt.msc â€” Freigegebene Ordner
eventvwr.msc â€” Ereignisanzeige
gpdedit.msc â€” Editor fÃ¼r Gruppenrichtlinien
lusrmgr.msc â€” Lokale Benutzer und Gruppen
perfmon.msc â€” LeistungsÃ¼berwachung
secpol.msc â€” Lokale Sicherheitsrichtlinie
services.msc â€” System Services

Einzelne Systemsteuerungselemente:

timedate.cpl â€” Datum und Uhrzeit
desk.cpl â€” Anzeige anpassen
inetcpl.cpl â€” Eigenschaften von Internet
powercfg.cpl â€” Energieoptionen
intl.cpl â€” Region Einstellungen
mmsys.cpl â€” Sound Einstellungen
firewall.cpl â€” Windows Firewall
wscui.cpl â€” Sicherheit und Wartung

Windows 10 / 11 im Auditmodus starten

Einleitung

In diesem Beitrag gehe ich kurz darauf ein, wie wir in **Windows 10 / 11** im **Audit Modus** starten können. Dies ist wichtig, wenn man z.B. Images für den **WDS Server** vorbereiten möchte.

Audit Modus starten

Dazu müssen wir nur in dem **Oobe Screen** (*Out of the box experience*) die Tastenkombination **[STRG] + [Shift] + [F3]** drücken. Dann wird der Administrator aktiviert, und Windows meldet uns als Administrator an.

Dies können wir nutzen, um zum Beispiel ein Image einer Windows-Installation anzufertigen.

How would you like to set up?



Set up for personal use

We'll help you set it up with a personal Microsoft account.
You'll have full control over this device.



Set up for an organization

You'll gain access to your organization's resources like email, network, apps, and services. Your organization will have full control over this device.

Next

Windows Netzwerkadapter einer VLAN ID zuweisen

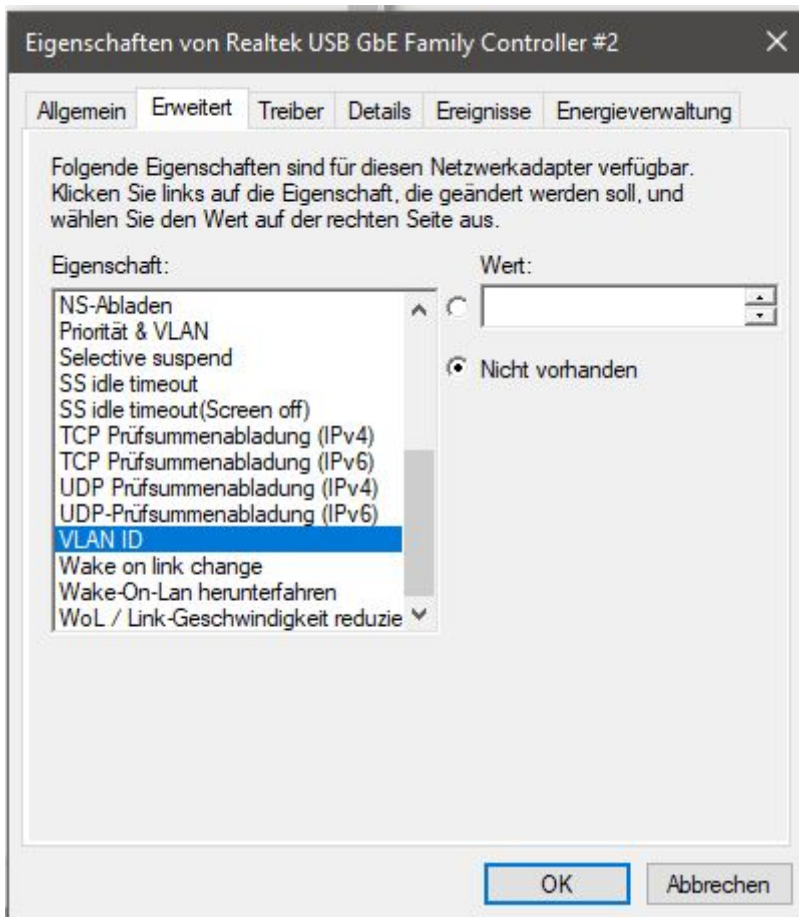
Einleitung

Sobald wir mit größeren Netzwerken arbeiten, und diese z.B. auch mal konfigurieren, müssen wir uns mit den **VLANs** eines Netzwerkes auseinandersetzen. Und damit wir nicht immer die **Ports** von einem **Switch** umkonfigurieren müssen, können wir einfach die **VLAN ID** in unserem **Netzwerkadapter** einstellen. Dadurch sind wir viel flexibler, um schnell von einem Netz in ein anderes Netzwerk zu springen.

Netzwerkadapter anpassen

Um den entsprechenden Netzwerkadapter anzupassen, müssen wir im ersten Schritt den **Geräte-Manager** öffnen. Dazu öffnen wir entweder die **Computerverwaltung** oder geben in der **Windows Suchleiste** **devmgmt.msc** ein.

Sobald sich der **Geräte-Manager** geöffnet hat, suchen wir unter dem Punkt **Netzwerkadapter** den entsprechenden Netzwerkadapter heraus, den wir umstellen möchten. Auf diesen **Netzwerkadapter** machen wir einen **Rechtsklick** und wählen **Eigenschaften**. Dort unter dem Punkt "**Erweitert**" finden wir den Eintrag **VLAN-ID**, dort können wir die entsprechende **VLAN ID** eintragen. Wenn wir dies mit dem **OK** bestätigen, läuft der Datenverkehr jetzt **tagged** mit dem eingegebenen **VLAN**.



Windows 10

Websuche im Startmenü deaktivieren

Einleitung

Windows 10 sucht bei Suchen im Startmenü parallel immer in Bing nach Ergebnissen. Falls es keine Übereinstimmungen im lokalen System gibt, wird automatisch Edge gestartet und die Ergebnisse im Browser angezeigt.

Wir wollen dieses jetzt deaktivieren.

Anwendung

Im ersten Schritt müssen wir den **Registrierungseditor** öffnen. Das machen wir indem wir die *Windows Taste* + *R* drücken und dort **regedit** eingeben.

Dort navigieren wir zu dem nachstehend angegebenen Pfad.

```
HKEY_CURRENT_USER\SOFTWARE\Policies\Microsoft\Windows\Explorer
```

Im nächsten Schritt legen wir einen **DWORD-Wert (32-bit)** an und nennen diesen **DisableSearchBoxSuggestions** an und setzen den Wert auf **1**.

Sobald wir den Rechner neu starten, ist die Änderung aktiv!

Windows 11 Hardware Anforderungen umgehen

Einleitung

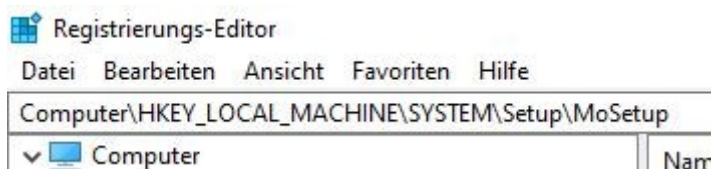
Wenn du auf deinem Rechner Windows 11 installieren möchtest, stößt du bei einem älteren Rechner auf das Problem, dass die Hardware für Windows 11 nicht ausreicht. Es ist möglich, einen Eintrag in der Registry zu hinterlegen, so kannst du Windows 11 dennoch installieren.

Microsoft gibt für Geräte keinen offiziellen Support, die diesen Weg angewandt haben.

Registrierungseintrag

Zuerst musst du den Registrierungseditor aufrufen. Dies kannst du machen, indem du die Tastenkombination **Windows Taste + R** drückst. Dort gibst du dann **regedit** ein und bestätigst dies mit der Eingabetaste.

Navigiere dann zu folgendem Pfad:



Eventuell musst du den Ordner **MoSetup** noch anlegen. Wenn dies der Fall ist, rechtsklicke auf den Ordner **Setup** und wähle dann **Neu**, und im nächsten Schritt dann im **Drop Down Menü** den Eintrag **Schlüssel**. Den Schlüssel nennst du dann **MoSetup**. Bestätige dies mit einem Enter.

Als nächstes erstellst du einen neuen Eintrag in der Registry. Rechtsklicke den Eintrag **MoSetup** und wähle dann unter **Neu** den Punkt **DWORD-Wert (32-bit)**. Den nennst du dann **AllowUpgradesWithUnsupportedTPMOrCPU**. Folgend setzt du den Wert von **0** (false) auf die Zahl **1** (true).

Nun kannst du die ISO von Windows 11 mounten, und im virtuellen Datenträger auf die **setup.exe** klicken und das Upgrade auf Windows 11 durchführen.

Windows 11 herunterladen

Wenn du jetzt die ISO von Windows 11 herunterladen möchtest, kannst du dies über die Website von Microsoft erledigen. Du kannst [hier](#) die ISO-Datei von Windows 11 herunterladen. Entweder lädst du dort direkt die ISO herunter, oder du verwendest das **Media Creation Tool**. Auf welche Weise du dies machst, bleibt dir selbst überlassen.

Bitlocker Status einsehen

Einleitung

In diesem Beitrag erläutere ich kurz, wie wir den **Bitlocker Status** einsehen können. Damit ist es ersichtlich, ob ein **Laufwerk verschlüsselt** ist, oder zu wie viel Prozent ein Laufwerk **verschlüsselt** oder **entschlüsselt** ist.

Status einsehen

Um den Status einzusehen, müssen wir im ersten Schritt eine **administrative Eingabeaufforderung** öffnen, und dann den nachstehenden Code eingeben:

```
manage-bde -status
```

Windows 11

Windows 11 Installation - TPM Anforderung umgehen

Einleitung

Bei der Installation von **Windows 11** in z.B. einer virtuellen Umgebung fordert das Betriebssystem das **TPM-Modul**, welches zur Hardwareverschlüsselung verwendet wird. Dadurch ist es möglich, **Bitlocker** ohne ein **Kennwort** zu verwenden. Wie wir diese Anforderung deaktivieren können, erkläre ich in der nachstehenden Anleitung.

TPM-Anforderung entfernen

Um die Anforderung zu entfernen, booten wir im ersten Schritt ganz normal das Installationsmedium. Sobald wir im "*Installer*" sind, drücken wir die Tasten **Shift + F10** und geben in der **Kommandozeile** folgenden Code ein:

```
REG ADD HKLM\SYSTEM\Setup\LabConfig /v BypassTPMCheck /t REG_DWORD /d 1
```

Jetzt können wir der Installation ganz normal folgen und der Installer sollte jetzt keine Fehler mehr auswerfen.

Windows Deployment Service

Treiber in WDS Startabbild integrieren

Einleitung

In diesem Beitrag gehe ich drauf ein, wie wir in unserem **WDS-Server**, neue **Treiber** integrieren können, die dann während der Installation vom **Windows Betriebssystem** geladen und installiert werden. Dadurch können wir den **Installationsprozess** beschleunigen, da wir später nicht erst wieder Zeit damit verbringen müssen, die **Treiber** wieder zu installieren.

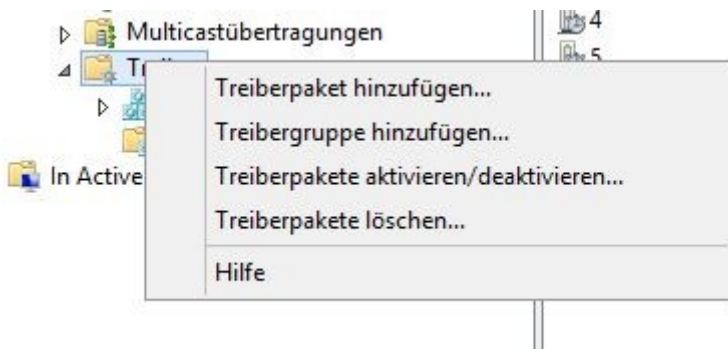
Treiber herunterladen

Im ersten Schritt müssen wir die Treiber herunterladen. Die Treiber dürfen aber **nicht** als **EXE** oder im **MSI Format** vorliegen. Diese Dateien legen wir dann auf unserem **WDS-Server** in einem Verzeichnis unserer Wahl ab. Ich verwende dazu folgenden Pfad:

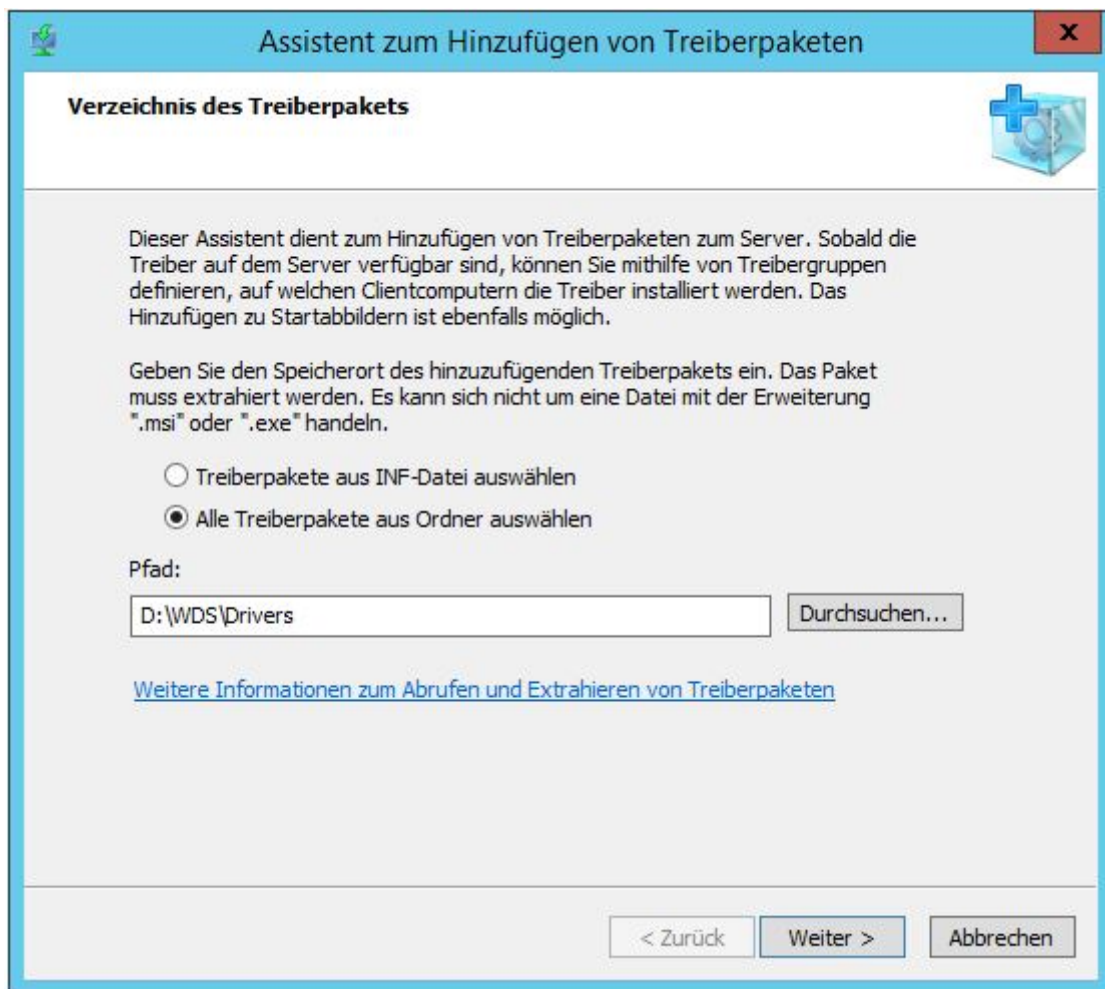
```
D:\WDS\Drivers
```

Treiberpaket hinzufügen

Im nächsten Schritt müssen wir ein Treiberpaket hinzufügen. Dazu öffnen wir die **Konsole** des **Windows Bereitstellungsdienstes**. Dort machen wir einen Rechtsklick auf **Treiber** und wählen "**Treiberpaket hinzufügen**" aus.



Im Anschluss öffnet sich ein Fenster. Dort geben wir jetzt den Pfad zu unserem **Treiber Verzeichnis** an, oder wir wählen die **INF-Datei** unseres Treibers aus, wenn wir nur gezielt einen Treiber hinzufügen möchten.




Sobald wir auf "Weiter" drücken, werden die Treiber eingelesen. Hier können wir auswählen, welche Treiber wir hinzufügen möchten. Die Auswahl bestätigen wir wieder mit einem "Weiter".


Im Anschluss bestätigen wir nach Überprüfung der Zusammenfassung auf Richtigkeit das Fenster wieder mit einem "Weiter". Die Treiber werden jetzt dem **WDS Server** hinzugefügt.


Nach dem Hinzufügen der Treiber werden wir abgefragt, ob wir die Treiber einer **Treibergruppe** hinzufügen möchten, dafür haben wir 3 Optionen zur Auswahl.

Vorhandene Treibergruppe auswählen:	Hinzufügen der Treiber einer vorhandenen Treibergruppe.
Neue Treibergruppe mit folgendem Namen erstellen:	Es wird eine Treibergruppe angelegt mit dem angegebenen Namen, und die Treiber werden dann dieser Treibergruppe hinzugefügt.
Derzeit keine Treiberpakete in eine Treibergruppe einfügen:	Hier werden die Treiber keiner Gruppe zugewiesen, und sind damit allen Geräten zugänglich. Die Treiber können im späteren Betrieb aber immer noch einer Gruppe zugewiesen werden.



Assistent zum Hinzufügen von Treiberpaketen



Treibergruppen

Bei einer Treibergruppe handelt es sich um eine Sammlung von Treiberpaketen, die für eine bestimmte Gruppe von Clients verfügbar ist.

Zum Bereitstellen dieser Pakete für Clients müssen diese einer Treibergruppe hinzugefügt werden. Ein Paket kann sich in zahlreichen Treibergruppen befinden, muss jedoch mindestens in einer Gruppe enthalten sein, um für Clients verfügbar zu sein.

Geben Sie die Treibergruppe für die hinzugefügten Pakete ein:

☐ Vorhandene Treibergruppe auswählen:

Alle-Geräte

☐ Neue Treibergruppe mit folgendem Namen erstellen:

☒ Derzeit keine Treiberpakete in eine Treibergruppe einfügen

< Zurück

Weiter >

Abbrechen

Damit haben wir jetzt Treiber unserem **WDS-Server** hinzugefügt.

Migration auf einen neuen WDS-Server

Einleitung

Bei der Installation eines **WDS (Windows Deployment Service Server)** ist es unter Umständen von Vorteil, wenn man die vorhandenen Dateien auf den neuen Server migrieren kann. So muss keine Arbeit mehr investiert werden, die vorhandenen Abbilder neu zu bauen und diese dann im WDS zu integrieren.

Migration eines WDS

Um den WDS erfolgreich zu migrieren, muss auf dem neuen Server im ersten Schritt die Rolle **Windows Bereitstellungsdienste / Windows Deployment Service** installiert werden. Sobald wir diese installiert haben, und die initialen Einstellungen getätigt haben, können wir beginnen mit der Migration.

Jetzt müssen wir auf dem alten **WDS-Server** einmal den **Dienst "Windows Bereitstellungsdienste"** stoppen! Sobald wir dies erledigt haben, müssen wir den **WDS-Ordner** im gesamten einmal auf den neuen Server in das **WDS-Verzeichnis** kopieren.

Im Anschluss starten wir auf dem neuen Server den **WDS-Dienst**. Dann sollte die Dateien erfolgreich auf den neuen Server migriert sein.

Windows Server

NTP Zeitserver Synchronisation auf einem Windows Server konfigurieren

Einleitung

Sobald wir einen **Windows-Server** konfigurieren, möchten wir diesen vielleicht auch mit einem **Zeitserver** synchronisieren. Dazu müssen wir die Konfiguration des **Windows-Servers** ändern. Dafür geben wir nur ein paar Befehle in die **Kommandozeile** ein.

Zeitserver verändern

Im ersten Schritt müssen wir die **Kommandozeile** mit **administrativen Berechtigungen** starten. Im Anschluss geben wir den folgenden Befehl ein, um den **NTP Server Dienst** zu stoppen.

```
net stop w32time
```

Im zweiten Schritt müssen wir jetzt die **NTP Zeitserver** angeben. Wir verwenden dabei einen öffentlichen, Deutschen Zeitserver. Dafür geben wir wieder den nachstehenden Befehl ein.

Wir können auch mehrere **Zeitserver** angeben, diese müssen wir dann mit einem **(Komma)** voneinander trennen.

```
w32tm /config /syncfromflags:manual /manualpeerlist:"de.pool.ntp.org"
```

Jetzt geben wir den folgenden Befehl ein, um die Verbindung zu verifizieren.

```
w32tm /config /reliable:yes
```

Zum Schluss müssen wir den **NTP Dienst** jetzt nur noch starten. Dann können sich die Clients wieder mit unserem **NTP Server** synchronisieren.

```
net start w32time
```

Active Directory Kennwort Änderung über Web-GUI

Einleitung

In diesem Beitrag erkläre ich kurz, wie wir es unseren Benutzern ermöglichen können, ihr Kennwort über eine Weboberfläche zu ändern. Dies hat den entscheidenden Vorteil, dass die Benutzer dafür nicht zwangsweise am Rechner sein müssen. Sie können sich an jedem Endgerät befinden, der diesen Webserver erreichen kann.

Kennwort Änderung ermöglichen

Remotedesktop Web Access installieren

Im ersten Schritt müssen wir über den **Server-Manager** über die **Server-Rollen** den Serverdienst **Remotedesktopdienste** installieren. Im nächsten Schritt können wir die **Features** so lassen, wie diese eingestellt sind. Wir benötigen da keine weiteren Features.

Jetzt bei den **Rollendiensten** müssen wir den **Web Access für Remotedesktop** auswählen und dann das "Setup" weiter fortführen. Bei den folgenden Fenstern klicken wir diese durch, wie wir es aus anderen **Server-Rollen Installationen** kennen.

DNS Eintrag für Webserver hinzufügen

Jetzt müssen wir auf unserem **DNS-Server** einen CNAME Eintrag erstellen. Dieser dient dazu, dass wir unsere Webseite nachher über eine "Domain" erreichen können. Falls wir unseren Server von Außen erreichbar machen möchten, müssen wir bei den öffentlichen DNS Anbietern, die Subdomain hinzufügen. Da wir aus Sicherheitsgründen die Seite nur aus dem **internen Netzwerk** erreichbar machen wollen, tragen wir einen **CNAME Eintrag** bei unserem lokalen **DNS-Server** ein.

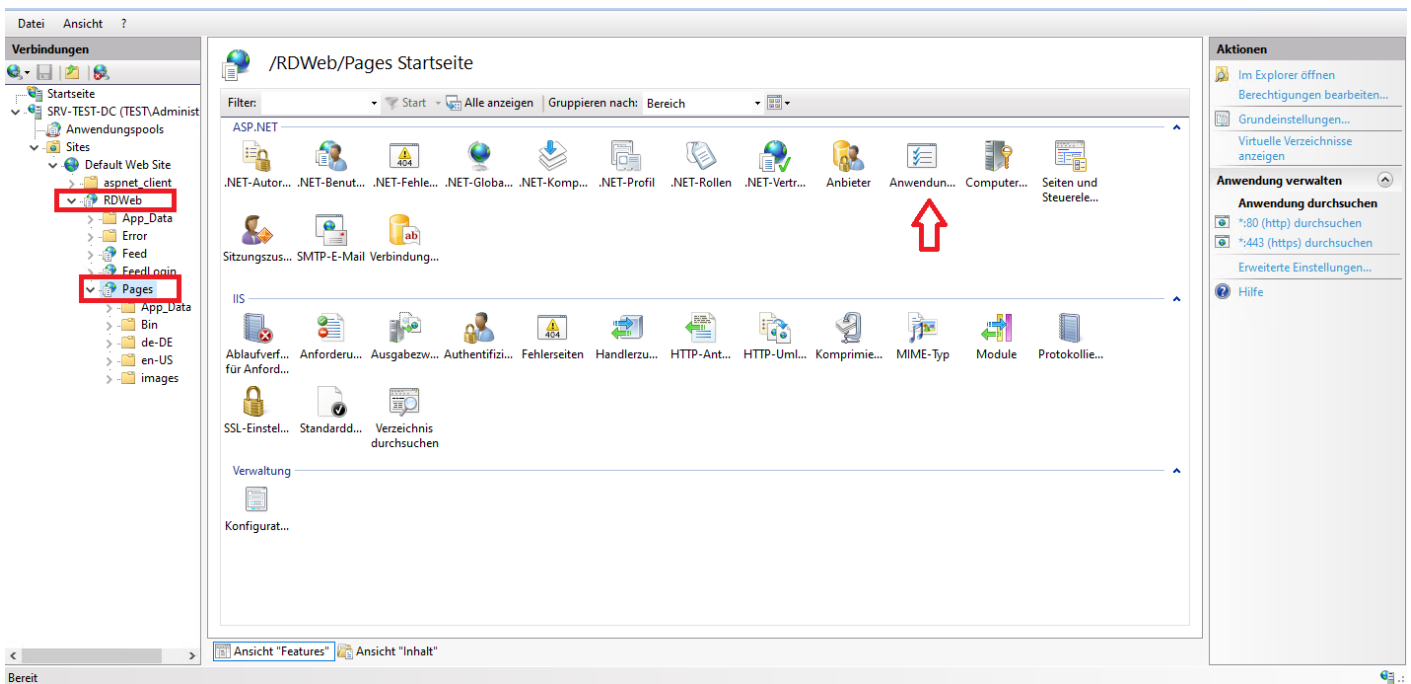
Beachte: Jeden Dienst den wir nach außen auf machen (Sprich, es kann sich jeder der unsere IP / Domain + ggf. Port kennt, die Seite öffnen), ist immer ein Sicherheitsrisiko. Hacker können so Sicherheitslücken ausnutzen, um so erheblichen Schaden anzurichten!

The screenshot shows a Windows dialog box titled "Eigenschaften von password". It has two tabs: "Alias (CNAME)" and "Sicherheit". The "Alias (CNAME)" tab is active. It contains three text input fields and a button. The first field is labeled "Aliasname (bei Nichtangabe wird übergeordnete Domäne verwendet):" and contains the text "password". The second field is labeled "Vollqualifizierter Domänenname:" and contains the text "password.ad.test.local". The third field is labeled "Vollqualifizierter Domänenname des Zielhosts:" and contains the text "srv-test-dc.ad.test.local.". To the right of this third field is a button labeled "Durchsuchen...". At the bottom of the dialog box are three buttons: "OK", "Abbrechen", and "Übernehmen".

Wenn wir jetzt, wie in meinem Beispiel die Domain `password.ad.test.local` anpingen, dann sollten wir eine Antwort erhalten. Damit stellen wir sicher, dass die Verbindung zum Server hergestellt werden kann.

Internetdienst konfigurieren

Jetzt müssen wir den "**IIS Manager**" so konfigurieren, dass die Benutzer auf unserer Webseite Ihr Kennwort ändern können. Dazu öffnen wir jetzt den **IIS Manager** und navigieren zum letzten Unterpunkt **Pages** im Ordner **RDWeb** und öffnen dort die **Anwendungseinstellungen**.



In den **Anwendungseinstellungen** suchen wir dann den Wert **PasswordChangeEnabled** und ändern den Wert dann auf `true`. Jetzt können wir mit einer der nachstehenden URLs testen, ob wir die Seite öffnen können.

Auf dem Server: `https://localhost/RDWeb/Pages/de-DE/password.aspx`

Intranet: `https://password.ad.test.local/RDWeb/Pages/de-DE/password.aspx`

Es könnte sein, dass der Browser eine Fehlermeldung ausgibt, dass das **SSL Zertifikat** fehlerhaft ist. Wenn dies der Fall ist, können wir im **IIS Manager** unter **SSL-Einstellungen** beim Punkt **SSL Erforderlich** den Haken entfernen. Wichtig ist, dass wir uns dann später dazu kümmern müssen, eine **HTTPS Verbindung** herstellen zu können. Daher sollte dieser Weg nur zu Test Zwecken verwendet werden.

Optional: HTTP Umleitung einstellen

Wenn wir ein funktionierendes Zertifikat haben, sollten wir dafür sorgen, dass die Verbindungen über HTTPS erfolgen. Dafür müssen wir lediglich wieder im **IIS Manager** unter dem Punkt **HTTP-Umleitung** den Haken bei *Anforderungen zu diesem Ziel umleiten*, und *alle Anforderungen an eigentliches Ziel (und nicht relativ zum Ziel) umleiten* setzen.

Im Anschluss tragen wir im ersten Textfeld den HTTPS Link ein. Und den Statuscode ändern wir auf **Dauerhaft (301)**.

Optional: Anpassungen an der Weboberfläche

Jetzt können wir nach Belieben die Oberfläche anpassen. Die sieht so ja noch recht langweilig und alt aus! :) *Windows Vista lässt grüßen!*

Dazu Öffnen wir die folgende Datei unter `C:\Windows\Web\RDWeb\Pages\de-DE\RDWAStrings.xml`

Dort können wir folgende Einträge ändern:

- **PageTitle:** "Passwort ändern"
- **HeadingRDWA:** "Passwort ändern"
- **HeadingApplicationName:** "Portal zur Passwortänderung"

Wenn man möchte, kann man noch in der Datei `C:\Windows\Web\RDWeb\Pages\de-DE\password.aspx` den String von `L_CompanyName_Text` auch auf "Passwort ändern" setzen.

Wer den Hintergrund ändern möchte, kann die Datei `C:\Windows\Web\RDWeb\Pages\images\bg_globe_01.jpg` austauschen.

Das Logo können wir ersetzen, indem wir die Datei `C:\Windows\Web\RDWeb\Pages\images\logo_02.png` ersetzen.

Wer will, kann sich bei der Seite noch mehr austoben! :)

Ordnerumleitung mit einer Gruppenrichtlinie konfigurieren

Einleitung

In diesem Beitrag erkläre ich kurz, wie wir mithilfe einer **Gruppenrichtlinie**, eine **Ordnerumleitung** einstellen können. Diese Art der Dateiablage für die Benutzerprofile hat einen enormen Geschwindigkeits-Vorteil und behebt einige Probleme im Vergleich zu den **Roaming Benutzerprofilen**.

Durchführung

Welche Ordner können umgeleitet werden?

Mithilfe der **Ordnerumleitung** können wir dann folgende Ordner auf einen **Netzwerk-Pfad** umleiten:

- ☒ AppData (Roaming)
- ☒ Desktop
- ☒ Startmenü
- ☒ Dokumente
- ☒ Bilder
- ☒ Musik
- ☒ Videos
- ☒ Favoriten
- ☒ Kontakte
- ☒ Download
- ☒ Verknüpfungen
- ☒ Suchvorgänge
- ☒ Gespeicherte Spiele

Voraussetzungen

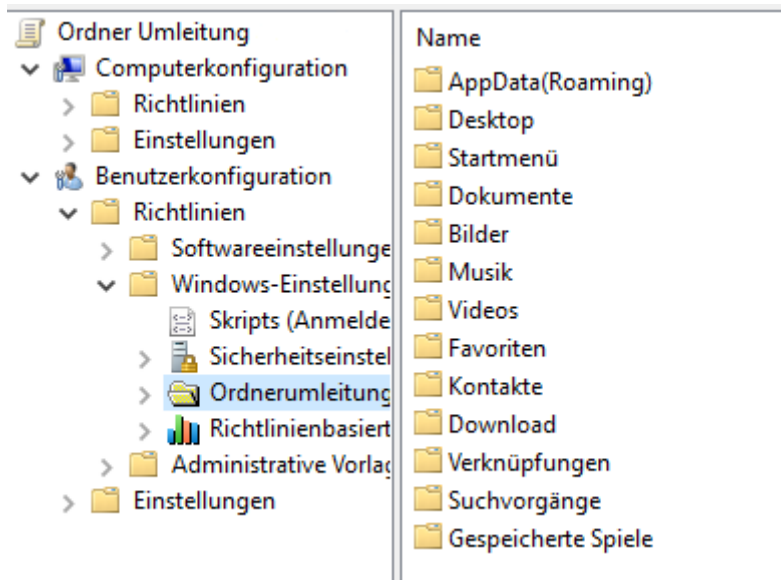
- Active Directory (Windows-Server 2016, Windows-Server 2019, Windows-Server 2022)
- Freigegebenen Netzwerk-Pfad mit entsprechenden Berechtigungen für die Benutzer
- Netzwerk muss bei der ersten Anmeldung vorhanden sein

Ordnerumleitung einrichten

Um die Ordnerumleitung mittels der **Gruppenrichtlinien** einzurichten, müssen wir im ersten Schritt die **Gruppenrichtlinienverwaltung** öffnen.

Dort legen wir eine neue **Gruppenrichtlinie** an, und bearbeiten diese. Im Editor navigieren wir zum folgenden Pfad:

Benutzerkonfiguration / Richtlinien / Windows-Einstellungen / Ordnerumleitung



Dort angekommen, können wir allen dort angegebenen Ordnern eine Umleitung verpassen. Dazu markieren wir den ersten Ordner und öffnen durch einen *Rechtsklick* und dem folgenden *Klick* auf "Eigenschaften" die Einstellungen für diesen Ordner.

In der Eingabemaske entscheiden wir uns für einen von beiden "Einstellungen".

Standard: Für jeden Benutzer werden die Ordner auf den gleichen Pfad umgeleitet

Erweitert: Spezielle Gruppen werden auf spezielle Freigaben umgeleitet.

Wir verwenden **Standard**, da dies wohl die meist verwendete Art sein wird. Im nächsten Schritt unter **Zielordner** geben wir an, wie mit den Dateien umgegangen werden soll. In unserem Fall verwenden wir "Einen Ordner für jeden Benutzer im Stammpfad erstellen". Dies ermöglicht, dass der Benutzer einen **eigenen Ordner** hat, indem seine eigenen Dateien liegen. Der Server legt in diesem Fall bei einer neuen Anmeldung für den Benutzer eine **eigene Verzeichnisstruktur** in dem **Stammpfad** an.

Wahlweise können wir auch "*an folgenden Pfad umleiten*" verwenden, falls alle Benutzer dieselben Dateien und Dokumente haben sollen. Es ist vielleicht von Vorteil bei Favoriten oder Desktop.

Als letzten Schritt geben wir nur noch das **Stammverzeichnis** an. In diesem angegebenen Verzeichnis werden dann einzelne Ordner für die entsprechenden Benutzer angelegt.

Diese Schritte wiederholen wir mit allen benötigten Ordnern, die wir umleiten möchten.

Lizenzen

Aktivierungsserver von Microsoft

Einleitung

Bei der Aktivierung von **Microsoft Produkten** kann es hin und wieder zu Problemen kommen, wenn die **Firewall** gewisse **URLs** oder **Ports** blockt. Da helfen ggf. die nachstehenden Informationen, um das Problem zu beseitigen.

URLs

Microsoft verwendet die folgenden URLs, um seine Produkte zu aktivieren:

```
https://go.microsoft.com/
http://go.microsoft.com/
https://login.live.com
https://activation.sls.microsoft.com/
http://crl.microsoft.com/pki/crl/products/MicProSecSerCA_2007-12-04.crl
https://validation.sls.microsoft.com/
https://activation-v2.sls.microsoft.com/
https://validation-v2.sls.microsoft.com/
https://displaycatalog.mp.microsoft.com/
https://licensing.mp.microsoft.com/
https://purchase.mp.microsoft.com/
https://displaycatalog.md.mp.microsoft.com/
https://licensing.md.mp.microsoft.com/
https://purchase.md.mp.microsoft.com/
```

Ports

Microsoft verwendet dazu auch nicht den Standard **HTTP (80)** oder **HTTPS (443)** Port. Microsoft verwendet den Port: **TCP / 1688**

Programme

Microsoft PowerToys installieren

Einleitung

In diesem Beitrag erkläre ich kurz, wie wir die **Microsoft PowerToys** auf einem **Windows 10 / Windows 11** System installieren können. Die **PowerToys** enthalten viele nützliche Tools, die unseren Arbeitsalltag erleichtern können. Die **PowerToys** werden von **Microsoft** entwickelt und zur Installation verteilt.

Installation der PowerToys

Um die **PowerToys** zu installieren, müssen wir einmal die **PowerShell** mit **administrativen Berechtigungen** öffnen. Dort geben wir dann den folgenden Befehl ein:

```
winget install Microsoft.PowerToys --source winget
```

Wenn die Installation jetzt abgeschlossen ist, können wir die **Microsoft PowerToys** jetzt nutzen!