

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, fügen den 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 darauf.

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
        }
    } else {
        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-WindowsImage -ImagePath $Pfad).ImageIndex
            ForEach ($Nummer in $Index) {
                Dism.exe /Get-ImageInfo /Imagefile:$Pfad /index:$Nummer
            }
        }
    }
}

```

```

    }
}
else {
    $Wims = Get-ChildItem -Path $Pfad -include *.wim,*.esd,*.swm -Recurse
    ForEach ($Pfad in $Wims) {
        $Index = (Get-WindowsImage -ImagePath $Pfad).ImageIndex
        ForEach ($Nummer in $Index) {
            Dism.exe /Get-ImageInfo /Imagefile:$Pfad /index:$Nummer
        }
    }
}
}
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 **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>

Revision #6

Created 21 March 2022 07:07:44 by Phillip U.

Updated 16 May 2024 14:26:13 by Phillip U.