

Handreichung  
zur Installation von Windows  
in Schulen

Klonen von Windows  
mit WinPE,  
DISM und IMAGEX

DISM oder ImageX sind Standardprogramme von Microsoft, mit dem Windows installiert wird. Diese Programme lassen sich auch gut verwenden, um selbst Windows-Images zu erzeugen und um Windows zu klonen. Vor allem in Verbindung mit einem WinPE-Live-System erhält man damit ein flexibles Werkzeug, als Rettungssystem, zur Systemsicherung, zur Systemwiederherstellung oder zum Klonen von Windows-Computern.

## INHALT

Eigenschaften von DISM oder ImageX .....	3
Systemsicherung .....	3
Systemwiederherstellung.....	4
Vorbereitung der Festplatte .....	4
Installation des Wim-Images mit DISM oder ImageX.....	5
Einrichten des Boot-Managers .....	6
Automatisieren der Installation.....	6
Installation vom Netzwerk.....	7
Auswahl des Wim-Images .....	7
Sicherheitsabfrage.....	9
Auswahl: Bios oder UEFI.....	9
Vorbereitung der Festplatte mit diskpart.....	10
Überprüfung auf Fehler .....	11
Pause, timeout oder Countdown .....	11
Installationsskript startjob.bat .....	12

## IMPRESSUM

Akademie für Lehrerfortbildung und Personalführung, Dillingen (<http://alp.dillingen.de>)

Die Handreichung wurde im Rahmen des Projektes SCHULNETZ von Systembetreuern und IT-Multiplikatoren erarbeitet. Sie ist unter der Adresse <http://alp.dillingen.de/schulnetz/materialien> abrufbar.

Dokumentation:       Georg Schlagbauer, Akademie Dillingen  
                           Barbara Maier, Akademie Dillingen  
                           Christian Maushart, Bürgernetz Dillingen  
 E-Mail:                 schlagbauer@alp.dillingen.de  
 Stand:                 Juni 2018



**DISM** und **ImageX** sind Befehlszeilenprogramme, mit dem Windows-Images (WIM-Dateien) erstellt oder zurückgespielt werden können. Um Computer mit DISM oder ImageX zu klonen, kann der Computer z. B. von einem WinPE-Live-System gestartet werden.

ImageX war bis Windows 7 das klassische Werkzeug von Microsoft, um WIM-Images zu erstellen und zu verteilen. DISM war das Werkzeug, um WIM-Images zu bearbeiten, z. B. um Treiber zu integrieren.

Seit Windows 8 ist der Funktionsumfang von DISM erweitert worden und deckt nun auch die Funktionen ab, die ImageX anbietet. ImageX ist also eigentlich überflüssig, ist aber immer noch in den Werkzeugen von Microsoft enthalten. Da sich viele Systembetreuer mit ImageX auskennen und auch funktionierende Skripte dafür haben, wird in dieser Dokumentation auf beide Möglichkeiten eingegangen.

## EIGENSCHAFTEN VON DISM ODER IMAGEX

Anders als die meisten gängigen Imaging-Programme (z. B. DriveSnapshot, Acronis TrueImage, PartImage, etc.) arbeitet DISM oder ImageX auf Dateiebene und verhält sich dadurch etwas anders als die sektorbasierten Imaging-Programme. Dies hat folgende Auswirkungen:

- Da mit DISM oder ImageX nur Dateien kopiert werden, spielt die Größe der Windows-Partition beim Erstellen oder Zurückspielen keine Rolle. Auch die Partitionsstrukturen von Quelle und Ziel können völlig unterschiedlich sein. Es spielt beispielsweise auch keine Rolle, wenn ein Windows-Image im BIOS-Modus erstellt wurde und auf ein UEFI-System übertragen wird.
- Die Partition und das Dateisystem (NTFS) müssen vor dem Zurückspielen einer WIM-Datei vorhanden sein. Wenn ein Image auf einen neuen Computer aufgespielt werden soll, müssen vorher (z. B. mit diskpart) die Partitionen angelegt und formatiert werden.
- Der Bootmanager wird grundsätzlich nicht mitgesichert. Nach dem Klonen wird der Bootmanager neu eingerichtet, damit Windows starten kann.

## SYSTEMSICHERUNG

### SYSTEMSICHERUNG MIT DISM

Der Computer wird vom WinPE-Live-System (CD oder USB-Stick) gestartet.

```
dism /capture-image /capturedir:<Quelle> /imagefile:<Ziel> /name:<Beschreibung>
```

```
dism /capture-image /capturedir:D:\ /imagefile:Win10.wim /name:Win10
```

### SYSTEMSICHERUNG MIT IMAGEX

Der Computer wird vom WinPE-Live-System (CD oder USB-Stick) gestartet. Einfachheitshalber sollte das Programm **imagex.exe** im Live-System enthalten sein.

```
imagex /capture <Quelle> <Ziel> <Beschreibung>
```

```
imagex /capture d: z:\win10.wim "Win10"
```



## SYSTEMWIEDERHERSTELLUNG

Der Computer wird vom WinPE-Live-System (CD oder USB-Stick) gestartet. Die Systemwiederherstellung geschieht in 3 Schritten:

1. Vorbereitung der Festplatte
2. Installation des WIM-Images
3. Einrichten des Bootmanagers

## VORBEREITUNG DER FESTPLATTE

Die nachfolgenden Beispiele gehen davon aus, dass die Festplatte komplett gelöscht und neu eingerichtet werden soll. Wenn auf der Festplatte Daten erhalten bleiben sollen, muss das Verfahren natürlich variiert werden.

Bei der Vorbereitung der Festplatte unterscheidet man zwischen dem BIOS und dem UEFI-Modus. Die Unterschiede zwischen BIOS und UEFI sind in der Handreichung: „BIOS und UEFI“ (<http://alp.dillingen.de/schulnetz/materialien/UEFI.pdf>) erläutert.

## VORBEREITUNG EINER MBR-FESTPLATTE (BIOS-MODUS)

Die Festplatte (disk 0) wird komplett gelöscht und neu eingerichtet:

- Partition 1 500 MB Systempartition
- Partition 2 Rest Windows-Startpartition

```
diskpart
DISKPART> list disk
DISKPART> select disk 0
DISKPART> clean
DISKPART> create partition primary size=500
DISKPART> format fs=ntfs quick label="Hide"
DISKPART> active
DISKPART> assign letter=s
DISKPART> create partition primary
DISKPART> format fs=ntfs quick label="Windows 10"
DISKPART> assign letter=w
DISKPART> exit
```

Die beiden „assign“-Befehle und die zugewiesenen Laufwerksbuchstaben haben nur eine temporäre Bedeutung, damit man sofort nach dem Einrichten (ohne Neustart) auf die Laufwerke zugreifen kann.



## VORBEREITUNG EINER GPT-FESTPLATTE (UEFI-MODUS)

Die Festplatte (disk 0) wird komplett gelöscht und neu eingerichtet:

- Partition 1 100 MB Systempartition
- Partition 2 16 MB MSR-Partition (Microsoft Reserved)
- Partition 3 Rest Windows-Partition

Auf eine Windows-Recovery-Partition kann beim Klonen verzichtet werden.

```
diskpart
DISKPART> list disk
DISKPART> select disk 0
DISKPART> clean
DISKPART> convert gpt
DISKPART> create partition efi size=100
DISKPART> format fs=fat32 quick label="System"
DISKPART> assign letter=s
DISKPART> create partition msr size=16
DISKPART> create partition primary
DISKPART> format fs=ntfs quick label="Windows"
DISKPART> assign letter=w
DISKPART> exit
```

## INSTALLATION DES WIM-IMAGES MIT DISM ODER IMAGEX

Das nachfolgende Beispiel geht davon aus, dass sich das WIM-Image im Laufwerk z: (z:\win10.wim) befindet und die Windows-Partition mit dem Laufwerksbuchstaben w: angesprochen wird.

### INSTALLATION MIT DISM

```
dism /apply-image /imagefile:<Quelle> /applydir:<Ziel> /Index:1
dism /apply-image /imagefile:z:\win10.wim /applydir:w:\ /Index:1
```

### INSTALLATION MIT IMAGEX

```
imagex /apply <Quelle> <Image-Nr.> <Ziel>
imagex /apply z:\win10.wim 1 w:
```

Die Installation des Images kann einige Minuten dauern.



## EINRICHTEN DES BOOT-MANAGERS

Bis Windows XP wurde der Bootloader NTLDR verwendet, der die Konfigurationsdatei boot.ini interpretierte, die vielen Systembetreuern bekannt war.

Seit Vista verwendet Windows den Bootmanager bootmgr. Dieser liegt bei MBR-Systemen in der Systempartition und greift auf das Verzeichnis \Boot zu, in dem sich die eigentliche Bootkonfiguration befindet. Bei GPT-Systemen liegt der Windows Bootmanager (bootmgr.efi) in der ESP-Partition in Verzeichnis \EFI\Microsoft\Boot. Systempartition bzw. ESP-Partition sind nicht mit einem Laufwerksbuchstaben belegt, so dass der Bootmanager für einen normalen Benutzer nicht sichtbar ist.

Zum Glück muss man dies eigentlich gar nicht wissen; Wenn alles glatt läuft, legt das Programm bcdboot den Bootmanager automatisch richtig an.

### BCDBOOT

Die Laufwerksbuchstaben im WinPE-Live-System und im laufenden Windows-System müssen nicht identisch sein. Das nachfolgende Beispiel geht davon aus, dass die Partition in der das Wim-Image installiert wurde, mit dem Laufwerksbuchstaben w: angesprochen wird und die Systempartition bzw. ESP-Partition ggf. mit dem Laufwerksbuchstaben s: angesprochen werden kann.

```
bcdboot w:\windows
```

Das Einrichten des Bootmanagers läuft üblicherweise problemlos ab, wenn WinPE im gleichen Modus (BIOS oder UEFI) gestartet wurde, wie das später zu startende System. Wenn dies nicht der Fall ist, kann man durch Zusatzoptionen den richtigen Bootmanager einrichten.

```
bcdboot w:\windows /s s:           Angabe der Systempartition bzw. ESP-Partition
```

```
bcdboot w:\windows /s s: /f UEFI   Angabe des Firmware-Typs
```

```
bcdboot w:\windows /s s: /f BIOS
```

```
bcdboot w:\windows /s s: /f ALL
```

## AUTOMATISIEREN DER INSTALLATION

Die Installationsschritte zur Systemwiederherstellung mit DISM oder ImageX lassen sich automatisieren. Dazu werden die einzelnen Befehle in ein Skript (startjob.bat) geschrieben. Das Skript wird entweder beim Booten von WinPE automatisch aufgerufen oder es wird manuell gestartet.

1. Vorbereitung der Festplatte
2. Installation des WIM-Images
3. Einrichten des Bootmanagers

Eine Automatisierung ist immer ein Blindflug. Vor allem das Löschen der Festplatte (select disk 0; clean) kann sich ungünstig auswirken, wenn disk 0 nicht die richtige Festplatte sein sollte.



## INSTALLATION VOM NETZWERK

Wenn das zu installierende Wim-Image auf einem Netzlaufwerk liegt (z. B. auf einer NAS-Box) wird nur eine Netzwerkverbindung hergestellt und das eigentliche Installationskript im Netzlaufwerk aufgerufen.

Voraussetzung ist natürlich, dass in WinPE die Netzwerk-Treiber enthalten sind. Auch soll davon ausgegangen werden, dass das WinPE-System per DHCP eine sinnvolle IP-Adresse erhalten hat.

---

### STARTJOB.BAT ZUR NETZVERBINDUNG

```
@echo off
net use n: \\10.36.104.24\Vorlagen /user:Lehrer 12345

if not exist n:\Wim-Images\startjob.bat (
    echo Verbindung ins Netz war nicht erfolgreich.
    goto Ende
)

n:
cd \Wim-Images

call startjob.bat

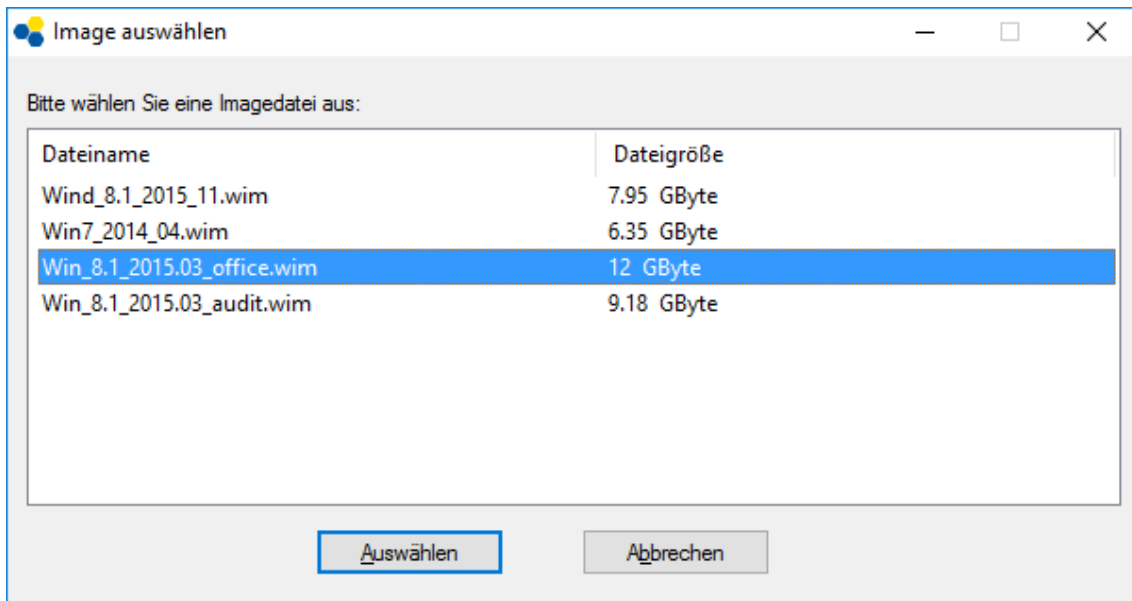
:Ende
```

## AUSWAHL DES WIM-IMAGES

Der Name des Wim-Images muss nicht starr vorgegeben werden. Falls mehrere Wim-Images zur Auswahl stehen, will man das richtige Image gegebenenfalls auswählen können.

Die Auswahl eines Wim-Images gestaltet sich mit einer grafischen Oberfläche etwas ansprechender. Das folgende Tool „Wim\_Auswaehlen.exe“ ist mit dem Programm AutoIT erstellt. Es bietet die Wim-Dateien im aktuellen Verzeichnis zur Auswahl an. Die Auswahl wird im Temp-Verzeichnis unter imagename.txt gespeichert.





---

**BEISPIELSKRIPT ZUR DEMONSTRATION VON WIM\_AUSWAEHLLEN.EXE**

```
@echo off
Wim_Auswaehlen.exe

if not exist %TEMP%\imagenname.txt goto Ende
set /p wimimage= < %TEMP%\imagenname.txt
if "%wimimage%"==" " (
    echo Es wurde kein Image ausgewaehlt.
    goto Ende
)

...(weitere Befehle)

:Ende
```

Wenn die Auswahl nicht abgebrochen wurde, steht in der Variablen %wimimage% der Name des Wim-Images zur Verfügung.

Das Programm Win\_Auswaehlen.exe enthält auch noch eine Sicherheitsabfrage, die darauf hinweist, dass anschließend die Festplatte gelöscht wird. Alternativ kann die Sicherheitsabfrage auch auf Kommandozeile erfolgen.



## SICHERHEITSABFRAGE

In vielen Skripten ist es sinnvoll – auch wenn eigentlich alles automatisch laufen sollte – eine Sicherheitsabfrage zu stellen, um den Benutzer die Möglichkeit zu geben, den Vorgang abubrechen.

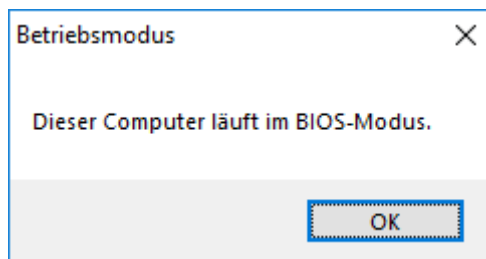
### BEISPIELSKRIPT FÜR EINE SICHERHEITSABFRAGE

```
@echo off
echo Neuinstallation von Windows 10
echo Es wird die gesamte Festplatte dieses Computers gelöscht.
set /p choice=Wollen Sie weitermachen (j/n)?
if not %choice%==j goto Ende
...
...
:Ende
```

## AUSWAHL: BIOS ODER UEFI

Solange alle Computer den BIOS-Modus unterstützen, gibt es keinen Grund auf UEFI umzusteigen. Wenn man die Installation dennoch flexibel halten möchte, kann man überprüfen, ob der Computer unter WinPE im BIOS- oder UEFI-Modus gestartet wurde. Je nachdem wird auch die Festplatte im BIOS- oder UEFI-Modus eingerichtet.

Das kleine AutoIT-Programm BiosUefi.exe bzw. die Kommandozeilenversion BisUefi\_Console.exe prüft, ob der Computer im Bios oder Uefi-Modus gestartet wurde.



### BEIPIELSKRIPT ZUM EINSATZ VON BIOSUEFI\_CONSOLE.EXE

```
@echo off
BiosUefi_Console.exe > %TEMP%\tmpFile
set /p BIOSUEFI= < %TEMP%\tmpFile
del %TEMP%\tmpFile
echo %BIOSUEFI%
```

## VORBEREITUNG DER FESTPLATTE MIT DISKPART

Um die Befehle von diskpart zu automatisieren werden die Befehle in eine Textdatei (z. B. hdprep.txt) geschrieben. Diese Textdatei wird beim Aufruf von diskpart mit übergeben.

### HDPREP\_MBR.TXT

```
select disk 0
clean
create partition primary size=500
format fs=ntfs quick label="Hide"
active
assign letter=s
create partition primary
format fs=ntfs quick label="Windows 10"
assign letter=w
exit
```

### HDPREP\_GPT.TXT

```
select disk 0
clean
convert gpt
create partition efi size=100
format fs=fat32 quick label="System"
assign letter=s
create partition msr size=16
create partition primary
format fs=ntfs quick label="Windows"
assign letter=w
exit
```

### AUFRUF VON DISKPART

```
diskpart /s hdprep_mbr.txt    bzw.
diskpart /s hdprep_gpt.txt
```



## BEISPIELSKIPT ZUM AUFRUF VON DISKPART FÜR BIOS ODER UEFI

```
@echo off
BiosUefi_Console.exe > %TEMP%\tmpFile
set /p BIOSUEFI= < %TEMP%\tmpFile
del %TEMP%\tmpFile

if "%BIOSUEFI%"=="BIOS" ( set hdpreg=hdpreg_bios.txt)
if "%BIOSUEFI%"=="UEFI" ( set hdpreg=hdpreg_uefi.txt)

diskpart /s %hdpreg%
```

## ÜBERPRÜFUNG AUF FEHLER

In vielen Fällen kann es sinnvoll sein, einen Befehl nur dann auszuführen, wenn der vorhergehende Befehl erfolgreich war. Dies lässt sich mit der Variablen %errorlevel% überprüfen.

```
diskpart /s hdpreg.txt
if errorlevel 1 goto ende

DISM /apply-image imagefile:%wimimage% /applydir:w:\ /Index:1
if errorlevel 1 goto ende

bcdboot w:\windows
if errorlevel 1 goto Ende

exit
:Ende
```

Wenn einer der Befehle nicht erfolgreich war (errorlevel 1), wird an das Ende des Skripts gesprungen ohne die nachfolgenden Befehle auszuführen.

## PAUSE, TIMEOUT ODER COUNTDOWN

Am Ende der Installation sollte ein Neustart des Computers erfolgen. Der Neustart wird nur ausgeführt, wenn die Installation ohne Fehler durchlaufen ist. Trotzdem kann es sinnvoll sein, mit dem automatischen Neustart ein paar Sekunden zu warten, um dem Systembetreuer die Möglichkeit zu geben, an dieser Stelle abzubrechen. Dazu gibt es folgende optionalen Möglichkeiten:

### PAUSE

Beim Pause-Befehl in einem Skript wartet das System auf einen Tastendruck, bis es weiter macht. Alternativ kann mit <Strg>+C das Skript abgebrochen werden.

```
pause
```



## TIMEOUT.EXE

Das Windows-Programm timeout.exe wartet die angegebene Zeit, bis das Skript fortgeführt wird. Während dieser Zeit kann das Skript mit <Strg>+C abgebrochen werden. Das Programm ist in WinPE nicht enthalten, kann aber kopiert werden (Windows\System32\timout.exe).

```
timeout.exe 10
```

## COUNTDOWN

Alternativ kann ein eigenes kleines Batchprogramm geschrieben werden, das im Wesentlichen auf dem Ping-Befehl auf localhost beruht.

Aufruf:

```
@ECHO OFF
SETLOCAL ENABLEEXTENSIONS ENABLEDELAYEDEXPANSION
FOR /F %%# IN ('COPY /Z "%~dpf0" NUL') DO SET "CR=%%#"
FOR /L %%# IN (10,-1,1) DO (SET/P "=Neustart in %%# Sekunden.
!CR!"<NUL:
    PING -n 2 127.0.0.1 >NUL:)
```

## INSTALLATIONSSKRIPT STARTJOB.BAT

Beim folgenden Installationskript wird davon ausgegangen, dass alle verwendeten Programme und Wim-Images im gleichen Verzeichnis liegen. Im Wesentlichen sind dies folgende Dateien:

```
startjob.bat
hdprep_mbr.txt
hdprep_gpt.txt
Wim_Auswaehlen.exe
BiosUefi_Console.exe
timeout.exe
Win_Test1.wim
Win_Test2.wim
etc.
```

Es wird in das Verzeichnis gewechselt und das Skript startjob.bat ausgeführt.



---

**STARTJOB.BAT**

```
@echo off
color 70
Wim_Auswaehlen.exe

if not exist %TEMP%\imagenname.txt goto Ende
set /p WimImage= < %TEMP%\imagenname.txt
del %TEMP%\imagenname.txt

if "%WimImage%"==" " (
    echo Es wurde kein Image ausgewaehlt.
    echo.
    goto ende
)

BiosUefi_Console.exe > %TEMP%\tmpFile
set /p BIOSUEFI= < %TEMP%\tmpFile
del %TEMP%\tmpFile

echo %BIOSUEFI%

if "%BIOSUEFI%"=="BIOS" ( set hdprep=hdprep_bios.txt)
if "%BIOSUEFI%"=="UEFI" ( set hdprep=hdprep_uefi.txt)

diskpart /s %hdprep%
if errorlevel 1 goto ende

dism /apply-image imagefile:%WimImage% /applydir:w:\ /Index:1
if errorlevel 1 goto ende

bcdboot w:\windows /s s:
if errorlevel 1 goto Ende

timeout.exe 10
exit

:Ende
```

