

Handreichung  
zur Installation von Windows  
in Schulen

# Erstellen eines WinPE Live-Systems

WinPE ist ein Live-Betriebssystem auf Windows-Basis, das von CD oder USB-Stick gestartet wird. In Verbindung mit einem Imaging-Werkzeug ermöglicht es die Systemsicherung, die Systemwiederherstellung oder das Klonen von Windows-Computern.

Die vorliegende Dokumentation bezieht sich auf die Windows-Versionen 7 – 10.

## INHALT

Erstellen eines WinPE Live-Systems .....	3
Einbinden zusätzlicher Programme .....	5
WinPE auf einer Live-CD .....	7
WinPE auf einem USB-Stick .....	7
WinPE auf einer mobilen Festplatte .....	10
Erste Schritte mit dem WinPE Live-System .....	11
Anpassungen der Imagedatei boot.wim .....	13
Ausführen eines Startjobs .....	15
Übersicht: Erstellen eines WinPE-Sticks .....	16

## IMPRESSUM

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

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

Dokumentation: Georg Schlagbauer, Akademie Dillingen  
Barbara Maier, Akademie Dillingen  
Christian Maushart, Bürgernetz Dillingen  
Thomas Pickel, Maximilian Kolbe Schule, Neumarkt

E-Mail: [schlagbauer@alp.dillingen.de](mailto:schlagbauer@alp.dillingen.de)

Stand: September 2020



## VORBEMERKUNG

Mit einem Live-Betriebssystem, das sich auf einer CD oder einem USB-Stick befinden kann, ist es möglich, einen Computer zu starten, ohne auf das dort vorhandene und evtl. defekte Betriebssystem zuzugreifen. Im Notfall können auf dem Rechner liegende Daten gesichert oder eine Systemsicherung zurückgespielt werden.

WinPE steht für „Windows Preinstallation Environment“. Es ist ein Live-Betriebssystem auf Windows-Basis und ermöglicht das Ausführen vieler Windows-Programme, sofern diese keine Installation oder besondere Systemumgebung benötigen.

## ERSTELLEN EINES WINPE LIVE-SYSTEMS

Um ein WinPE Live-System zu erstellen, bietet Microsoft Hilfsprogramme an, die im Programmpaket ADK (Assessment and Deployment Kit) enthalten sind.

## INSTALLATION DES ADK

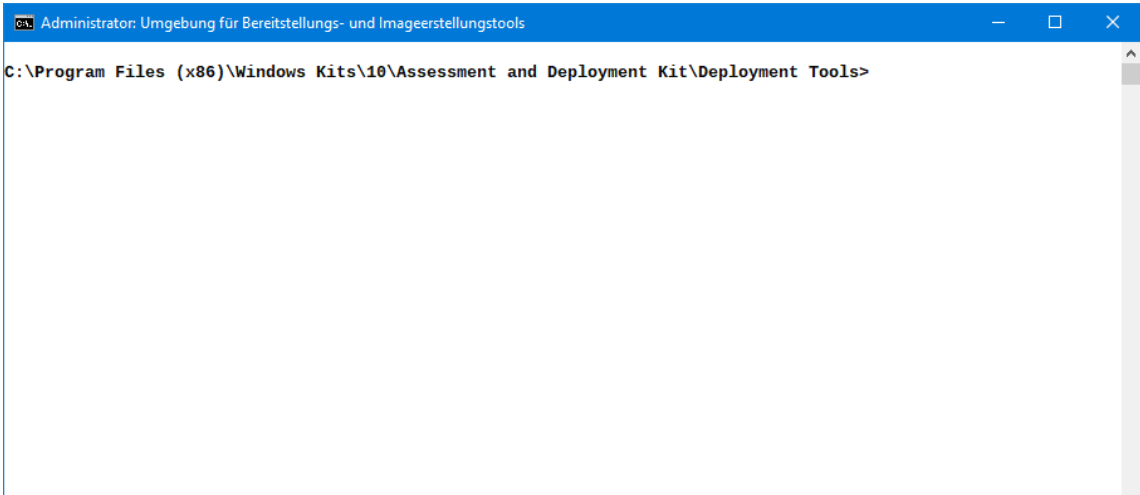
Das ADK (Assessment and Deployment Kit) wird von Microsoft zum Download angeboten (Suchbegriffe: Microsoft ADK Download; ggf. mit der gewünschten Windows-Version):

- Windows ADK für Windows 10, Versionsnummer vierstellig z.B. 2004
- Windows PE Add-On für das ADK

Aus dem ADK wird nur der Bereich „Bereitstellungstools“ benötigt. Anschließend wird noch das „Windows PE Add-On für das ADK“ installiert.

## UMGEBUNG FÜR BEREITSTELLUNGS- UND IMAGEERSTELLUNGSTOOLS

Nach der Installation der beiden Programmteile des ADK startet man die „Umgebung für Bereitstellungs- und Imageerstellungstools“ **als Administrator**. Man erhält ein Eingabefenster, das alle notwendigen Skripte bereitstellt.



```
Administrator: Umgebung für Bereitstellungs- und Imageerstellungstools
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment Tools>
```



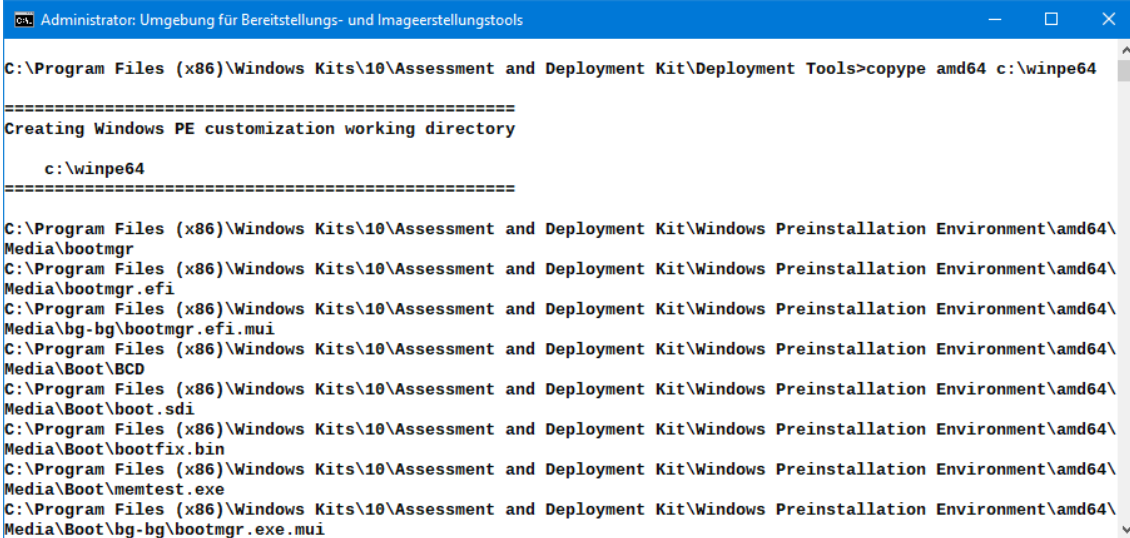
## GRUNDSTRUKTUR DES SPÄTEREN LIVE-SYSTEMS

Im ersten Schritt wird die Grundstruktur für das spätere WinPE Live-System angelegt. Dazu dient der Befehl **copype**.

```
copype <Hardwarearchitektur> <Zielverzeichnis>
copype x86 c:\winpe          bzw.
copype amd64 c:\winpe64
```

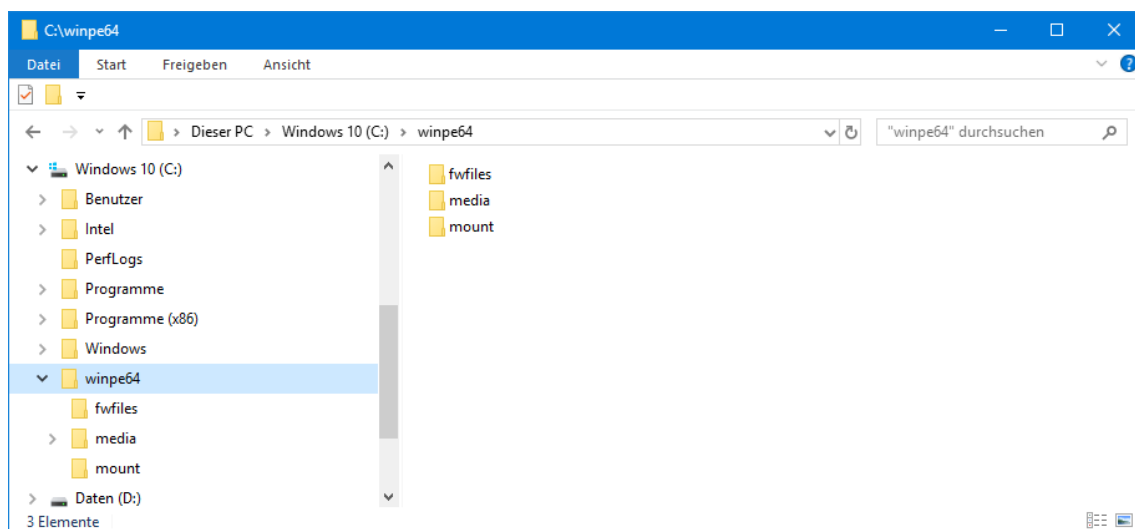
Als Hardwarearchitektur kann x86 (32-bit) oder amd64 (64-bit) gewählt werden.

Das Zielverzeichnis (z. B. C:\winpe) kann beliebig gewählt werden.



```
Administrator: Umgebung für Bereitstellungs- und Imageerstellungstools
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment Tools>copype amd64 c:\winpe64
=====
Creating Windows PE customization working directory
      c:\winpe64
=====
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\
Media\bootmgr
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\
Media\bootmgr.efi
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\
Media\bg-bg\bootmgr.efi.mui
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\
Media\Boot\BCD
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\
Media\Boot\boot.sdi
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\
Media\Boot\bootfix.bin
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\
Media\Boot\memtest.exe
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\
Media\Boot\bg-bg\bootmgr.exe.mui
```

Man erhält die nachfolgend abgebildete Verzeichnisstruktur.



Das Verzeichnis media enthält den späteren Inhalt der Live-CD bzw. des Live-USB-Sticks.

## ENTSCHEIDUNG: 32-BIT ODER 64-BIT

Die 64-Bit-Version von Windows PE enthält keinen Emulator für 32-Bit-Programme. Deshalb laufen auf einem 64-Bit-System nur 64-Bit-Programme und auf einem 32-Bit-System nur 32-Bit-Programme. Wenn man unter Windows PE ein Programm verwenden möchte, das nur in der 32-Bit-Version vorliegt, würde dies für diese Architektur sprechen.

Aktuelle Computer verfügen zwar bereits über ein UEFI-Setup (Nachfolger von BIOS), arbeiten aber manchmal noch im „BIOS-Kompatibilitätsmodus“ (CSM), in dem sich UEFI so verhält, wie das klassische BIOS. Solange diese Emulation aktiv ist, hat man die freie Wahl zwischen der 32-Bit- und 64-Bit-WinPE Architektur.

Wenn man den BIOS-Kompatibilitätsmodus abschaltet oder dieser nicht mehr unterstützt wird (z. B. Microsoft Surface), gibt es diese Wahlmöglichkeit nicht mehr: Ein 32-Bit-UEFI startet nur ein 32-Bit-System und ein 64-Bit-UEFI nur ein 64-Bit-System. 32-Bit-UEFI-Firmware gibt es nur noch sehr vereinzelt bei einigen älteren Windows-Tablets, aktuelle Systeme unterstützen immer 64-Bit.

### Hinweis:

Da inzwischen viele neue Geräte nur noch 64-Bit-UEFI unterstützen, empfiehlt sich die Erstellung der 64-Bit-Version von Windows PE.

Nähere Informationen dazu findet man auch in der Dokumentation „BIOS und UEFI – Einführung und Unterschiede“ unter <http://alp.dillingen.de/schulnetz/materialien/UEFI.pdf>

## EINBINDEN ZUSÄTZLICHER PROGRAMME

Auf dem WinPE-Live-System können zusätzliche Programme abgelegt werden (z. B. Drive-Snapshot, Virens Scanner, Skripte, etc.). Diese werden in das Verzeichnis media der vorher angelegten Struktur (z. B. C:\winpe\media) kopiert. Die Lauffähigkeit der einzelnen Programme aus dem Live-System heraus muss getestet werden.

Ebenso können aus dem Verzeichnis media überflüssige Dateien (z. B. Sprachpakete) gelöscht werden.

## DISM UND IMAGEX

Mit **ImageX** konnten Windows-Abbilddateien erfasst oder installiert werden. Seit Windows 8 wird ImageX von Microsoft nicht mehr empfohlen. Das Deployment Image Servicing and Management **dism** übernimmt alle bisher mit ImageX ausgeführten Funktionen.

ImageX ist nur noch aus Kompatibilitätsgründen vorhanden und muss normalerweise nicht auf das Live-System kopiert werden und wird in dieser Anleitung nicht mehr benutzt.

Aus dem Live-System heraus sollen mit **dism** Windows-Abbilddateien (auch WIM-Dateien oder WIM-Images) erfasst oder installiert werden. Das Programm **dism** ist im Live-System bereits enthalten. Wichtig ist, die korrekte Version (32-Bit oder 64-Bit) zu verwenden.

## DRIVESNAPSHOT

Das Imaging-Programm **DriveSnapshot** kann problemlos von einem Windows-Live-System gestartet werden. Es besteht nur aus einer EXE-Datei, die als 32-Bit- und 64-Bit-Version erhältlich ist.



## GRAPHISCHER DATEIMANAGER

Wer gerne mit einem graphischen Dateimanager arbeitet, kann ggf. einen der Dateimanager **A43**, **A56** oder **Expicula** einbinden (auf die 32- oder 64-Bit-Version achten). Alternativ hat man über den „Datei öffnen“-Dialog des Texteditors **notepad** auch eine grafische Oberfläche zur Orientierung im Dateisystem.

## NETZWERKMANAGER

Netzwerkeinstellungen können alle auf der Kommandozeile über das enthaltene Programm **netsh** erfolgen. Wer gerne mit einem grafischen Tool arbeitet, kann den **PE-Netzwerk-Manager** ausprobieren.

## VIRENSCANNER

Der **Microsoft Safety Scanner** ist unter Windows PE lauffähig. Es muss die richtige Version (32- bzw. 64-Bit) gewählt werden.

Die portable Version des Virenschanners **ClamWin** (vermutlich nur als 32-Bit-Version erhältlich) lässt sich von Windows PE aus starten. Beim ersten Aufruf werden aktuelle Virensignaturen aus dem Internet geladen.



## WINPE AUF EINER LIVE-CD

Mit dem Skript **makewinpemedia** wird die im Unterverzeichnis `media` vorbereitete Struktur in ein ISO-Image gepackt, das anschließend leicht in einer virtuellen Maschine als Startmedium verwendet werden kann.

```
makewinpemedia /iso <Arbeitsverzeichnis> <Ziel>
```

```
makewinpemedia /iso c:\winpe64 c:\winpe64\winpe64.iso
```

Der Name der Zieldatei und des Arbeitsverzeichnisses (hier `C:\winpe64` bzw. `C:\winpe64\winpe64.iso`) kann beliebig gewählt werden.

## WINPE AUF EINEM USB-STICK

Auf einem USB-Stick wird Windows-PE als „embedded System“ zusammen mit den Bootmechanismen (BIOS oder UEFI) eingerichtet. Das eigentliche WinPE-Live-System befindet sich in der Datei `boot.wim`, die erst beim Starten des Sticks als Laufwerk X gemountet wird.

Anstatt mit dem Skript `makewinpemedia` kann ein USB-Stick mit einem WinPE-Live-System auch manuell erstellt werden.

## ERSTELLEN EINES WINPE-LIVE-USB-STICKS MIT MAKEWINPEMEDIA

```
makewinpemedia /ufd <Arbeitsverzeichnis> <Ziel>
```

```
makewinpemedia /ufd c:\winpe64 F:
```

Als Ziel (hier `F:`) muss der Laufwerksbuchstabe des USB-Sticks gewählt werden.

Falls das Skript mit einer Fehlermeldung abbricht, kann der Stick vor dem Ausführen des Skripts mit **diskpart** manuell formatiert werden. Es gelingt jedoch nicht mit jedem Stick, diesen bootfähig zu machen.

Das Skript `makewinpemedia` formatiert den USB-Stick mit FAT32. Um den Stick mit NTFS zu formatieren, kann man das Skript anpassen (`fat32` durch `ntfs` ersetzen) oder den Stick manuell erstellen.

## ENTSCHEIDUNG: FAT32 ODER NTFS

Das Skript `makewinpemedia` formatiert den USB-Stick standardmäßig mit FAT32.

FAT32 erlaubt eine maximale Partitionsgröße von 32 GB und eine Dateigröße von maximal 4 GB. Sollen auf dem USB-Stick auch (mehrere) WIM-Images abgelegt werden, kann es günstiger sein, den Stick mit NTFS zu formatieren. NTFS unterliegt diesen Beschränkungen nicht.

Startet ein PC im BIOS-Kompatibilitätsmodus, ist es egal, ob der Boot-Stick mit FAT32 oder NTFS formatiert ist. Soll der Stick auch verwendet werden, um einen PC im UEFI-Modus zu starten, so muss er mit FAT32 formatiert sein.



Bis Windows 10 Version 1609 war es nicht möglich, einen USB-Stick zu partitionieren. Man musste sich also zwischen FAT32 und NTFS entscheiden. Mit speziellen Programmen oder unter Linux war das Partitionieren eines USB-Sticks schon immer möglich, Windows konnte aber immer nur auf die erste Partition zugreifen.

Seit Windows 10 Version 1703 ist es möglich, auf einem USB-Stick mehrere Partitionen einzurichten bzw. anzusprechen. Man kann also zwei Partitionen anlegen:

- FAT32-Partition (Größe 2,0 GB) mit allen Boot-Dateien
- NTFS-Partition mit den WIM-Images und/oder andere Dateien

Damit hat man die Möglichkeit, einen WinPE-Stick so vorzubereiten, dass dieser sowohl BIOS- als auch UEFI-Systeme bootet und es existiert eine weitere Partition um dort große WIM-Images abzulegen.

Um dies nutzen zu können, muss man folgendermaßen vorgehen:

- Die Einrichtung des WinPE-Sticks erfolgt auf einem Windows-10 PC ab Version 1703.
- Es muss eine möglichst aktuelle ADK-Version verwendet werden, damit auch das spätere Live-System zwei Partitionen auf dem USB-Stick ansprechen kann.
- Der Stick muss unter Windows mit einer FAT32-Partition (aktive Partition) und einer NTFS-Partition eingerichtet werden (z. B. mit diskpart).
- Anschließend kann das Skript makewinpemedia ausgeführt werden oder der Stick kann manuell erstellt werden. Das Skript makewinpemedia greift dabei nur auf die erste Partition des Sticks zu und richtet diese ein.

## MANUELLES ERSTELLEN EINES WINPE-LIVE-USB-STICKS

Folgende Skripte erstellen die zwei verschiedenen Varianten von USB-Sticks.

### WINDOWS 10 BIS VERSION 1609 (EINE PARTITION)

```
diskpart
```

```
DISKPART> list disk
```

```
DISKPART> select disk 1
```

```
DISKPART> clean
```

```
DISKPART> convert mbr
```

```
DISKPART> create partition primary
```

```
DISKPART> active
```

```
DISKPART> format fs=fat32 quick
```

```
DISKPART> assign letter=u
```

```
DISKPART> exit
```

Die Disk-Nr. muss eventuell angepasst werden.

Der gesamte USB-Stick wird gelöscht.

Das Partitionierungsschema wird auf MBR festgelegt.

Eine Partition wird erzeugt.

Die ausgewählte Partition wird als aktiv markiert.

Formatieren mit FAT32 (Alternative: NTFS)

Der Stick erhält den Laufwerksbuchstaben U.





---

**WINDOWS 10 AB VERSION 1703 (ZWEI PARTITIONEN)**

diskpart

```
DISKPART> list disk
DISKPART> select disk 1           Die Disk-Nr. muss eventuell angepasst werden.
DISKPART> clean
DISKPART> convert mbr
DISKPART> create partition primary size=2000
DISKPART> active
DISKPART> format fs=fat32 quick
DISKPART> assign letter=u
DISKPART> create partition primary
DISKPART> format fs=ntfs quick
DISKPART> assign letter=v
DISKPART> exit
```

**Hinweis:**

Durch die Zuweisung des Laufwerksbuchstaben „U“ mit dem Befehl „assign letter=u“ erspart man sich den Neustart und kann den Stick sofort ansprechen. Die Zuweisung ist nur temporär. Es ist nicht möglich, einen USB-Stick so zu konfigurieren, dass er bei jedem Windows-PC den gleichen Laufwerksbuchstaben erhält.

---

**MEDIA-VERZEICHNIS KOPIEREN**

Auf den vorbereiteten USB-Stick wird der gesamte Inhalt des media-Verzeichnisses C:\WinPE\media kopiert:

```
xcopy c:\winpe64\media\*.* /s u:\
```

Anschließend sollten alle zusätzlich eingebundenen Programme auf dem bootfähigen USB-Stick zur Verfügung stehen.



## WINPE AUF EINER MOBILEN FESTPLATTE

Windows PE lässt sich auch auf einer mobilen Festplatte installieren. Der Unterschied zur Installation auf einem USB-Stick ist dabei, dass die Datei boot.wim nun entpackt als Verzeichnisstruktur auf der Festplatte liegt und der Bootsektor entsprechend angepasst wird.

Ein USB Stick wird von Windows 10 in der Regel als Wechseldatenträger angesprochen. Externe Festplatten werden als lokale Laufwerke angesprochen und können ebenfalls bootfähig konfiguriert werden.

### EINRICHTEN DER MOBILEN FESTPLATTE:

```
diskpart
```

```
DISKPART> list disk
```

```
DISKPART> select disk 1
```

Die Disk-Nr. muss eventuell angepasst werden.

```
DISKPART> clean
```

```
DISKPART> convert mbr
```

```
DISKPART> create partition primary size=2000
```

```
DISKPART> active
```

```
DISKPART> format fs=fat32 quick label=WinPE
```

```
DISKPART> assign letter=u
```

```
DISKPART> create partition primary
```

```
DISKPART> active
```

```
DISKPART> format fs=ntfs quick label=Daten
```

```
DISKPART> assign letter=v
```

```
DISKPART> exit
```

### INSTALLATION VON WINPE AUF DER MOBILEN FESTPLATTE

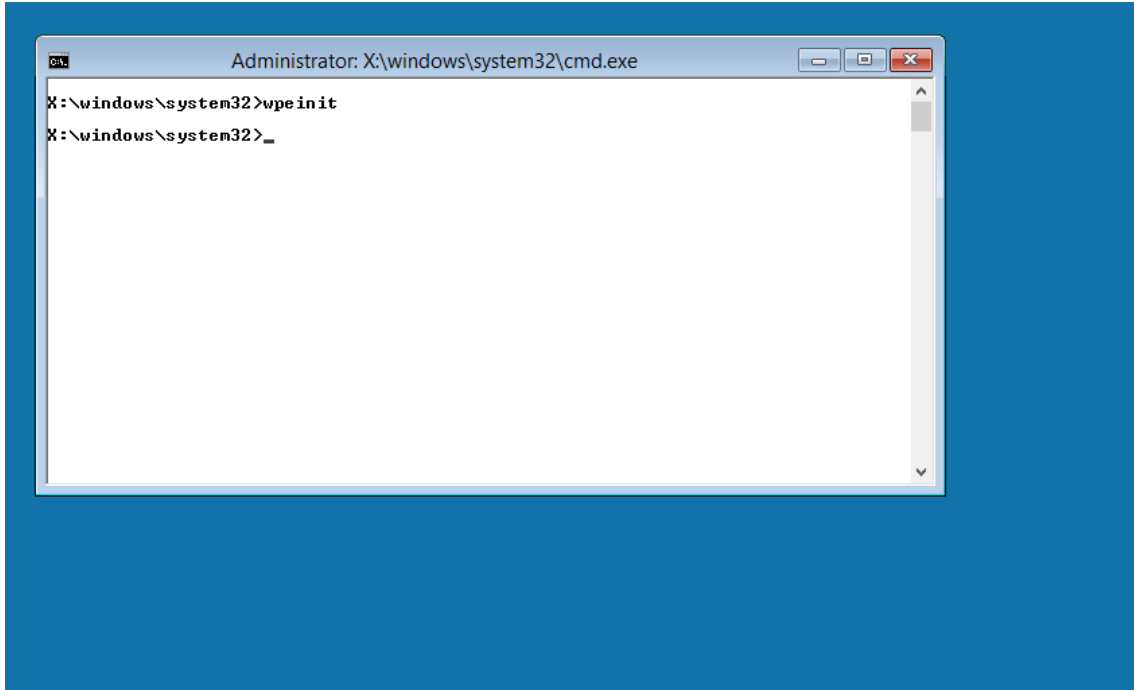
```
dism /apply-image /imagefile:c:\winpe64\media\sources\boot.wim  
/index:1 /applydir:u:\
```

```
bcdboot u:\windows /s u: /f all
```



## ERSTE SCHRITTE MIT DEM WINPE LIVE-SYSTEM

Nach dem Start präsentiert sich WinPE mit einer Kommandozeilenoberfläche. Es ist zwar eine grafische Oberfläche gestartet, alle Programme werden jedoch von dieser Kommandozeile aus aufgerufen.



## DATEISYSTEM

Zunächst sollte man versuchen, sich im Dateisystem zurechtzufinden und den Inhalt aller vorhandenen Laufwerke betrachten:

```
dir c: /a    (Mit der Option /a zeigt dir auch versteckte Dateien an)
dir d: /a
...
dir x: /a
...
```

Alternativ kann man sich auch über einen grafischen Dateimanager, einen bereitgestellten graphischen Dateimanager oder über den „Öffnen-Dialog“ des Texteditors notepad orientieren.

Üblicherweise findet man unter C:\ die Bootpartition der ersten Festplatte. Ab D:\ findet man weitere Partitionen sowie das Live-System (CD oder USB-Stick) auf dem die Programme oder Skripte liegen, die man bei der Vorbereitung im media-Verzeichnis abgelegt hat. X:\ enthält den Inhalt des gemounteten WIM-Images boot.wim.

## AUFRUF VON PROGRAMMEN

Windows Programme oder Programme, die in den Systemverzeichnissen X:\windows bzw. X:\windows\system32 enthalten sind, kann man von der Kommandozeile von jedem Pfad aus durch Eingabe des Programmnamens starten.

**Beispiele:**

notepad	Texteditor
taskmgr	Taskmanager
xcopy	Kopieren von Dateien
regedit	Registrierungseditor

Bei anderen Programmen (z. B. snapshot.exe) gibt man den Laufwerksbuchstaben mit an oder wechselt vorher in das richtige Verzeichnis.

```
f:\a56.exe
f:\snapshot64.exe
f:\imagex.exe
```

**NETZWERKZUGRIFF**

Im WinPE-Live-System stehen die üblichen Windows-Programme zur Verfügung, so dass die Verbindung zum Netzwerk getestet oder auf eine Freigabe zugegriffen werden kann.

```
ipconfig           Anzeige der IP-Konfiguration
ping              ICMP Anfrage
net use n: \\10.36.16.10\windows   Verbindung zu einem Netzlaufwerk
net use n: \\10.36.16.10\windows /user:10.36.16.10\admin 12345
```

Die Netzlaufwerkszuordnung vom Live-System aus gelingt meist nur, wenn der Benutzername und das Passwort mit angegeben werden.

**MANUELLE VERGABE EINER IP-ADRESSE**

Falls im Netzwerk kein DHCP-Server vorhanden ist, kann eine IP-Adresse auch auf der Kommandozeile vergeben werden.

```
netsh              Interaktiver Modus von netsh
netsh interface show interface   Schnittstellen anzeigen

netsh interface ipv4 add address "Ethernet" 192.168.0.10 255.255.255.0
netsh interface ipv4 add address "Ethernet" gateway=192.168.0.1 gwmetric=2
```

Der Name der Verbindung „Ethernet“ muss eventuell angepasst werden.

**DEUTSCHES TASTATURLAYOUT**

Standardmäßig ist vermutlich ein amerikanisches Tastaturlayout aktiv. Auf die deutsche Tastaturbelegung kann man umschalten:

```
wpeutil SetKeyboardLayout 0407:00000407
start cmd.exe
```



Der Befehl `start cmd.exe` öffnet ein neues Fenster, in dem die deutsche Tastaturbelegung zur Verfügung steht. Wie man dauerhaft nach dem Start die deutsche Tastaturbelegung einstellt, ist im Kapitel „Optionale Ergänzungen“ beschrieben.

## BEENDEN VON WINPE

Bootmedium entfernen und **exit** eingeben. Der PC startet neu.

## ANPASSUNGEN DER IMAGEDATEI BOOT.WIM

Das eigentliche Windows PE-System, das im Betrieb unter dem Laufwerksbuchstaben X erscheint, ist als WIM-Imagedatei `boot.wim` auf dem Livesystem abgelegt. Um das Livesystem anzupassen (z.B. um zusätzliche Treiber hinzuzufügen oder von vorneherein ein deutsches Tastaturlayout auszuwählen), ist es notwendig die Datei `boot.wim` anzupassen.

Änderungen an der WIM-Imagedatei `boot.wim` benötigen drei Schritten:

### MOUNTEN DER WIM-IMAGEDATEI BOOT.WIM

```
dism /mount-wim /wimfile:c:\winpe64\media\sources\boot.wim /index:1  
/mountdir:c:\winpe64\mount
```

### DURCHFÜHREN DER GEWÜNSCHTEN ÄNDERUNGEN

Tastaturlayout oder Treiber hinzufügen, ...

### UNMOUNTEN DER WIM-IMAGEDATEI UND BESTÄTIGEN DER ÄNDERUNGEN

```
dism /unmount-wim /mountdir:c:\winpe64\mount /commit
```

#### Hinweis:

Die Option `/commit` veranlasst die Übernahme der Veränderungen in die WIM-Imagedatei. Der Befehl funktioniert nur zuverlässig, wenn vorher alle Fenster geschlossen werden, die auf das Mountverzeichnis verweisen. Falls hier ein Fehler unterläuft, hilft eventuell folgender Befehl:

```
dism /cleanup-wim
```

## INTEGRATION ZUSÄTZLICHER TREIBER

Mit einem Live-System sollte man mindestens auf die Festplatten und das Netzwerk zugreifen können. Bei neuen Computern können dazu im ungünstigsten Fall Treiber erforderlich sein, die in das verwendete Boot-Image integriert werden müssen. Diese müssen im INF-Format vorliegen und werden dazu ggf. vorher entpackt und in ein Verzeichnis (z.B. `C:\Treiber`) kopiert. Eine Möglichkeit, die nötigen Treiber aus einer bereits fertig eingerichteten Windows-Installation zu extrahieren, ist im Anhang der Handreichung „Vorbereiten einer Windows-Installation zum Klonen“ beschrieben.



```
dism /mount-wim /wimfile:c:\winpe64\media\sources\boot.wim /index:1
      /mountdir:c:\winpe64\mount

dism /image:c:\winpe64\mount /Add-Driver /driver:c:\Treiber /recurse

dism /unmount-wim /mountdir:c:\winpe64\mount /commit
```

Durch Angabe des Ordners „C:\Treiber“ und der Option „/recurse“ werden alle für Windows 10 signierten Treiber aus diesem Ordner und dessen Unterordnern installiert.

Die Option /ForceUnsigned erzwingt die Integration eines nicht signierten Treibers xy.inf aus dem Verzeichnis C:\Treiber\_W10 in das WIM-Image:

```
dism /image:c:\winpe64\mount /Add-Driver
      /driver:c:\Treiber_W10\xy.inf /forceunsigned
```

Die hinzugefügten Treiber befinden sich anschließend im Verzeichnis \Windows\Inf und erhalten die Namen Oem1.inf, Oem2.inf, etc..

## DEUTSCHES TASTATURLYOUT

Im Boot-Image boot.wim kann ein deutsches Tastaturlayout voreingestellt werden.

Die Umstellung des Zeichensatzes geschieht in folgenden Schritten:

```
dism /mount-wim /wimfile:c:\winpe64\media\sources\boot.wim /index:1
      /mountdir:c:\winpe64\mount

dism /image:c:\winpe64\mount /add-package /packagepath:"c:\Program
Files (x86)\Windows Kits\10\Assessment and Deployment
Kit\Windows Preinstallation Environment\amd64\WinPE_OC\de-
de\lp.cab"

dism /image:c:\winpe64\mount /set-allintl:de-DE

dism /unmount-wim /mountdir:c:\winpe64\mount /commit
```

### Hinweis:

Jeder dism Befehl muss ohne Zeilenumbruch eingegeben werden. Die Eingabe langer Befehlsfolgen auf der Kommandozeile gelingt oft nicht fehlerfrei. Hilfestellung beim Erstellen der Befehlszeilen liefern:

- Automatische Befehlszeilenergänzung (<TAB>-Taste) und die Befehlshistorie (Pfeiltasten)
- Umgebungsvariable als Abkürzungen setzen
- Befehle in Skripten zusammenfassen

### Beispiel: Automatische Ergänzung:

```
dism /image: c:\wi <tab>
```

Wenn man in einem Befehl vor der Pfadangabe (vor c:) ein Leerzeichen lässt (das man natürlich vor der Ausführung des Befehls erst wieder entfernen muss), funktioniert die automatische Ergänzung durch mehrfaches Drücken der <TAB>-Taste.

### Beispiel: Setzen von Variablen für komplexe Pfade



```
set pfad="c:\winpe64\mount"  
echo %pfad%
```

## AUSFÜHREN EINES STARTJOBS

Werden mit dem Live-System immer wieder gleichartige Tätigkeiten durchgeführt (z. B. Klonen von PCs), bietet es sich an, die dazu nötigen Befehle in einer Batchdatei automatisiert auszuführen.

Diese Batchdatei (z.B. startjob.bat) legt man wie die externen Programme in das media-Verzeichnis oder in das Wurzelverzeichnis des USB-Sticks.

### BEISPIEL FÜR EINE BATCHDATEI STARTJOB.BAT

```
@echo off  
cls  
color f0
```

Standardmäßig meldet sich WinPE nach dem Start mit einer Kommandozeile im Laufwerk X: und wartet auf eine Eingabe. Soll die vorbereitete Batchdatei startjob.bat automatisch ausgeführt werden, ist es erforderlich, das Startskript startnet.cmd von WinPE (\windows\system32\startnet.cmd) anzupassen:

```
dism /mount-wim /wimfile:c:\winpe64\media\sources\boot.wim /index:1  
/mountdir:c:\winpe64\mount
```

```
notepad c:\winpe64\mount\windows\system32\startnet.cmd
```

In der Datei startnet.cmd findet man nur den Startbefehl wpeinit. Der Aufruf von startjob.bat könnte einfach ergänzt werden, wenn man den Laufwerksbuchstaben des USB-Sticks kennen würde. Dieser Laufwerksbuchstabe kann aber variieren, so dass eine Prüfung notwendig wird:

```
wpeinit  
@echo off  
for %%q in (c,d,e,f,g,h,i,j) do if exist %%q:\startjob.bat set lw=%%q:  
%lw%  
call startjob.bat
```

Nach dieser Änderung wird startnet.cmd gespeichert, Notepad beendet und das Image geschlossen.

```
dism /unmount-wim /mountdir:c:\winpe64\mount /commit
```

Das Skript startjob.bat sollte nun bei jedem Start von WinPE automatisch ausgeführt werden. Vorteil dieser Lösung ist, dass sich dieses Script später modifizieren lässt, ohne dass man das Boot-Image erneut verändern muss.



## ÜBERSICHT: ERSTELLEN EINES WINPE-STICKS

Es wird als Administrator in der „Umgebung für Bereitstellungs- und Imageerstellungstools“ gearbeitet.

### GRUNDSTRUKTUR

```
copy c:\winpe64
```

### SPRACHDATEIEN LÖSCHEN

```
for /d %i in ("c:\winpe64\media\*-*.") do rmdir %i /s/q
```

### DEUTSCHE TASTATURBELEGUNG

```
@echo off
echo Installation der deutschen Tastatur
set wimfile=c:\winpe64\media\sources\boot.wim
set mountdir=c:\winpe64\mount
set package="c:\Program Files (x86)\Windows Kits\10\Assessment and
Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\de-
de\lp.cab"
dism /mount-wim /wimfile:%wimfile% /index:1 /mountdir:%mountdir%
dism /image:%mountdir% /add-package /packagepath:%package%
dism /image:%mountdir% /set-allintl:de-DE
dism /unmount-wim /mountdir:%mountdir% /commit
Pause
```

### AUTOMATISCHE SUCHE UND AUSFÜHREN VON STARTJOB.BAT

```
set wimfile=c:\winpe64\media\sources\boot.wim
set mountdir=c:\winpe64\mount
dism /mount-wim /wimfile:%wimfile% /index:1 /mountdir:%mountdir%

notepad %mountdir%\windows\system32\startnet.cmd

wpeinit
@echo off
for %%q in (c,d,e,f,g,h,i,j) do if exist %%q:\startjob.bat set lw=%%q:
%lw%
call startjob.bat

dism /unmount-wim /mountdir:%mountdir% /commit
```

### STARTJOB.BAT MIT NETZWERKZUGRIFF AN DER ALP DILLINGEN





```
@echo off
color F0
echo.
echo.
echo Verbindung ins Netz wird hergestellt:
echo.
net use n: \\10.36.104.24\Vorlagen /user:10.36.104.24\Lehrer Lehrer
if errorlevel 1 goto Ende

n:
cd Wim-Images
call startjob.bat

:Ende
```

---

#### GEGEBENENFALLS: VORBEREITUNG EINES USB-STICKS MIT DISKPART

```
diskpart

DISKPART> list disk
DISKPART> select disk x
DISKPART> clean
DISKPART> convert mbr
DISKPART> create partition primary size=2000
DISKPART> active
DISKPART> format fs=fat32 quick label=WinPE
DISKPART> assign letter=u
DISKPART> create partition primary
DISKPART> format fs=ntfs quick label=Daten
DISKPART> assign letter=v
DISKPART> exit
```

---

#### KOPIEREN DER DATEIEN MIT MAKEWINPEMEDIA

```
makewinpemedi /ufd c:\winpe64 u:
```

