

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
Integration zusätzlicher Treiber .....	14
Deutsche Tastatur .....	14
Ausführen eines Startjobs .....	15
Zusammenfassung: Erstellen eines WinPE-Sticks .....	17

## 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](mailto:schlagbauer@alp.dillingen.de)  
Stand: Juli 2018



## 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 ggf. defekte Betriebssystem zuzugreifen. Im Notfall können auf dem Rechner liegende Daten gesichert oder eine Systemsicherung zurückgespielt werden.

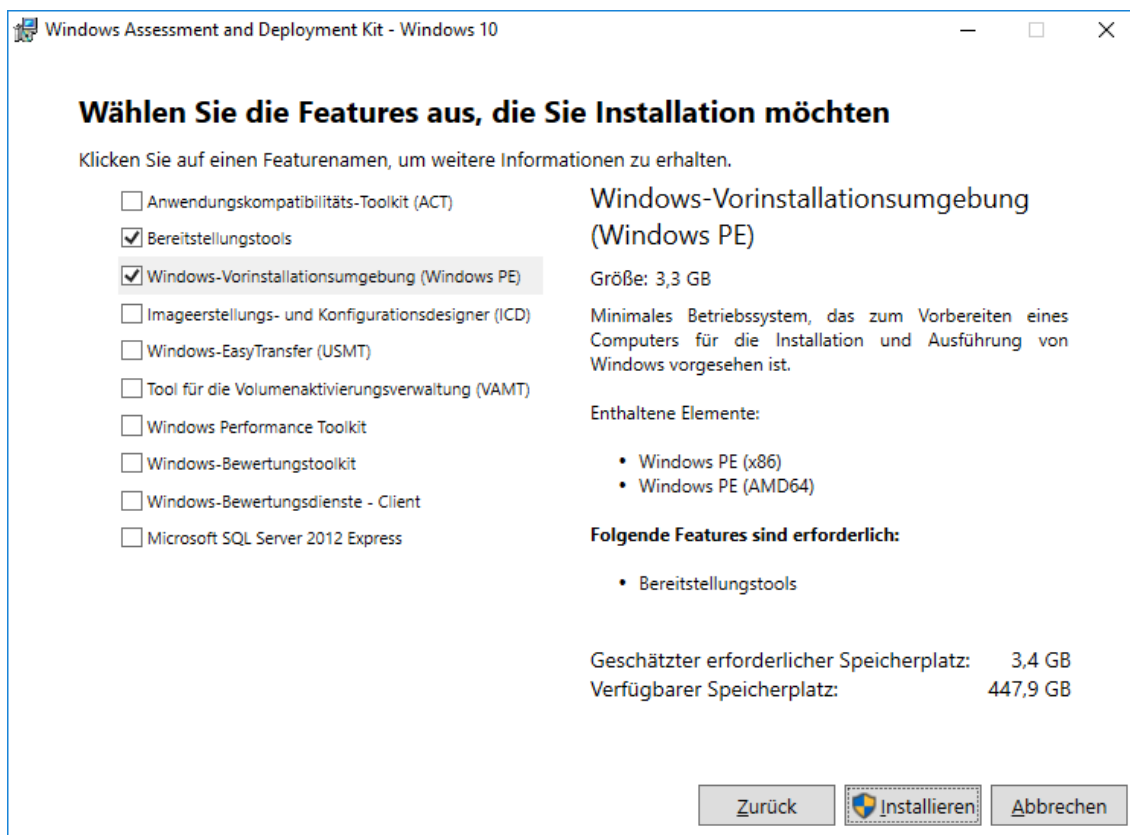
WinPE steht für „Windows Pre-Install Environment“. Es ist ein Live-Betriebssystem auf Windows-Basis und ermöglicht das Ausführen vieler Windows-Programme, sofern diese keine Installation oder besondere Systemumgebung benötigen.

## 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 VON ADK

ADK (Assessment and Deployment Kit) wird von Microsoft zum Download angeboten (Suchbegriffe: Microsoft ADK Download; ggf. mit der gewünschten Windows-Version). Man erhält zunächst das Programm adksetup.exe. Dieses startet den eigentlichen Download.

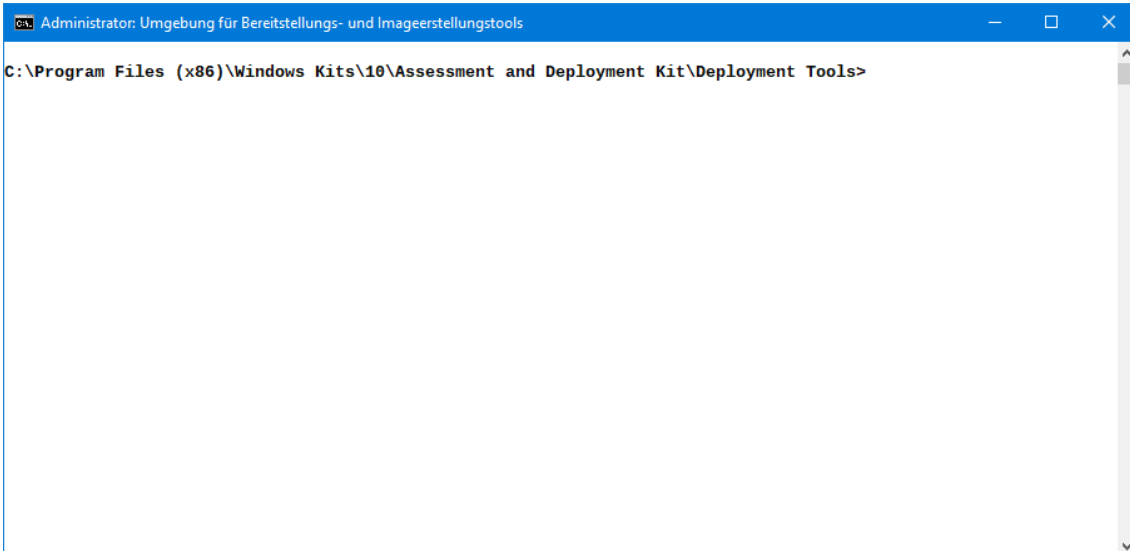


Für das Live-System genügen die beiden Programmteile „Bereitstellungstools“ und „Windows PE“.

## UMGEBUNG FÜR BEREITSTELLUNGS- UND IMAGEERSTELLUNGSTOOLS

Nach der Installation der beiden Programmteile von ADK startet man die „Umgebung für Bereitstellungs- und Imageerstellungstools“.

Man erhält ein Eingabefenster, das alle notwendigen Skripte bereitstellt.



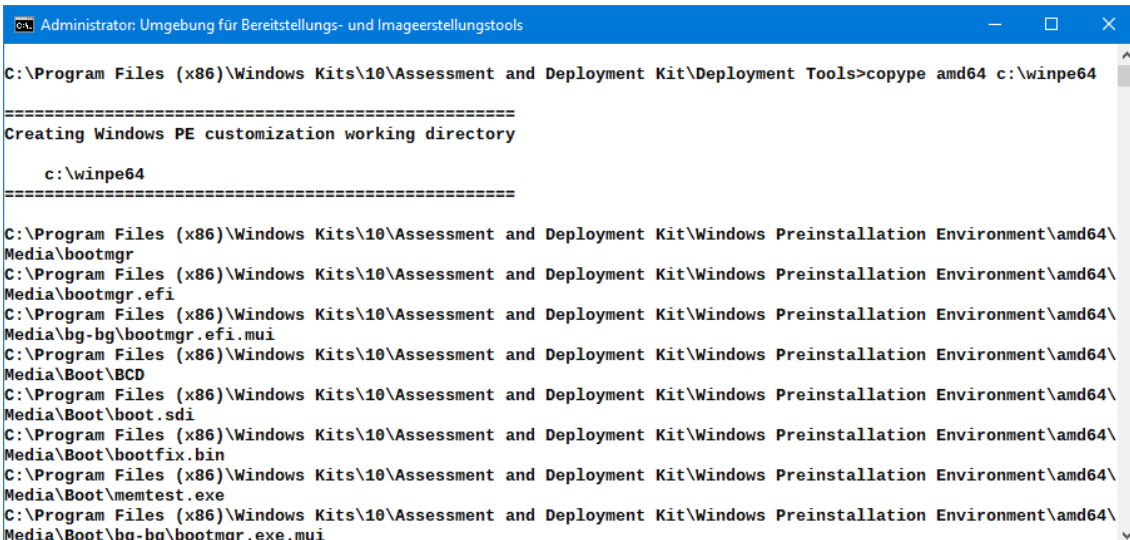
```
Administrator: Umgebung für Bereitstellungs- und Imageerstellungstools
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment Tools>
```

## 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
```

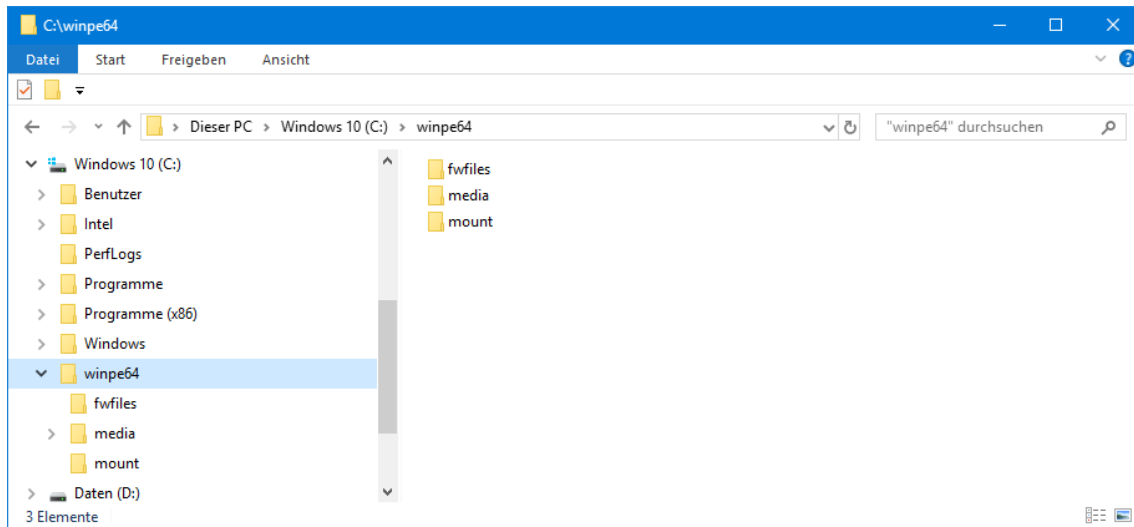
Als Hardwarearchitektur kann x86 (32-bit) oder amd64 (64-bit) gewählt werden. Das Zielverzeichnis (z. B. c:\winpe) kann beliebig gewählt werden.



```
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 ein unter WinPE 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 meist noch im „BIOS-Kompatibilitätsmodus“ (CSM), in dem sich UEFI so verhält, wie das klassische BIOS. Solange die so ist, hat man die freie Wahl zwischen 32-bit- und 64-bit-WinPE. Anders ist es, wenn man den BIOS-Kompatibilitätsmodus abschaltet oder wenn dieser nicht mehr unterstützt wird (z. B. Microsoft Surface). 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 bei einigen älteren Windows-Tablets, aktuelle Systeme unterstützen immer 64-bit.

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. ImageX, Drive-Snapshot, Virenschanner, 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.

## IMAGEX

Aus dem Live-System heraus sollen später mit **dism** oder **ImageX** Windows-Abbilddateien (WIM-Dateien) erfasst oder installiert werden. Das Programm **dism** ist im Live-System bereits enthalten, das Programm **imagex.exe** kann in das Verzeichnis **media** kopiert werden. Wichtig ist, die korrekte Version (32-bit oder 64-bit) zu verwenden. Die 64-bit-Version von **imagex.exe** findet man z. B. im Verzeichnis

```
Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\  
Deployment Tools\amd64\DISM
```

Zu **ImageX** gibt es auch eine Ergänzung mit grafischer Oberfläche (**GImageX**). Diese ist kostenlos zum Download erhältlich.

## KOPIEREN VON IMAGEX.EXE

```
xcopy "c:\Program Files (x86)\Windows Kits\10\Assessment and  
Deployment Kit\Deployment Tools\amd64\DISM\imagex.exe"  
c:\winpe64\media
```

## 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 von **notepad** auch eine grafische Oberfläche zur Orientierung im Dateisystem.

## NETZWERKMANAGER

Netzwerkeinstellungen können natürlich alle auf Kommandozeile erfolgen. Wer gerne mit einem grafischen Tool arbeitet, kann den PE-Netzwerk-Manager ausprobieren.

## VIRENSCANNER

Der **Microsoft Safety Scanner** ist von 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 die 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 auf CD gebrannt werden kann. Das Skript `makewinpemedia` verwendet dabei das Programm `oscdimg.exe`, mit dem unter Windows 7 und WAIK die Live-CD erstellt wurde. Das Verfahren ist also im Prinzip gleich geblieben.

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

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

Der Name der Zielfeile (hier `c:\winpe64\winpe64.iso`) kann beliebig gewählt werden.

Anschließend wird das ISO-Image auf CD gebrannt.

## WINPE AUF EINEM USB-STICK

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

Die Erstellung eines WinPE-USB-Sticks wird durch das Skript `makewinpemedia` unterstützt. Alternativ dazu kann der Live-USB-Stick 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 das Laufwerk 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. Wenn man lieber NTFS hätte, kann man das Skript anpassen (`fat32` mit `ntfs` ersetzen) oder den Stick – wie anschließend beschrieben – manuell erstellen.



## ENTSCHEIDUNG: FAT32 ODER NTFS

Das Skript makewinpemedial.cmd formatiert den USB-Stick standardmäßig mit fat32.

Fat32 erlaubt eine maximale Partitionsgröße von 32 GB und eine Dateigröße vom maximal 4 GB auf dem Stick. Sollen auf dem USB-Stick auch WIM-Images abgelegt werden, kann es günstiger sein, den Stick mit NTFS zu formatieren. NTFS kennt diesbezüglich keine Beschränkungen.

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.

Unter Windows (bis Windows 10 Version 1609) war es nicht möglich, einen USB-Stick zu partitionieren. Man musste sich also zwischen FAT32 und NTFS entscheiden. (PS: 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 GB) mit allen Boot-Dateien
- NTFS-Partition, auf der die Images oder andere Daten liegen

Damit hat man die Möglichkeit, einen WinPE-Stick so vorzubereiten, dass dieser sowohl BIOS- als auch UEFI-Systeme bootet und man hat eine weitere Partition, um große Images dort 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 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 makewinpemedial ausgeführt werden oder der Stick kann manuell erstellt werden. Das Skript makewinpemedial greift nur auf die erste Partition des Sticks zu und richtet diese ein.

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

Das Skript makewinpemedial.cmd formatiert den USB-Stick mit fat32 und erlaubt deshalb nur eine Dateigröße vom maximal 4 GB auf dem Stick. Sollen auf dem USB-Stick auch WIM-Images abgelegt werden, ist es günstiger, den Stick mit NTFS zu formatieren oder wie oben beschrieben eine zweite Partition mit ntfs einzurichten. Dazu wird der USB-Stick – wie nachfolgend beschrieben – mit diskpart eingerichtet und die media-Dateien anschließend kopiert.





## USB-STICK FORMATIEREN

Die meisten USB-Sticks sind zunächst nicht bootfähig und müssen deshalb entsprechend vorbereitet werden. Dazu gibt es unterschiedliche Werkzeuge, z. B. das Kommandozeilentool **diskpart**, das auf jedem Windows-Computer vorhanden ist.

```
diskpart
```

```
DISKPART> list disk
DISKPART> select disk 1
DISKPART> clean
DISKPART> create partition primary
DISKPART> active
DISKPART> format fs=ntfs quick
DISKPART> assign letter=u
DISKPART> exit
```

Die Disk-Nr. muss ggf. angepasst werden.  
Damit wird der gesamte USB-Stick gelöscht.  
Eine Partition wird erzeugt.  
Die ausgewählte Partition wird als aktiv markiert.  
Formatieren mit NTFS (Alternative: FAT32)  
Der Stick erhält einen Laufwerksbuchstaben.

oder bei neueren Windows-Versionen

```
diskpart
```

```
DISKPART> list disk
DISKPART> select disk 1
DISKPART> clean
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
```

Die Disk-Nr. muss ggf. angepasst werden.

**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 hat nur eine vorübergehende Bedeutung. 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. Dies kann man mit xcopy auf Kommandozeile oder mit dem Windows-Explorer durchführen.

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

Anschließend sollte der USB-Stick bootfähig sein.



## 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 als entpacktes System auf der Festplatte liegt.

Seit der Windows 10-Version 1703 lässt sich ein USB-Stick wie eine Festplatte behandeln, so dass dieses Verfahren auch für USB-Sticks funktioniert.

### EINRICHTEN DER MOBILEN FESTPLATTE:

```
diskpart
```

```
DISKPART> list disk
```

```
DISKPART> select disk 1
```

Die Disk-Nr. muss ggf. angepasst werden.

```
DISKPART> clean
```

```
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 DEM USB-STICK

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

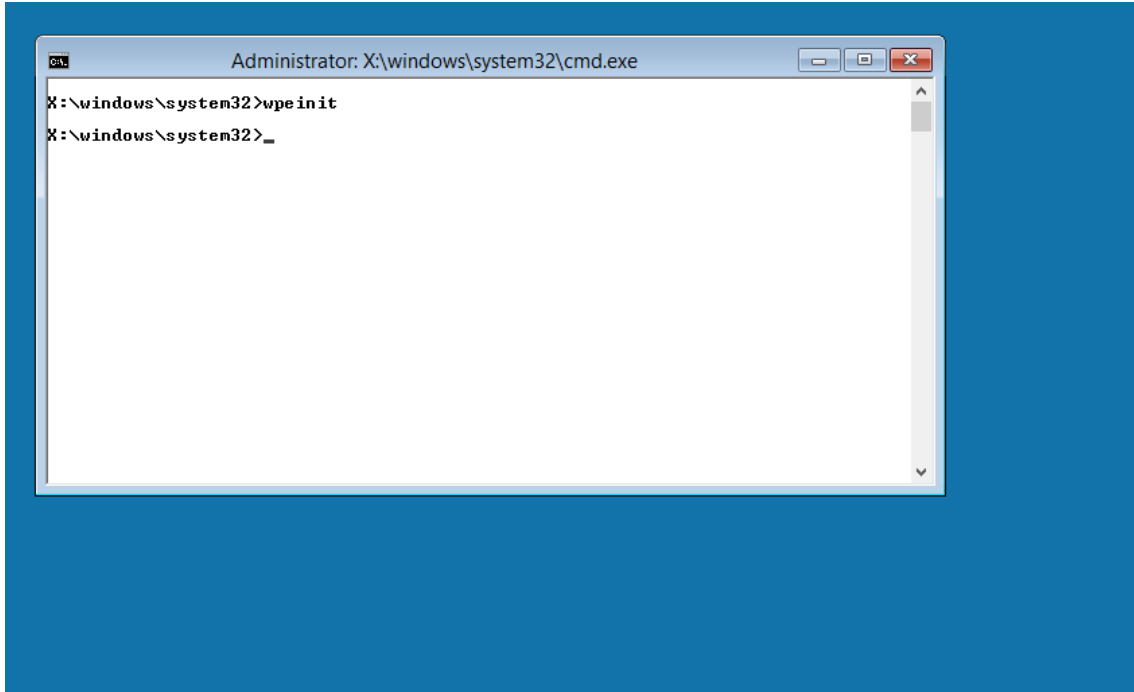
```
bcdboot u:\windows /s u: /f all
```

Einen echten Vorteil bringt dieses Verfahren in der Praxis nicht, da das Erstellen eines USB-Sticks auf diese Art deutlich länger dauert.



## ERSTE SCHRITTE MIT DEM WINPE LIVE-SYSTEM

Nach dem Start präsentiert sich WinPE mit einer Kommandozeilenoberfläche. Es ist zwar ein grafisches System gestartet, alle Programme werden jedoch auf Kommandozeile aufgerufen.



## DATEISYSTEM

Zunächst sollte man versuchen, sich im Dateisystem zurechtzufinden und den Inhalt der Laufwerke C;, D;, E;, F; und gegebenenfalls weiterer Laufwerke betrachten.

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

Alternativ kann man sich auch über einen grafischen Dateimanager oder über den „Öffnen-Dialog“ von notepad orientieren.

Üblicherweise findet man unter C: die Bootpartition der ersten Festplatte, unter D:, E:, F:, etc. 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: ist das Systemverzeichnis der Live-Umgebung (Inhalt der Datei boot.wim).

## AUFRUF VON PROGRAMMEN

Interne Programme oder Programme, die in den Systemverzeichnissen X:\windows bzw. X:\windows\system32 enthalten sind, sind von jeder Stelle aus direkt aufrufbar.

Beispiele:

notepad	Editor
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	Verbindungstest
net use n: \\10.36.16.10\windows	Verbindung zu einem Netzlaufwerk
net use n: \\10.36.16.10\windows /user: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 Kommandozeile vergeben werden.

netsh	Interaktiver Modus von netsh (Hilfe mit help oder /?)
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 ggf. angepasst werden.



## DEUTSCHES TASTATURLAYOUT

Standardmäßig ist vermutlich das amerikanische Tastaturlayout aktiv. Auf die deutsche Tastatur kann man umschalten mit

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

Der Befehl `start cmd.exe` öffnet ein neues Fenster, in dem man die deutsche Tastatur zur Verfügung hat.

Wie man dauerhaft beim Start die deutsche Tastatur 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 Imagedatei `boot.wim` auf dem Livesystem abgelegt. Um das Livesystem anzupassen (z. B. für Treiber oder deutsches Tastaturlayout), ist es notwendig die Datei `boot.wim` anzupassen.

Änderungen an der Imagedatei `boot.wim` geschehen in drei Schritten:

1. Öffnen (mount) des Image `boot.wim`
2. Durchführen der gewünschten Änderungen
3. Schließen (unmount) des Image

1. Öffnen (mount) des Image `boot.wim`

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

2. Durchführen der gewünschten Änderungen

...

3. Schließen (unmount) des Image

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

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

```
dism /cleanup-wim
```



## INTEGRATION ZUSÄTZLICHER TREIBER

Mit einem Live-System sollte man mindestens auf die Festplatten und auf das Netzwerk zugreifen können. Bei neuen Computern können dazu Treiber erforderlich sein, die in das verwendete Boot-Image boot.wim integriert werden. Die Treiber müssen im inf-Format vorliegen und werden dazu ggf. vorher entpackt und in ein Verzeichnis (z. B. C:\Treiber) kopiert.

```
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.

Um einen nicht signierten Treiber in das WIM-Image zu integrieren, kann mit der Option /ForceUnsigned nachgeholfen werden, z. B.

```
dism /image:c:\winpe\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.

## DEUTSCHE TASTATUR

Im Boot-Image boot.wim kann der deutsche Zeichensatz 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
```



## HINWEISE ZUR EINGABE

Jeder dism-Befehl steht in einer Zeile.

Die Eingabe langer Befehlsfolgen auf Kommandozeile gelingt meist nur dann fehlerfrei, wenn man die automatische Ergänzung der Kommandozeile zu Hilfe nimmt oder mit Skripten bzw. Variablen arbeitet.

### 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 wieder entfernen muss), funktioniert die automatische Ergänzung durch mehrfaches Drücken der <TAB>-Taste.

### Setzen von Variablen

```
set pfad="c:\Program Files (x86)"\Win...  
echo %pfad%
```

Komplexe Pfade oder Eingaben können mit Variablen einfacher gehalten werden.

## 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 bzw. 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 automatisiert werden, ist es erforderlich, das Startskript von WinPE (**\windows\system32\startnet.cmd**) anzupassen.

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

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

In der Datei startnet.cmd findet man den Startbefehl wpeinit. Diese Datei startnet.cmd soll nun um den Aufruf von startjob.bat ergänzt werden.



Eine kleine Herausforderung dabei ist, dass der Laufwerksbuchstabe des USB-Sticks und damit der Ort von startjob.bat nicht in jedem Fall vorhergesagt werden kann. Die nachfolgenden Befehle prüfen deshalb zuerst, in welchem Laufwerk die Datei startjob.bat liegt, bevor diese aufgerufen wird.

Ergänzung der Datei:

```
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 Notepad beendet und das Image geschlossen.

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

Das Skript startjob.bat sollte nun bei jedem Start von WinPE automatisch ausgeführt werden. Dieses Skript lässt sich später modifizieren, ohne dass das Boot-Image erneut verändert werden muss.





## ZUSAMMENFASSUNG: ERSTELLEN EINES WINPE-STICKS

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

Die nachfolgenden Befehle können ggf. kopiert werden, wenn das Dokument mit einem geeigneten PDF-Reader (z. B. Internet-Explorer oder Microsoft Edge) geöffnet wird.

### GRUNDSTRUKTUR

```
copy c:\winpe64 c:\winpe64
```

### SPRACHDATEIEN LÖSCHEN

Gegebenenfalls können überflüssige Sprachdateien in c:\winpe64\media gelöscht. Dies ist allerdings eine rein optische Sache, um im Basisverzeichnis des USB-Sticks nicht die Übersicht zu verlieren. Auf Kommandozeile geschieht dies mit dem Befehl

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

### DEUTSCHE TASTATUR

Die nachfolgenden Befehle können auf Kommandozeile ausgeführt werden oder in ein Skript geschrieben werden. Sie sollten zunächst einzeln getestet werden.

```
@echo off
echo Installation der deutschen Tastatur
```

```
REM die nachfolgenden Pfade muessen ggf. angepasst werden
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

Die nachfolgenden Befehle sollten auf Kommandozeile (mit administrativen Rechten) eingegeben werden. Die Pfade müssen gegebenenfalls angepasst werden.

```
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

Die Datei startjob.bat (c:\winpe64\media\startjob.bat) auf dem USB-Stick stellt nur eine Netzverbindung her und ruft die eigentliche startjob.bat im Netzwerk auf.

```
@echo off
color F0
echo.
echo.
echo Verbindung ins Netz wird hergestellt:
echo.
net use n: \\10.36.104.24\Vorlagen /user: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          Die Disk-Nr. muss angepasst werden.
DISKPART> clean
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**

makewinpemedia /ufd c:\winpe64 u: (Laufwerksbuchstabe des USB-Sticks)

