VHDX BUILDER / WINPE BUILDER

INSTRUKCJA CAŁEGO PROJEKTU

Physical Disk Clone v1.8.1 • stan architektury 2026-10-04

Dokument uaktualniony o klonowanie RAW, TailCache/HARD LIMIT, retry bloków, weryfikację, restore IMG.GZ oraz zarządzanie ONLINE/OFFLINE.

Zakres aktualizacji: zachowano strukturę i rozwiązania z dotychczasowej instrukcji, a opis projektu rozszerzono o aktualny moduł Physical Disk Clone v1.8.1 oraz praktyczne rozwiązania wypracowane podczas testów dysków ADT.

1. CEL PROJEKTU

VHDX Builder jest aplikacją C# WinForms (.NET 8) uruchamianą w Windows jako administrator. Projekt składa się z trzech głównych modułów:

A) WinToGo / Native-boot VHDX

- źródło: instalacyjne ISO Windows
- źródło: IMG / IMG.GZ / DD / RAW
- wynik: Windows w VHDX
- opcjonalne przygotowanie pod Ventoy
- osobny tryb kopii dysku 1:1

B) WinPE Builder

- źródło systemowe: Windows ADK + Windows PE Add-on
- wspólne przygotowanie boot.wim
- język, komponenty, sterowniki, Tools, skrypty, REG
- osobne formaty wynikowe:

1. ISO
2. Ventoy VHDX
3. Hyper-V Gen2 VHDX
4. VirtualBox VDI / VMDK

C) Physical Disk Clone

- PhysicalDrive -> IMG.GZ (master / archiwum)
- PhysicalDrive -> PhysicalDrive (Disk-to-Disk)
- IMG.GZ -> PhysicalDrive (restore bez Rufusa)
- pełna / szybka / pominięta weryfikacja
- narzędzie ONLINE / OFFLINE dla dysków fizycznych
- specjalny tryb ADT/Rufus z TailCache i HARD LIMIT

WAŻNE:

Nie próbujemy już tworzyć jednego „uniwersalnego VHDX” do wszystkiego.

Każdy format ma własną metodę przygotowania.

Docelowy podział:

```
ISO
-> Rufus / Ventoy
Ventoy VHDX
-> Ventoy VHD Boot
Hyper-V VHDX
-> Hyper-V Generation 2
VDI / VMDK
-> VirtualBox
```

2. WYMAGANIA

Podstawowe:

- Windows 10/11 x64
- konto z prawami administratora
- Visual Studio 2022
- .NET 8 SDK z obsługą Windows Desktop / WinForms
- wystarczająca ilość wolnego miejsca na dysku systemowym i docelowym

Do WinPE:

- Windows ADK
- Windows PE Add-on for the Windows ADK
- w ADK wymagane są Deployment Tools

Do Ventoy:

- zainstalowany Ventoy na nośniku
- dla bootowania VHDX: ventoy_vhdboot.img
- opcjonalnie VentoyVlnk.exe dla układu dwóch dysków

Do VirtualBox:

- Oracle VirtualBox
- VBoxManage.exe

Do testów Hyper-V:

- Hyper-V
- maszyna Generation 2

UWAGA:

Program należy uruchamiać jako administrator, ponieważ korzysta m.in. z:

- DISM
- DiskPart
- montowania ISO/WIM/VHDX
- zapisu BCD/EFI
- pracy z offline registry

3. STRUKTURA PROJEKTU

Przykładowa struktura:

```
VHDXBuilder/
    |
    |-- VHDXBuilder.csproj
    |-- Program.cs
    |-- MainForm.cs
    |-- MainForm.Designer.cs
    |-- app.manifest
    |
    |-- Models/
    |   |-- SourceImageType.cs
    |   |-- SourceAnalysisResult.cs
    |   |-- PartitionInfo.cs
    |   |-- DiskLayoutInfo.cs
    |   `-- WindowsImageInfo.cs
    |
    |-- Services/
    |   |-- SourceAnalyzerService.cs
    |   |-- DiskLayoutParser.cs
    |   |-- ProcessRunner.cs
    |   |-- IsoService.cs
    |   |-- DismService.cs
    |   |-- RawImageReader.cs
    |   |-- RawToVhdxService.cs
    |   |-- IsoToVhdxService.cs
    |   |-- VentoyService.cs
    |   `-- NativeBootVhdxService.cs
    |
    `-- WinPeBuilder/
    |-- WinPeBuilderForm.cs
    |-- WinPeBuilderForm.Designer.cs
    |-- WinPeBuilderService.cs
    |
    |-- Common/
    |   |-- WinPeModels.cs
    |   |-- WinPeAdkService.cs
    |   |-- WinPeWorkspaceService.cs
    |   |-- WinPeProcessHelper.cs
    |   |-- WinPeUefiDiskHelper.cs
    |   |-- WinPeBuilderForm.Adk.cs
    |   `-- WinPeBuilderForm.Customization.cs
    |
    |-- ISO/
    |   |-- WinPeIsoService.cs
    |   `-- WinPeBuilderForm.ISO.cs
    |
    |-- VentoyVHDX/
    |   |-- WinPeVentoyVhdxService.cs
    |   `-- WinPeBuilderForm.VentoyVHDX.cs
    |
    |-- HyperVVHDX/
    |   |-- WinPeHyperVVhdxService.cs
```

```
|   `-- WinPeBuilderForm.HyperVVHDX.cs
|
`-- VirtualBox/
|-- WinPeVirtualBoxService.cs
`-- WinPeBuilderForm.VirtualBox.cs
|-- PhysicalDiskClone/
|   |-- PhysicalDiskCloneForm.cs
|   |-- PhysicalDiskCloneForm.Designer.cs
|   `-- PhysicalDiskCloneService.cs
```

4. ZASADA PRACY GŁÓWNEGO OKNA

Główne okno jest modułem WinToGo / Native-boot.

Typowy przebieg:

1. Obraz źródłowy
2. Analiza obrazu
3. **Dla ISO Windows:**
   - WinToGo / Native-boot VHDX
   - WinPE Builder
4. Windows / dodatkowe pliki
5. Sterowniki offline
6. Pierwsze uruchomienie / ChildCompletion
7. Docelowy VHDX
8. Ventoy
9. Physical Disk Clone — osobny moduł do klonowania i odtwarzania dysków

WinPE jest osobnym modułem i nie należy mieszać jego pól wynikowych z polem docelowego VHDX WinToGo.

5. ANALIZA ŹRÓDŁA

Obsługiwane typy:

- ISO
- IMG
- IMG.GZ
- DD
- RAW

Program ustala m.in.:

- typ pliku
- wielkość pliku
- wielkość obrazu dysku
- GPT / MBR
- partycje
- boot mode
- obecność instalacyjnego install.wim / install.esd

Dla instalacyjnego ISO Windows:

- ISO jest montowane w Windows
- sprawdzany jest katalog sources
- wyszukiwany jest:

  `sources\install.wim`

albo

  `sources\install.esd`

- pobierana jest lista edycji Windows

6. WINTOGO Z INSTALACYJNEGO ISO WINDOWS

To jest sprawdzona ścieżka dla Ventoy.

KLUCZOWA ZASADA:

Docelowy VHDX ma jedną partycję PRIMARY NTFS.

NIE tworzymy w tym VHDX:

- EFI
- MSR
- lokalnego BCDBoot

Układ:

```
VHDX
`-- PRIMARY NTFS
`-- Windows
```

Ventoy zapewnia mechanizm VHD Boot z zewnątrz.

6.1. Montowanie ISO

Program montuje ISO i uzyskuje literę napędu, np.:

```
D:\
```

Wtedy obraz instalacyjny znajduje się np.:

```
D:\sources\install.wim
```

lub:

```
D:\sources\install.esd
```

6.2. Odczyt edycji Windows

Technicznie DISM wykonuje operację typu:

```
dism.exe /Get-WimInfo /WimFile:"D:\sources\install.wim"
```

albo:

```
dism.exe /Get-WimInfo /WimFile:"D:\sources\install.esd"
```

Użytkownik wybiera odpowiedni INDEX.

6.3. Tworzenie VHDX

Schemat DiskPart dla dynamicznego VHDX:

```
create vdisk file="C:\TEST\Windows.vhdx" maximum=65536 type=expandable
select vdisk file="C:\TEST\Windows.vhdx"
attach vdisk
create partition primary
format fs=ntfs quick label="Windows"
assign letter=W
exit
```

Dla VHDX stałego:

```
type=fixed
```

Rozmiar w DiskPart podawany jest w MB.

6.4. Zastosowanie install.wim / install.esd

Przykład:

```
dism.exe /Apply-Image ^
/ImageFile:"D:\sources\install.wim" ^
/Index:1 ^
/ApplyDir:W:\
```

Dla ESD analogicznie:

```
dism.exe /Apply-Image ^
/ImageFile:"D:\sources\install.esd" ^
/Index:1 ^
/ApplyDir:W:\
```

Po operacji program sprawdza m.in.:

```
W:\Windows\System32\Config\SYSTEM
```

oraz:

```
W:\Windows\System32\winload.efi
```

6.5. Sterowniki offline Windows

Jeżeli wskazany został folder sterowników:

```
dism.exe /Image:W:\ ^
/Add-Driver ^
/Driver:"C:\Drivers" ^
/Recurse
```

Sterowniki muszą być rozpakowane do postaci INF/CAT/SYS.

6.6. Tools

Dodatkowe narzędzia mogą zostać skopiowane do:

```
W:\Tools
```

Program może kopiować:

- pojedynczy plik
- cały katalog

6.7. ChildCompletion

Opcjonalna poprawka pierwszego uruchomienia:

```
HKLM\SYSTEM\Setup\Status\ChildCompletion
setup.exe = 3
```

Przy edycji offline można załadować hive:

```
reg load HKLM\VHDX_SYSTEM W:\Windows\System32\Config\SYSTEM
```

ustawić:

```
reg add "HKLM\VHDX_SYSTEM\Setup\Status\ChildCompletion" ^
/v setup.exe ^
/t REG_DWORD ^
/d 3 ^
/f
```

następnie:

```
reg unload HKLM\VHDX_SYSTEM
```

6.8. Odłączanie VHDX

Przykład DiskPart:

```
select vdisk file="C:\TEST\Windows.vhdx"
detach vdisk
exit
```

Po detach program czeka, aż system zwolni uchwyt do pliku VHDX.

Jest to ważne przed:

- ponownym użyciem VHDX
- kopiowaniem
- etapem Ventoy
- VentoyVlnk

7. IMG / IMG.GZ / DD / RAW -> NATIVE-BOOT VHDX

Ta ścieżka jest inna niż prosta kopia 1:1.

Schemat:

```
IMG / IMG.GZ / DD / RAW
|
v
tymczasowy obraz/VHDX 1:1
|
v
podłączenie źródła
|
v
znalezienie partycji Windows
|
v
DISM Capture-Image -> tymczasowy WIM
|
v
nowy docelowy VHDX
jedna PRIMARY NTFS
|
v
DISM Apply-Image
|
v
sterowniki / Tools / registry
|
v
detach
|
v
Ventoy
```

7.1. IMG.GZ

IMG.GZ jest najpierw dekompresowany strumieniowo.

Nie należy zakładać, że rozmiar pliku .gz jest wielkością dysku.

Przykładowo:

- plik skompresowany może mieć ~40 GB
- obraz dysku po rozpakowaniu może reprezentować ~120 GB

7.2. Tymczasowy obraz 1:1

RawToVhdxService zachowuje geometrię źródła i umożliwia podłączenie obrazu przez Windows.

Jeżeli źródłowy VHDX po podłączeniu jest OFFLINE, należy przełączyć dysk ONLINE.

7.3. Znalezienie partycji Windows

Program szuka partycji zawierającej:

```
Windows\System32\Config\SYSTEM
```

To jest bardziej niezawodne niż zakładanie numeru partycji.

7.4. Capture Windows do tymczasowego WIM

Techniczny wzorzec DISM:

```
dism.exe /Capture-Image ^
/ImageFile:"C:\TEMP\CapturedWindows.wim" ^
/CaptureDir:S:\ ^
/Name:"Captured Windows" ^
/Compress:max ^
/CheckIntegrity
```

gdzie S:\ jest znalezioną partycją Windows ze źródłowego obrazu.

7.5. Nowy docelowy VHDX

Tworzony jest nowy VHDX z jedną partycją NTFS:

```
create vdisk file="C:\TEST\Windows.vhdx" maximum=65536 type=expandable
select vdisk file="C:\TEST\Windows.vhdx"
attach vdisk
create partition primary
format fs=ntfs quick label="Windows"
assign letter=W
exit
```

7.6. Apply przechwyconego WIM

```
dism.exe /Apply-Image ^
/ImageFile:"C:\TEMP\CapturedWindows.wim" ^
/Index:1 ^
/ApplyDir:W:\
```

Dalej wykonywane są opcjonalnie:

- Add-Driver
- Tools
- ChildCompletion

Tak samo jak przy ISO Windows.

8. PHYSICAL DISK CLONE V1.8.1

Physical Disk Clone jest osobnym modułem do pracy na fizycznych dyskach RAW.

Moduł nie przebudowuje systemu plików ani partycji. Kopiuje strumień sektorowy i zachowuje Primary GPT/MBR, offsety partycji oraz dane rozruchowe.

8.1. Dostępne tryby

1. PhysicalDrive -> IMG.GZ

- tworzy skompresowany master RAW,
- SHA-256 jest liczony podczas kopiowania,
- dla ADT można użyć zgodnego EOF przed końcem ostatniej partycji.

2. PhysicalDrive -> PhysicalDrive

- bezpośredni klon dysk-do-dysku,

- bez pliku pośredniego i bez Rufusa,
- target jest ustawiany OFFLINE przed zapisem,
- po zakończeniu target pozostaje OFFLINE.

3. IMG.GZ -> PhysicalDrive

- bezpośrednie rozpakowanie GZIP na fizyczny dysk,
- Rufus nie jest wymagany,
- przed zapisem sprawdzany jest Primary GPT i rozmiar targetu,
- plik IMG.GZ nie może leżeć na tym samym PhysicalDrive, który ma być nadpisany.

8.2. Tryby weryfikacji

Pełna SHA-256

- najbezpieczniejsza,
- po zapisie cały wynik jest ponownie odczytywany,
- najwolniejsza.

Szybka — początek / środek / koniec

- domyślnie próbki po 8 MiB,
- dla Disk-to-Disk i IMG.GZ -> Disk daje duży zysk czasu,
- dla tworzenia IMG.GZ przejście po skompresowanym strumieniu nadal może wymagać sekwencyjnego dojścia do końca GZIP.

Bez weryfikacji

- najszybsza,
- zalecana wyłącznie do testów roboczych.

8.3. Bezpieczeństwo

Moduł:

- blokuje użycie dysku systemowego/boot jako celu,
- nie pozwala wybrać tego samego dysku jako source i target,
- wymaga odpowiedniej pojemności targetu,
- ustawia target OFFLINE przed zapisem,
- utrzymuje source tylko do odczytu,
- próbuje ustabilizować source przez OFFLINE lub blokady woluminów,
- po błędzie/anulowaniu nie udaje sukcesu i nie pomija danych.

Po Disk-to-Disk lub IMG.GZ -> Disk target celowo pozostaje OFFLINE.

Dwa dyski z identycznymi GUID-ami GPT nie powinny być jednocześnie automatycznie montowane przez Windows.

8.4. Narzędzie ONLINE / OFFLINE

Przycisk ONLINE / OFFLINE otwiera listę PhysicalDrive z informacją:

- ONLINE / OFFLINE,
- READ-ONLY,
- SYSTEM / BOOT,
- GPT / MBR,
- model, numer, magistrala i pojemność.

Dysk systemowy/boot jest chroniony przed ustawieniem OFFLINE.

Po sklonowaniu target można ustawić ONLINE dopiero po upewnieniu się, że drugi dysk z identycznymi GUID-ami nie jest równocześnie aktywny.

9. VENTOY DLA WINTOGO / NATIVE-BOOT

Dostępne tryby:

1. Brak
2. Jeden dysk Ventoy
3. Dwa dyski / link

9.1. Jeden dysk

Gotowy VHDX jest kopiowany na dużą partycję Ventoy.

9.2. VHD Boot plugin

Plugin powinien znajdować się na partycji Ventoy:

```
\ventoy\ventoy_vhdboot.img
```

9.3. Dwa dyski

Jeżeli VHDX znajduje się na innym dysku, używany jest:

```
VentoyVlnk.exe
```

Wynikiem jest plik:

```
*.vlnk.vhdx
```

umieszczony na partycji Ventoy.

Program zachowuje ten tryb oddzielnie od zwykłego kopiowania.

10. WINPE BUILDER - OGÓLNA ZASADA

WinPE Builder nie korzysta z install.wim wybranego w głównym oknie.

Podstawa WinPE pochodzi z:

```
Windows ADK + Windows PE Add-on
```

Program przygotowuje JEDEN wspólny workspace i boot.wim.

Dopiero potem z tego samego przygotowanego WinPE tworzony jest jeden

z wybranych formatów:

```
ISO
Ventoy VHDX
Hyper-V VHDX
VirtualBox VDI / VMDK
```

Dzięki temu:

- język jest ten sam
- komponenty są te same
- sterowniki są te same
- Tools są te same
- startnet.cmd jest ten sam

11. WYKRYWANIE ADK

Program szuka m.in.:

```
C:\Program Files (x86)\Windows Kits\10\
Assessment and Deployment Kit\
Windows Preinstallation Environment
```

Wymagane pliki/katalogi:

- copype.cmd
- MakeWinPEMedia.cmd
- amd64\WinPE_OCs
- Deployment Tools\DandISetEnv.bat
- DISM z ADK, jeżeli jest dostępny

Polecenia ADK są uruchamiane przez wrapper CMD:

```
call "...\DandISetEnv.bat"
call "...\copype.cmd" ...
lub
call "...\MakeWinPEMedia.cmd" ...
```

12. PRZYGOTOWANIE WSPÓLNEGO WINPE WORKSPACE

Domyślny katalog roboczy:

```
C:\VHDXBuilder_Work\WinPE_<GUID>\
```

Jeżeli utworzenie go się nie powiedzie, używany jest TEMP użytkownika.

12.1. copype

Przykład:

```
copype.cmd amd64 "C:\VHDXBuilder_Work\WinPE_xxx\work"
```

Powstaje m.in.:

```
work\media\sources\boot.wim
work\mount
```

12.2. Mount boot.wim

```
dism.exe /Mount-Image ^
/ImageFile:"...\work\media\sources\boot.wim" ^
/Index:1 ^
/MountDir:"...\work\mount"
```

12.3. Pakiet językowy

Dla pl-PL:

```
dism.exe /Image:"...\mount" ^
/Add-Package ^
/PackagePath:"...\WinPE_OCs\pl-pl\lp.cab"
```

12.4. Optional Components

Obsługiwane komponenty:

- WinPE-WMI
- WinPE-NetFX
- WinPE-Scripting
- WinPE-PowerShell
- WinPE-DismCmdlets
- WinPE-StorageWMI
- WinPE-SecureStartup
- WinPE-HTA

Przykład:

```
dism.exe /Image:"...\mount" ^
/Add-Package ^
/PackagePath:"...\WinPE_OCs\WinPE-WMI.cab"
```

Następnie pakiet językowy komponentu, jeśli istnieje:

```
dism.exe /Image:"...\mount" ^
/Add-Package ^
/PackagePath:"...\WinPE_OCs\pl-pl\WinPE-WMI_pl-pl.cab"
```

Kolejność zależności jest kontrolowana przez program.

Przykład:

```
PowerShell automatycznie wymaga m.in.:
```

- WMI
- NetFX
- Scripting

12.5. Ustawienie języka

```
dism.exe /Image:"...\mount" /Set-AllIntl:pl-pl
```

12.6. Sterowniki WinPE

```
dism.exe /Image:"...\mount" ^
/Add-Driver ^
/Driver:"C:\Drivers\WinPE" ^
/Recurse
```

Przy braku touchpada/myszki mogą być potrzebne sterowniki:

- chipset
- USB/xHCI
- Intel/AMD Serial IO
- I2C
- GPIO
- HID
- sterownik touchpada

Sam sterownik touchpada może nie wystarczyć, jeżeli urządzenie pracuje przez I2C/Serial IO.

12.7. Tools w boot.wim

Jeżeli wybrano Tools wewnątrz boot.wim:

```
mount\Tools
```

Po uruchomieniu WinPE:

```
%SystemDrive%\Tools
```

Dla RAM-boot ISO %SystemDrive% jest zwykle X:.

Dla flat-boot dysku może być inną literą.

12.8. Tools poza boot.wim

Jeżeli wybrano Tools na nośniku:

```
work\media\Tools
```

Program tworzy znacznik:

```
.vhdxbuilder-tools
```

startnet.cmd może wyszukać nośnik po literach dysków.

12.9. startnet.cmd

Program tworzy / modyfikuje:

```
Windows\System32\startnet.cmd
```

Podstawowa zawartość:

```
@echo off
wpeinit
set TOOLS_ROOT=
```

Opcjonalna inicjalizacja sieci:

```
wpeutil InitializeNetwork
ipconfig /renew >nul 2>&1
```

Wyszukanie Tools:

```
if exist "%SystemDrive%\Tools\.vhdxbuilder-tools" set TOOLS_ROOT=%SystemDrive%\Tools
```

Jeżeli Tools są na nośniku:

```
for %%D in (C D E F G H I J K L M N O P Q R S T U V W Y Z) do ...
```

Opcjonalne dodanie do PATH:

```
if defined TOOLS_ROOT set PATH=%PATH%;%TOOLS_ROOT%
```

Import REG:

```
reg import "%SystemDrive%\Scripts\VHDXBuilder-Startup.reg" >nul 2>&1
PowerShell:
powershell.exe -NoLogo -NoProfile -ExecutionPolicy Bypass ^
-File "%SystemDrive%\Scripts\VHDXBuilder-Start.ps1"
```

12.10. Commit boot.wim

Zapis jest rozdzielony od odmontowania:

```
dism.exe /Commit-Image /MountDir:"...\mount"
```

Następnie:

```
dism.exe /Unmount-Image /MountDir:"...\mount" /Discard
```

Dlaczego /Discard po /Commit-Image?

Ponieważ zmiany zostały już zapisane przez /Commit-Image.

Etap /Discard służy tutaj do zwolnienia mount pointu.

W przypadku problemu program może:

- ponowić Unmount
- sprawdzić:
  `dism.exe /Get-MountedImageInfo`
- spróbować:
  `dism.exe /Remount-Image /MountDir:"...\mount"`

13. WINPE -> ISO

ISO jest najprostszym i najbardziej uniwersalnym wynikiem dla fizycznego

nośnika USB.

Przeznaczenie w projekcie:

- Rufus
- Ventoy

Polecenie:

```
MakeWinPEMedia.cmd /ISO /F ^
"C:\VHDXBuilder_Work\WinPE_xxx\work" ^
"C:\TEST\WinPE.iso"
```

ISO korzysta bezpośrednio z przygotowanego:

```
work\media\sources\boot.wim
```

WAŻNE:

Ścieżki ISO nie należy komplikować, jeżeli działa poprawnie.

14. WINPE -> VENTOY VHDX

Ventoy VHDX jest specjalnie przygotowany pod Ventoy VHD Boot.

Układ:

```
VHDX
`-- PRIMARY NTFS
`-- Windows
```

NIE MA:

- EFI
- MSR
- BCDBoot

To jest celowe.

14.1. Tworzenie VHDX

Przykład DiskPart:

```
create vdisk file="C:\TEST\WinPE_Ventoy.vhdx" maximum=4096 type=expandable
select vdisk file="C:\TEST\WinPE_Ventoy.vhdx"
attach vdisk
create partition primary
assign letter=Z
format fs=ntfs quick label="WinPE-Ventoy"
exit
```

14.2. Apply boot.wim jako system

```
dism.exe /Apply-Image ^
/ImageFile:"...\work\media\sources\boot.wim" ^
/Index:1 ^
/ApplyDir:Z:\
```

Po operacji wymagany jest m.in.:

```
Z:\Windows\System32\winload.efi
```

Jeżeli Tools były skonfigurowane jako MediaRoot, są kopiowane także do docelowego \Tools.

14.3. Finalizacja

```
select vdisk file="C:\TEST\WinPE_Ventoy.vhdx"
detach vdisk
exit
```

Program czeka na zwolnienie pliku.

Gotowy VHDX jest przeznaczony do:

```
Ventoy VHD Boot
```

15. WINPE -> HYPER-V GEN2 VHDX

Ten VHDX jest SAMODZIELNIE BOOTOWALNY w UEFI.

Przeznaczenie:

```
Hyper-V Generation 2
```

Układ GPT:

```
VHDX
|-- EFI 260 MB FAT32
|-- MSR 16 MB
`-- Windows NTFS
```

15.1. DiskPart

Przykład:

```
create vdisk file="C:\TEST\WinPE_HyperV.vhdx" maximum=4096 type=expandable
select vdisk file="C:\TEST\WinPE_HyperV.vhdx"
attach vdisk
convert gpt
create partition efi size=260
format fs=fat32 quick label="SYSTEM"
assign letter=S
create partition msr size=16
```

```
create partition primary
format fs=ntfs quick label="WinPE-HyperV"
assign letter=W
exit
```

15.2. Apply boot.wim

```
dism.exe /Apply-Image ^
/ImageFile:"...\work\media\sources\boot.wim" ^
/Index:1 ^
/ApplyDir:W:\
```

15.3. BCDBoot UEFI

```
bcdboot.exe W:\Windows /s S: /f UEFI
```

Po operacji sprawdzany jest:

```
S:\EFI\Microsoft\Boot\BCD
```

15.4. Hyper-V

Tworząc VM:

- Generation 2
- podłączyć gotowy VHDX jako dysk
- firmware UEFI
- w razie problemów z własnym WinPE można odpowiednio ustawić Secure Boot

16. WINPE -> VIRTUALBOX VDI / VMDK

VirtualBox nie korzysta bezpośrednio z naszego Ventoy VHDX.

Program przygotowuje tymczasowy, samodzielnie bootowalny dysk UEFI:

```
VirtualBox-Staging.vhd
```

Układ tego VHD:

- GPT
- EFI 260 MB FAT32
- MSR 16 MB
- NTFS Windows
- DISM Apply-Image
- BCDBoot /f UEFI

Czyli baza jest taka sama jak dla Hyper-V, ale formatem pośrednim jest VHD.

16.1. Konwersja przez VBoxManage

Do VDI:

```
VBoxManage.exe clonemedium ^
"VirtualBox-Staging.vhd" ^
"C:\TEST\WinPE.vdi" ^
disk ^
--format=VDI ^
--variant=Standard
```

Dla dysku stałego:

```
--variant=Fixed
```

Do VMDK:

```
VBoxManage.exe clonemedium ^
"VirtualBox-Staging.vhd" ^
"C:\TEST\WinPE.vmdk" ^
disk ^
--format=VMDK ^
--variant=Standard
```

Program ma także fallback do starszego polecenia:

```
VBoxManage.exe clonehd ...
```

Po konwersji staging VHD jest usuwany.

16.2. Ustawienia VirtualBox

W maszynie VirtualBox należy:

- podłączyć utworzony VDI lub VMDK
- włączyć EFI
- bootować z dysku

17. RÓŻNICE POMIĘDZY FORMATAMI WINPE

ISO:

- boot.wim pozostaje w sources
- MakeWinPEMedia /ISO
- Rufus / Ventoy

Ventoy VHDX:

- jedna partycja NTFS
- boot.wim rozpakowany przez DISM Apply-Image
- bez EFI i bez BCDBoot
- start zapewnia Ventoy VHD Boot

Hyper-V VHDX:

- GPT
- EFI FAT32
- MSR
- NTFS
- Apply-Image
- BCDBoot UEFI
- samodzielny boot Hyper-V Gen2

VirtualBox VDI/VMDK:

- najpierw samodzielny UEFI VHD
- Apply-Image
- BCDBoot
- potem konwersja VBoxManage

18. DISM - ZESTAW NAJWAŻNIEJSZYCH POLECEŃ

Lista edycji WIM:

```
dism /Get-WimInfo /WimFile:"install.wim"
```

Lista edycji ESD:

```
dism /Get-WimInfo /WimFile:"install.esd"
```

Mount WIM:

```
dism /Mount-Image /ImageFile:"boot.wim" /Index:1 /MountDir:"C:\Mount"
```

Add Package:

```
dism /Image:"C:\Mount" /Add-Package /PackagePath:"pakiet.cab"
```

Add Driver:

```
dism /Image:"C:\Mount" /Add-Driver /Driver:"C:\Drivers" /Recurse
```

Set language:

```
dism /Image:"C:\Mount" /Set-AllIntl:pl-pl
```

Commit:

```
dism /Commit-Image /MountDir:"C:\Mount"
```

Unmount po zapisanym Commit:

```
dism /Unmount-Image /MountDir:"C:\Mount" /Discard
```

Discard bez zapisu:

```
dism /Unmount-Image /MountDir:"C:\Mount" /Discard
```

Stan mountów:

```
dism /Get-MountedImageInfo
```

Remount:

```
dism /Remount-Image /MountDir:"C:\Mount"
```

Apply:

```
dism /Apply-Image /ImageFile:"image.wim" /Index:1 /ApplyDir:W:\
```

Capture:

```
dism /Capture-Image ^
/ImageFile:"Captured.wim" ^
/CaptureDir:W:\ ^
/Name:"Windows" ^
/Compress:max ^
/CheckIntegrity
```

19. BCDBOOT - GDZIE UŻYWAMY, A GDZIE NIE

```
BCDBoot UŻYWAMY:
```

- Hyper-V Gen2 VHDX
- staging VHD dla VirtualBox

Przykład:

```
bcdboot W:\Windows /s S: /f UEFI
BCDBoot NIE UŻYWAMY:
```

- WinToGo / Native-boot VHDX pod Ventoy
- WinPE Ventoy VHDX

To jest bardzo ważne.

Nie wolno ponownie mieszać:

- samodzielnego dysku UEFI

z:

- dyskiem VHDX przeznaczonym dla Ventoy VHD Boot

20. OBSŁUGA BLOKAD PLIKÓW VHDX

Windows może jeszcze trzymać uchwyt do VHDX po komendzie detach.

Program powinien:

1. wykonać detach
2. próbować otworzyć VHDX z FileShare.None
3. czekać do około 30 sekund

4. dopiero potem:

- kopiować plik
- usuwać plik
- przekazywać do Ventoy
- wykonywać VentoyVlnk

To rozwiązanie jest szczególnie ważne przy dużych VHDX.

21. PROBLEMY DISM PRZY UNMOUNT

Przykładowy błąd:

```
Error: 5
Access is denied.
```

Projekt rozdziela dlatego:

```
/Commit-Image
```

od:

```
/Unmount-Image /Discard
```

Po poprawnym Commit obraz jest już zapisany.

Przy problemach:

- ponowienie unmount
- GC / oczekiwanie
- Get-MountedImageInfo
- Remount-Image
- ponowny unmount

Roboczy WinPE jest trzymany poza typowym katalogiem profilu użytkownika:

```
C:\VHDXBuilder_Work\...
```

22. STEROWNIKI WINPE

Jeżeli ISO WinPE uruchamia się, ale nie działa:

- touchpad
- mysz USB
- klawiatura
- karta sieciowa
- dysk NVMe/VMD/RAID

należy przygotować folder z rozpakowanymi sterownikami INF.

Przykład:

```
C:\WinPE-Drivers\
Chipset\
SerialIO\
I2C\
HID\
USB\
LAN\
Storage\
VMD\
RAID\
```

i wskazać cały folder w WinPE Builderze.

Program użyje:

```
DISM /Add-Driver /Recurse
```

23. SIEĆ WINPE

Podstawą jest LAN.

startnet.cmd:

```
wpeinit
wpeutil InitializeNetwork
ipconfig /renew
```

Wi-Fi nie jest traktowane jako podstawowa, gwarantowana funkcja WinPE.

Może być rozwijane osobno jako funkcja eksperymentalna.

24. TOOLS

Dwa sposoby:

A) InsideBootWim

- narzędzia są częścią systemu
- po boot: %SystemDrive%\Tools
- najlepszy wariant dla przewidywalności

B) MediaRoot

- narzędzia są poza boot.wim
- \Tools na nośniku
- startnet szuka litery nośnika

Marker:

```
.vhdxbuilder-tools
```

Opcja:

```
Dodaj Tools do PATH
```

ustawia:

```
PATH=%PATH%;%TOOLS_ROOT%
```

25. SKRYPTY STARTOWE WINPE

Obsługiwane:

- PowerShell .ps1
- REG
- dodatkowe linie startnet.cmd

```
PowerShell jest kopiowany jako:
\Scripts\VHDXBuilder-Start.ps1
REG jako:
\Scripts\VHDXBuilder-Startup.reg
```

Program używa %SystemDrive%, a nie sztywnego X:.

Jest to ważne, bo:

- ISO RAM-boot zwykle pracuje jako X:
- flat-boot może użyć innej litery

26. BUDOWANIE PROJEKTU W VISUAL STUDIO

1. Otwórz VHDXBuilder.sln albo VHDXBuilder.csproj.

2. Sprawdź .NET 8.

3. Sprawdź, czy wszystkie pliki WinPeBuilder są widoczne.

4. Nie zostawiaj duplikatów:

```
MainForm(1).cs
WinPeBuilderForm(1).cs
WinPeBuilderService(1).cs
itd.
```

5. Clean Solution.

6. Rebuild Solution.

7. Uruchom Visual Studio jako administrator podczas testów operacji dyskowych.

8. Uruchom aplikację jako administrator.

27. KOLEJNOŚĆ TESTÓW

Zalecana kolejność po większych zmianach:

A. WinPE

1. ISO

2. Ventoy VHDX

3. Hyper-V VHDX

4. VirtualBox VDI

5. VirtualBox VMDK

B. WinToGo

1. ISO -> VHDX

2. IMG.GZ -> Native-boot VHDX

3. Ventoy single disk

4. Ventoy .vlnk.vhdx

5. kopia 1:1

C. Physical Disk Clone

1. PhysicalDrive -> IMG.GZ + pełna SHA-256

2. IMG.GZ -> PhysicalDrive + szybka weryfikacja

3. PhysicalDrive -> PhysicalDrive + szybka weryfikacja

4. ONLINE / OFFLINE tool

Nie testować kilku nowych mechanizmów naraz bez pośredniego sprawdzenia.

28. TEST ISO WINPE

1. Utwórz ISO.

2. Sprawdź, czy plik istnieje.

3. Uruchom:

- Ventoy

albo

- Rufus -> pendrive

4. Sprawdź:

- start WinPE
- język
- wpeinit
- LAN
- Tools
- skrypt PS1
- REG

29. TEST VENTOY VHDX WINPE

1. Utwórz Ventoy VHDX.

2. Sprawdź, czy kończy się bez BCDBoot.

3. Skopiuj VHDX na Ventoy.

4. Upewnij się, że jest:

```
\ventoy\ventoy_vhdboot.img
```

5. Uruchom VHDX z menu Ventoy.

30. TEST HYPER-V

1. Utwórz Hyper-V VHDX.
2. Utwórz VM Generation 2.
3. Podłącz VHDX.
4. Ustaw boot z dysku.
5. Uruchom.
6. Jeśli firmware blokuje własny obraz WinPE, sprawdź ustawienie Secure Boot.

31. TEST VIRTUALBOX

1. Utwórz VDI lub VMDK.
2. Utwórz nową maszynę.
3. Włącz EFI.
4. Podłącz utworzony dysk.
5. Uruchom.

Jeżeli VBoxManage nie jest znaleziony:

- sprawdzić instalację VirtualBox
- sprawdzić lokalizację VBoxManage.exe

32. WAŻNE ZASADY ARCHITEKTURY

1. Nie ruszać sprawdzonego mechanizmu WinToGo ISO -> VHDX bez potrzeby.
2. Nie dodawać EFI/MSR/BCDBoot do Ventoy VHDX.
3. Hyper-V i VirtualBox mają własny UEFI bootloader.
4. WinPE przygotowujemy raz, format wynikowy budujemy osobno.
5. ISO pozostaje prostą ścieżką dla Rufusa i Ventoy.
6. VDI/VMDK jest przeznaczony dla VirtualBox.
7. Hyper-V Gen2 dostaje własny VHDX GPT/UEFI.
8. Tools i skrypty nie mogą zakładać na sztywno litery X:.
9. Po detach zawsze czekać na zwolnienie VHDX.
10. Przy DISM Commit i Unmount nie traktować tych operacji jako jednego kroku.

33. PODSUMOWANIE PRZEPŁYWÓW

WINTOGO ISO:

```
ISO Windows
-> mount
-> install.wim/esd
-> DISM Apply-Image
-> PRIMARY NTFS VHDX
-> Drivers / Tools / ChildCompletion
-> detach
-> Ventoy
```

WINTOGO IMG.GZ:

```
IMG.GZ
-> decompress
-> tymczasowy dysk źródłowy
-> znalezienie Windows
-> DISM Capture-Image
-> nowy PRIMARY NTFS VHDX
-> DISM Apply-Image
-> Drivers / Tools / ChildCompletion
-> detach
-> Ventoy
```

WINPE COMMON:

```
ADK
-> copype amd64
```

```
    -> Mount boot.wim
    -> language
    -> Optional Components
    -> Drivers
    -> Tools
    -> Scripts / REG / startnet
    -> Commit-Image
    -> Unmount
    -> wspólny przygotowany WinPE
```

WINPE ISO:

```
    prepared WinPE
    -> MakeWinPEMedia /ISO
    -> WinPE.iso
    -> Rufus / Ventoy
```

WINPE VENTOY VHDX:

```
    prepared boot.wim
    -> PRIMARY NTFS VHDX
    -> DISM Apply-Image
    -> detach
    -> Ventoy VHD Boot
```

WINPE HYPER-V:

```
    prepared boot.wim
    -> GPT VHDX
    -> EFI + MSR + NTFS
    -> DISM Apply-Image
    -> BCDBoot UEFI
    -> Hyper-V Gen2
```

WINPE VIRTUALBOX:

```
    prepared boot.wim
    -> GPT staging VHD
    -> EFI + MSR + NTFS
    -> DISM Apply-Image
    -> BCDBoot UEFI
    -> VBoxManage
    -> VDI / VMDK
    -> VirtualBox EFI
```

34. UWAGA DOTYCZĄCA POLECEŃ W TEJ INSTRUKCJI

Polecenia DISM, DiskPart, BCDBoot, copype, MakeWinPEMedia i VBoxManage są zapisane jako techniczne wzorce z przykładowymi literami dysków i ścieżkami.

W programie:

- litery dysków są przydzielane dynamicznie,
- katalog roboczy ma GUID,
- ścieżki są cytowane,
- błędy procesów są przechwytywane,
- VHD/VHDX są odłączane w finally,
- pliki są sprawdzane po wykonaniu operacji.

Nie należy kopiować przykładowych liter S:, W:, Z: do kodu na stałe.

35. PHYSICAL DISK CLONE — CZYTANIE BLOKOWE I RETRY

Podstawowy bufor kopiowania RAW ma domyślnie 4 MiB.

Odczyt i zapis są wykonywane sekwencyjnie, aby ograniczyć liczbę operacji I/O.

Przy zwykłym odczycie:

```
    BufferSizeBytes = 4 * 1024 * 1024
```

Jeżeli PhysicalDrive zgłosi IOException, np. Win32=121 (ERROR_SEM_TIMEOUT):

1. numer próby jest liczony osobno dla aktualnego bloku,
2. uchwyt \\.\PhysicalDriveN jest zamykany,
3. wykonywane jest krótkie opóźnienie,
4. uchwyt jest otwierany ponownie,
5. seek wraca dokładnie do tego samego offsetu,

6. blok jest zmniejszany:

```
4 MiB -> 1 MiB -> 256 KiB -> 64 KiB
```

7. po poprawnym odczycie następny blok ma własny licznik prób.

Domyślne parametry:

```
ReadRetryCount = 4
ReadRetryDelayMilliseconds = 750
MinimumRetryReadBlockBytes = 64 * 1024
```

Ważna zasada:

program nie przesuwa offsetu po błędzie i nie „udaje", że uszkodzony fragment został odczytany. Dane nie są automatycznie zastępowane zerami.

36. ADT / RUFUS — EOF, TAILCACHE I HARD LIMIT

W trybie zgodnym z ASUS ADT wynikowy RAW kończy się:

```
LastPartitionEnd - LogicalSectorSize
```

Czyli dokładnie jeden sektor logiczny przed końcem ostatniej partycji GPT.

Nie są zapisywane:

- końcowa wolna przestrzeń za TinyLinux,
- Backup GPT znajdujący się na fizycznym końcu nośnika.

Problem ujawniony podczas testów:

mostek USB Realtek RTL9210B-CG potrafił zwracać Win32=121 dla ostatnich sektorów

```
\\.\PhysicalDriveN po ustawieniu źródła OFFLINE, mimo że te same dane były
```

czytelne przez uchwyt partycji.

Rozwiązanie v1.8.1:

1. przed ustawieniem source OFFLINE ustalana jest ostatnia partycja GPT,

2. końcowe 4 MiB są zawsze czytane przez:

```
\\.\HarddiskNPartitionM
```

3. TailCache otrzymuje własne SHA-256,

4. początek TailCache jest kontrolnie porównywany z PhysicalDrive,

5. PhysicalReadLimit = TailCacheStartOffset,

6. PhysicalDrive nie może zostać odczytany ani o jeden bajt za HARD LIMIT,

7. od HARD LIMIT do EOF dane są zapisywane wyłącznie z TailCache w RAM.

Warunek integralności:

```
TailCacheStartOffset + TailCacheBytes.Length == ImageRawBytes
```

Jeżeli warunek nie jest spełniony, klon nie startuje.

37. DLACZEGO NIE UŻYWAMY ZERO-FILL DLA KOŃCÓWKI ADT

Zero-fill jest użyteczny w narzędziach ratunkowych, gdy sektor naprawdę jest nieodczytywalny i akceptujemy utratę danych.

W naszym przypadku końcowe sektory TinyLinux były dostępne przez uchwyt partycji, więc zastąpienie ich zerami uszkodziłoby prawidłowe dane.

Zasada projektu:

- retry dla przejściowych błędów I/O,
- TailCache dla znanego problemu końcówki ADT,
- błąd końcowy zamiast cichego zerowania danych.

38. DISK-TO-DISK — PRZEPŁYW

Schemat:

```
PhysicalDrive source
-> odczyt blokowy RAW
-> SHA-256
-> zapis blokowy RAW
-> PhysicalDrive target
```

Dla ADT obowiązuje ten sam TailCache i HARD LIMIT co przy IMG.GZ.

Target musi mieć pojemność wystarczającą dla obrazu.

Po zakończeniu:

- target pozostaje OFFLINE,
- source wraca ONLINE, jeśli program sam przełączył go OFFLINE,
- nie należy ustawiać obu klonów ONLINE równocześnie przy identycznych GUID GPT.

39. IMG.GZ -> PHYSICALDRIVE — RESTORE

Przed pierwszym zapisem:

1. sprawdzane jest rozszerzenie .img.gz,
2. ustalane jest, na jakim PhysicalDrive znajduje się plik źródłowy,
3. target nie może być tym samym dyskiem,
4. sprawdzany jest Primary GPT w pierwszych sektorach RAW,
5. z GPT odczytywany jest logiczny rozmiar dysku,
6. target musi być odpowiednio duży,
7. target jest ustawiany OFFLINE i writable.

Podczas zapisu:

- GZipStream jest dekompresowany sekwencyjnie,
- RAW jest zapisywany blokami,
- jednocześnie liczony jest SHA-256,
- dla szybkiej weryfikacji zapamiętywane są próbki początek/środek/koniec.

Po EOF obrazu program czyści niewielkie obszary targetu:

- do 4 MiB bezpośrednio za EOF obrazu,
- końcowe 4 MiB fizycznego nośnika.

Cel: usunięcie pozostałości starego Backup GPT / podpisów poprzedniego układu bez zerowania całej niewykorzystanej przestrzeni dysku.

40. WERYFIKACJA — SZCZEGÓŁY

Pełna SHA-256:

- źródłowy RAW ma SHA-256 liczony podczas kopiowania,
- wynik jest ponownie czytany,
- obliczone skróty muszą być identyczne.

Szybka:

- początek,
- środek,
- koniec,
- domyślna wielkość próbki: 8 MiB.

Dla IMG.GZ -> Disk próbki są przechwytywane podczas rozpakowywania, a później ponownie odczytywane z targetu i porównywane przez SHA-256.

Bez weryfikacji:

- zapis kończy się po Flush(true),
- tryb tylko do sytuacji, gdy czas jest ważniejszy od dodatkowej kontroli.

Postęp:

- przy aktywnej weryfikacji kopiowanie zajmuje zwykle 0–90% paska,
- weryfikacja 90–100%.

Dlatego 89–90% pod koniec kopiowania nie oznacza brakujących danych.

41. ONLINE / OFFLINE — ZASADY

Źródło:

- może zostać ustawione OFFLINE, aby ograniczyć zmiany w trakcie klonowania,
- po zakończeniu jest przywracane ONLINE tylko wtedy, gdy program sam je wyłączył.

Target:

- przed zapisem OFFLINE,
- ReadOnly=false,
- po klonowaniu pozostaje OFFLINE.

Narzędzie stanu dysków umożliwia ręczne:

```
Ustaw ONLINE
```

```
Ustaw OFFLINE
```

Nigdy nie ustawiaj OFFLINE dysku systemowego/boot.

Przy identycznych GUID GPT najpierw odłącz drugi klon.

42. TYPOWE BŁĘDY I ROZWIĄZANIA

Win32=121 / ERROR_SEM_TIMEOUT

- zmniejszanie bloku,
- ponowne otwarcie PhysicalDrive,
- TailCache dla końcówki ADT,
- brak pomijania danych.

Win32=5 / Access denied

- uruchom aplikację jako administrator,
- sprawdź blokady woluminów,
- użyj OFFLINE dla source/target zgodnie z trybem.

Target za mały

- porównuj realną liczbę bajtów/sektorów, nie sam napis marketingowy „512 GB”.

Dwa identyczne klony

- po Disk-to-Disk target pozostaw OFFLINE,
- nie montuj obu równocześnie bez potrzeby.

43. PODPISYWANIE APLIKACJI I SHA1

Do podpisywania wymagany jest plik PFX z kluczem prywatnym.

Plik CER nie wystarcza jako /f dla SignTool.

```
PowerShell:
& "C:\Program Files (x86)\Windows Kits\10\bin\<wersja>\x64\signtool.exe" sign `
/f "certyfikat.pfx" /p "<hasło>" `
/fd SHA256 /tr "http://timestamp.sectigo.com" /td SHA256 `
"VHDXBuilder.exe"
```

Do projektu przygotowano pomocniczy skrypt:

```
Sign-And-SHA1.ps1
```

Skrypt:

- pozwala wskazać PFX,
- pyta o hasło,
- pozwala wskazać EXE/DLL,
- podpisuje plik,
- oblicza SHA1 podpisanego pliku,
- kopiuje SHA1 do schowka.

44. AKTUALNY PRZEPŁYW DLA MASTERÓW ADT

Master do archiwum:

```
działający PhysicalDrive
-> Physical Disk Clone
-> IMG.GZ
-> pełna SHA-256 dla finalnego mastera
```

Szybkie powielanie:

```
działający PhysicalDrive
-> Disk-to-Disk
-> szybka weryfikacja
-> target pozostaje OFFLINE
```

Odtwarzanie archiwum:

```
IMG.GZ
-> IMG.GZ -> PhysicalDrive
-> szybka lub pełna weryfikacja
-> target pozostaje OFFLINE
```

Rufus nie jest wymagany do codziennego klonowania, ale pozostaje przydatnym zewnętrznym narzędziem kontrolnym dla zgodności obrazu RAW.

KONIEC INSTRUKCJI

Powiązany plik techniczny: README_VHDXBuilder_PhysicalDiskClone.md