USB Creator Tool Anleitung

In diesem Tool habe ich einen – hoffentlich nachvollziehbaren – einfachen aber von MSFT „fully supported“ Ansatz gewählt.

Das Teil hat vor etlichen Monaten mit einem simplen PowerShell Skript angefangen.

Parallel dazu haben Michael Niehaus und Johan Arwidmark offensichtlich an etwas Ähnlichem gearbeitet, was mich natürlich angeregt & bestätigt hat 😊

Blog von Johan:

A Blast From The Past - Jurassic Deployment - Deployment Research

Blog von Michael:

Booting your own Windows PE image – Out of Office Hours (oofhours.com)

Voraussetzungen

Das Tool funktioniert nur unter folgenden Voraussetzungen:

1. Man hat FULLY Local Admin Rechte auf der Box, wo man das Tool nutzen möchte (PowerShell oder PowerShell ISE als ADMIN starten, Execution-Policy muss mindestens „RemoteSigned“ sein)
2. Aktuelles ADK für Windows 10 ist installiert
3. Aktuelles WinPE Addon für ADK Windows 10 ist installiert
4. Aktuelles PowerShell (Host 5.x aufwärts) ist am Start
5. .Net Framework 4.7 aufwärts ist installiert (Bestandteil von PS 5.0)

ADK für Windows 10

http://go.microsoft.com/fwlink/p/?LinkId=526740&ocid=tia-235208000

WinPE Addon für ADK für Windows 10

https://go.microsoft.com/fwlink/?linkid=2120253

Das Tool checked das Vorhandensein der üblichen Installations-Verzeichnisse der o.g. Komponeten 😊

Es baut (u.a.) auch ein frisches neues WinPE X64 Root Verzeichnis (für die FAT32 Boot Partition des USB Sticks), wenn selbiges nicht vorhanden sein sollte; der Inhalt des „media“ Verzeichnisses wird (bis auf die WinPE Standard Boot.wim darin) auf die ersste (FAT32) Boot-Partition des Sticks kopiert.

WICHTIG:

Man habe nur genau EIN USB Medium (Stick) an der Box angeschlossen, wo man (als ADMIN!) das Tool started, von dem man sicher weiss, dass dasselbe vom Tool neu formatiert werden könnte !!

No Risk, No Fun 😊

Da aktuelle UEFI/GPT Maschinen nicht von NTFS booten können, wird – anderes als z.B. beim Rufus Tool – das USB Medium in zwei Partitionen aufgeteilt (eine kleine FAT32 für den Boot-Code und der Rest als grosse NTFS-Partition mit den Installationsquellen).

Das Tool checked auch die notwendigen Admin-Rechte & warnt entsprechend 😊

Auch der **Formatier-Vorgang** (des – wenn angeschlossen & gefundenen – USB Drives) wird - per Default - **nur mit „Confirm" Messages möglich**, ausser man stellt das „Confirm" in der zugehörigen USBCREATOR.xml um 😊

Um in der GUI Version des Tools eine Art Fortschrittsanzeige für die diversen Robocopy-Jobs innerhalb des Ablaufs zu haben, habe ich nach längerem Suchen die Funktion *Copy-ItemWithProgress* von Keith Garner gefunden und eingebaut:

https://keithga.wordpress.com/2014/06/23/copy-itemwithprogress

Die aktuellle Version kann auch das Boot.wim (auf derFAT32 Partition des Sticks!) um ggf. notwendige Treiber ergänzen, was bei aktueller Hardware (USB 3, NVMe SSDs usw.) oft notwendig ist, weil sonst ggf. weder Tastatur noch Netzwerk noch lokclae Festplatten „gefunden" werden 😊

Hierzu muss lediglich ein passendes Verzeichnis (mit den WinPE X64 Treibern) bereitgestellt werden.

Das Tool liest seine Konfiguration aus einer XML Datei, aus, die im gleichen Verzeichnis wie das Tool selbst liegen muss.

In der XML (*USBCreator.xml*) kann man auch die Minimalgrösse des USB-Mediums (in Byte!) festlegen (Default sind 28 GB = 30064771072 Bytes).

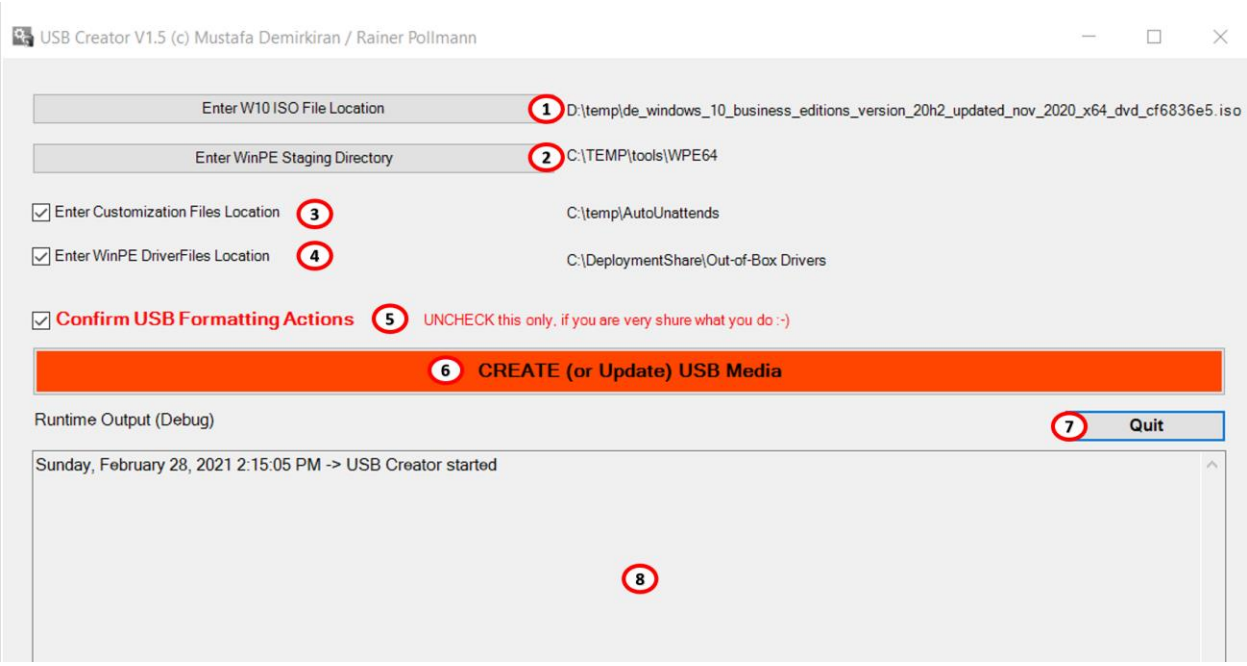
Schliesslich werden – bei Bedarf – diverse Dateien für das Customzing in den OEM-Ordner der NTFS Installationspartition kopiert sowie ein einfaches aber effektives Unattend Skript als „AutoUnattend.xml" in die Root der (FAT32) Boot-Partition kopiert, wo auch die setup.exe liegt.

Das ursprüngliche PowerShell Skript („nacktes" .PS1, keine GUI!) könnt Ihr natürlich alternativ zur GUI Version verwenden, darin sind ein paar „brauchbare" Sachen enthalten 😊

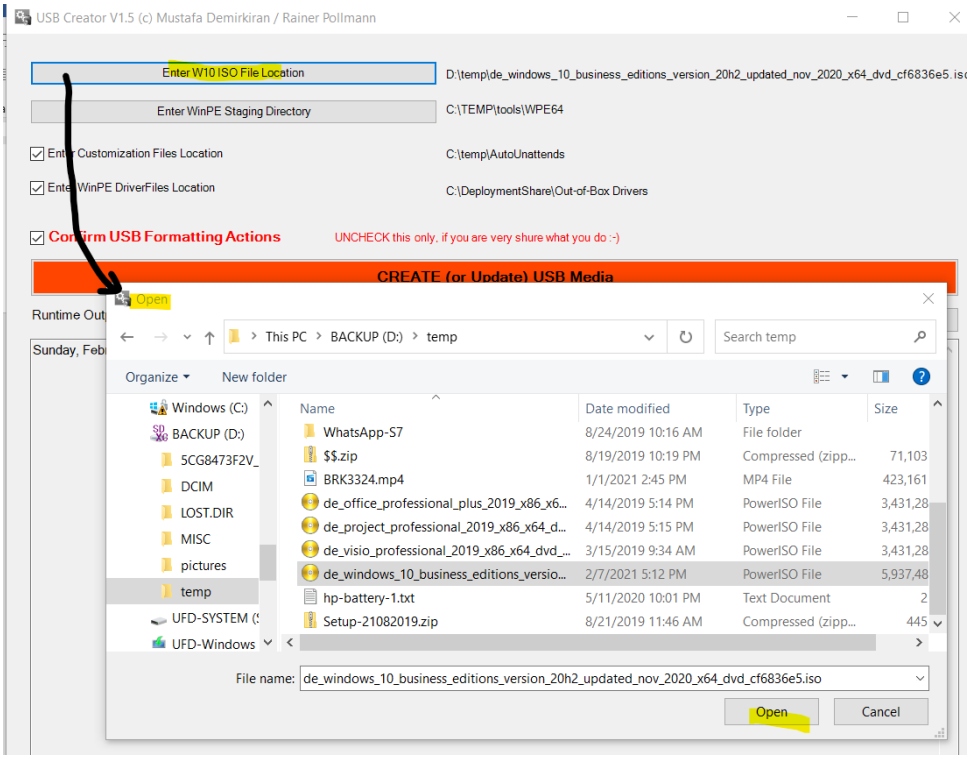
Selbiges benutzt die „gewöhnlichen" Fortschrittsanzeigen von PowerShell.

BugFixes & Wünsche werden auch gerne genommen 😊

GUI Elemente



1 Enter W10 ISO File Location



Startet einen Dateibrowser, gewählte Datei wird in das Textfeld rechts davon übernommen

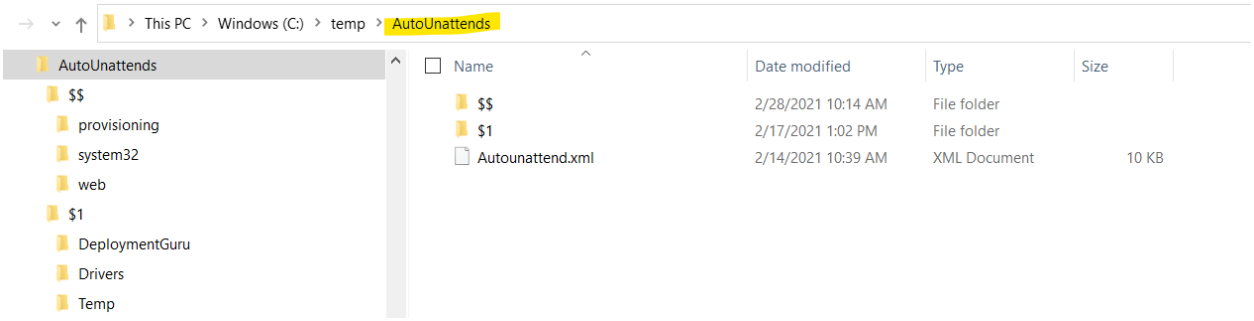
2 Enter WinPE Staging Directory

Startet einen Verzeichnis-Browser; in diesem Verzeichnis wird das AMD64 WinPE erwartet bzw. generiert, wenn noch nicht vorhanden.

Letzteres regelt das Skript *Create_WinPE64_Root.cmd*, welches dazu im gleichen Verzeichnis wie das Tool liegen muss.

3 Enter Customization File Location

Startet einen Verzeichnis-Browser, wenn gesetzt. Alle Dateien in diesem Verzeichnis werden in den Ordner OEM der NTFS Partittion (Installations-Quellen) kopiert.



Typisches Verzeichnis, das Autounattend.xml wird dabei NICHT kopiert, aber später in die Root der Boot-Partition kopiert (s.o.)

Zur Erinnerung:

Alle Dateien in $1 landen auf der fertigen Maschine auf der Root (C:)

Alle Dateien in $$ landen auf der fertigen Maschine im %windir%, also i.d.R. in c:\windows

Dieser Mechanismus funktioniert immer noch 😊

Das kann man dazu benutzen, Bildschirmhintergründe, Anpassungsskripte, JSON Files usw usw zu „transportieren“, sodaß selbige auch nach potentiellem SysPrep auf der lokalen Platte erhalten bleiben.

Wenn man den Haken aus der Checkbox herausnimmt, wird das Verzeichnis auf „none“ gesetzt & es werden KEINE Robocopy Aktionen dazu durchgeführt.

4 Enter WinPE DriverFiles Location

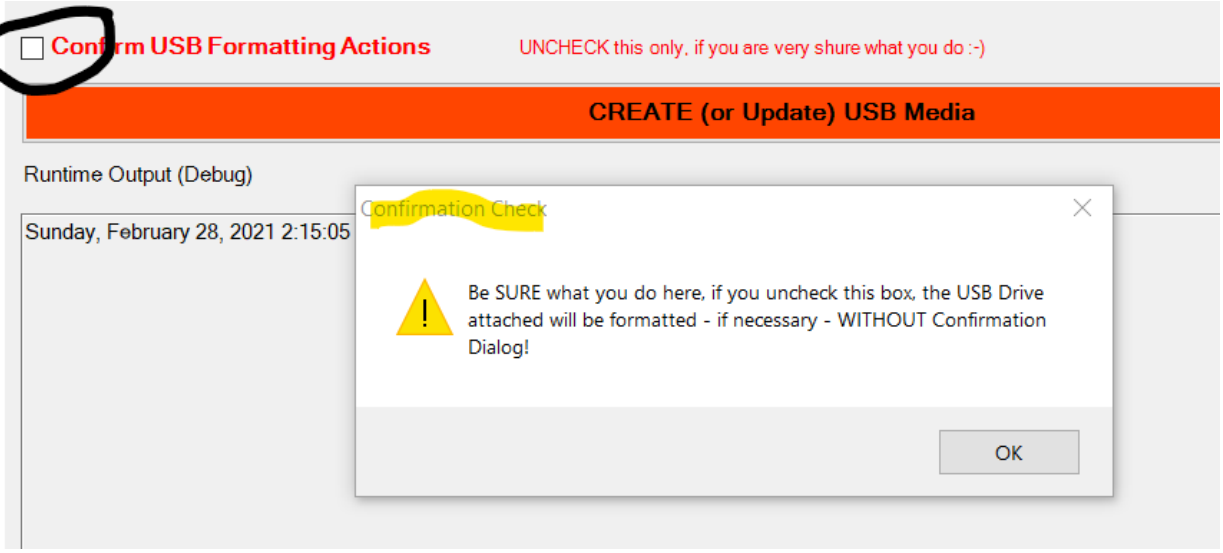
Startet einen Verzeichnis-Browser, wenn gesetzt.

Alle PnP Treiber darin werden (rekursiv!) auf Basis ihrer INF Dateien in das Boot-Image eingebaut.

Daher eignet sich am besten ein passender Out-of-Box Drivers Ordner, z.b. aus einem MDT Share wie oben gezeigt.

Wenn man den Haken aus der Checkbox herausnimmt, wird das Verzeichnis auf „none“ gesetzt & die boot.wim bleibt im Originalzustand (wie im Installations-ISO).

5 Confirm USB Formatting Actions



Gibt eine eindeutige Warnung aus 😊

Wenn die Checkbox nicht gesetzt ist, wird beim Formatieren des USB Mediums NICHT vorher nachgefragt!

6 CREATE (or Update) USB Media

Führt die “eigentlichen” Aktionen durch, u.a. die Partitionierung und Formatierung des USB Sticks (!)

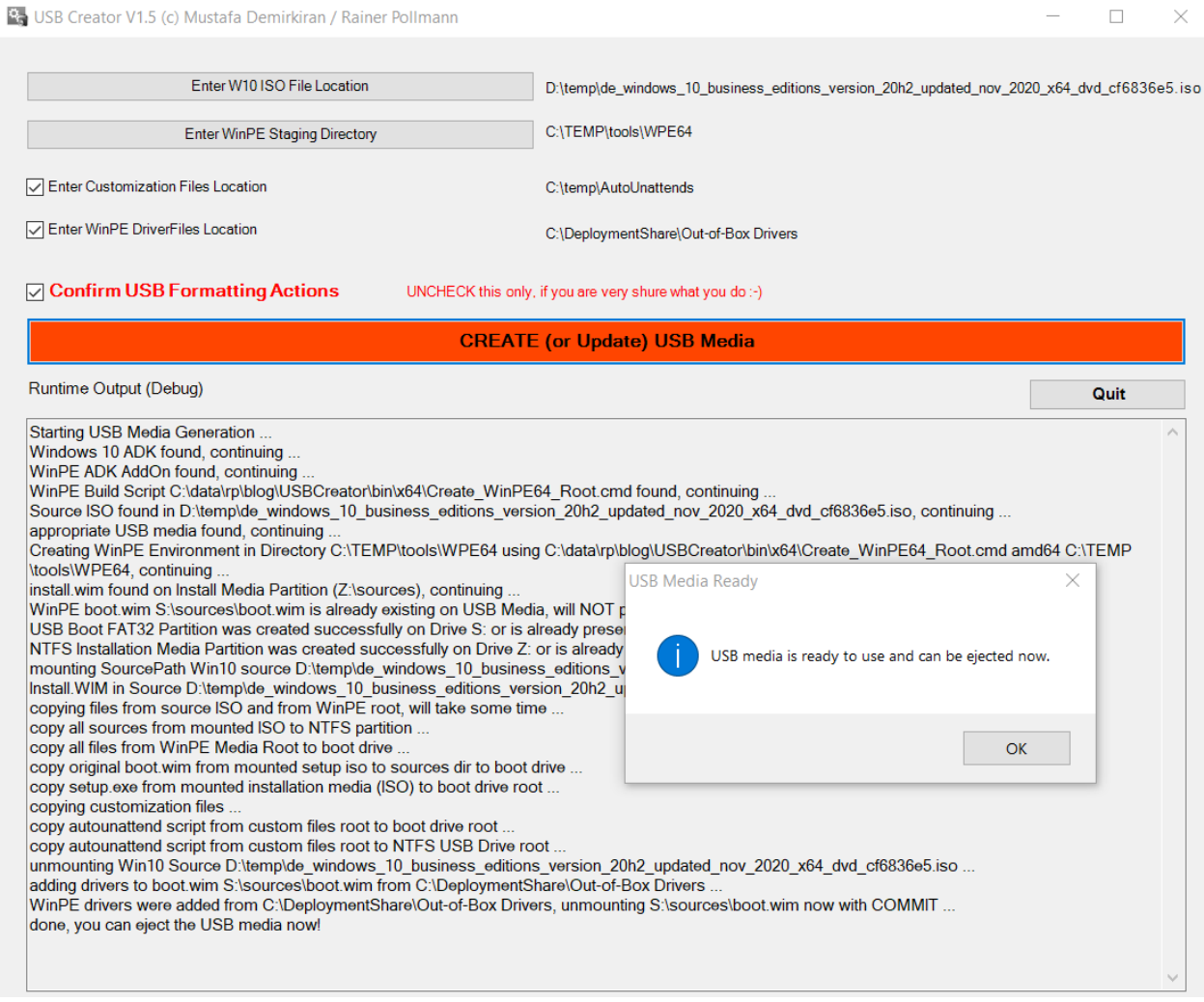
Falls selbiger aber schon die richtige Struktur aufweist (S: LW, Z: LW, u.a.m.), wird der Stick OHNE Formatierung geupdated (Bsp. neues ISO, neue Treiber …)

7 Quit

Beendet das Tool; hierbei wird auch die Config XML (*USBCreator.xml*) zurückgeschrieben, weil ja ggf. Änderungen an den o.g. Parametern vorgenommen wurden 😊

8 Runtime Output (Debug)

In diesem Bereich werden alle Aktionen des Tools ausgegeben.



Kompletter Durchlauf, nach Klick auf OK wird die Config XML zurückgeschrieben und das Tool beendet.