

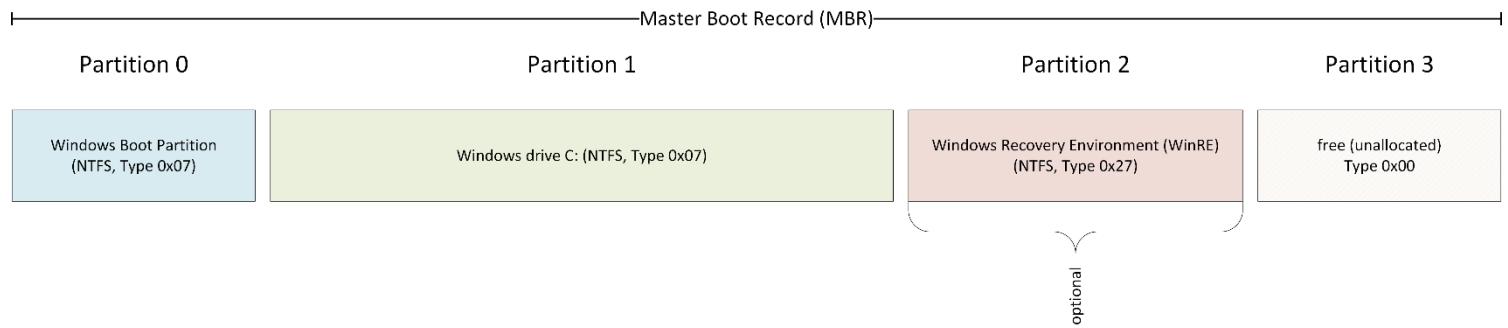
# part-y – Convert your MS Windows 10 system from MBR to GPT

## step-by-step guide

### Introduction

Most likely, Windows 11 will only boot in secure UEFI mode. There are still a lot of Windows 10 machines around, which boot in the legacy BIOS mode using a Master Boot Record. A limited number of these machines can be converted using Microsoft's tool `mbr2gpt.exe`.

`mbr2gpt.exe` is only able to convert systems not having any extended partitions on their boot disk. Only three (primary) partitions may be defined. The standard partitioning scheme looks like this:



part-y is able to convert all kinds of MBR-partitioning schemes, whether or not there are extended partitions, logical drives, etc.

It splits the Windows Boot partition into two partitions (EFI System Partition and Microsoft Reserved Partition), converts the MBR to the GPT (all partitions), re-creates the Boot Configuration Data (BCD), formats the new EFI System Partition (FAT32), copies all required Windows start files on it (also the newly created BCD) and establishes an EFI-NVRAM variable (load option) so that MS Windows can start in UEFI mode.

Two consecutive runs of the tool part-y are required: The first run is performed on your running Windows 10 system; the second run requires a Linux Live system (here: Debian 11 ‘Bullseye’). The Linux version of part-y performs the ‘hard work’ of converting everything.

It uses the Linux `efibootmgr` tool to establish the required load option (in the NVRAM of the EFI system).

### Prerequisites

Although part-y backups all disk sectors it writes to, it is still alpha software that contains bugs:

**DO BACKUP YOUR WINDOW SYSTEM DRIVE ENTIRELY BEFORE PROCEEDING!**

## **Step 1: Prepare a Linux Live (Debian 11 ‘Bullseye’) USB stick**

Please take a 16GB or 32GB USB stick and download the free software RUFUS (<https://rufus.ie/en/>) and the Debian 11 ‘Bullseye’ Live ISO image: <https://cdimage.debian.org/debian-cd/current-live/amd64/iso-hybrid/debian-live-11.0.0-amd64-mate.iso>

On your Windows system, insert the USB stick, and start RUFUS. Select the Debian 11 Live ISO Image and configure the partitioning scheme to ‘GPT’ and the target system to ‘UEFI (no CSM)’. Use the ISO write mode to write the image to the USB stick.

Either build the Linux and Windows versions of part-y from the source code or use the files in the Github project folder ‘prebuilt’. On Linux, building from source is easily performed by entering ‘make’. The binary is in the subfolder ‘build’. On Windows, use Microsoft Visual Studio 2019 Community Edition to open the solution file and build the x64 release version.

In both cases (everything from prebuilt folder or part-y build from source), you have to copy the following six (6) files to the root folder of the USB stick (it is FAT32 formatted):

`efibootmgr, efivar, libefiboot.so.1, libefivar.so.1, part-y, and part-y.exe`

The first ‘part-y’ is the ELF binary from Linux, the second one ‘part-y.exe’ the PE executable file for Windows.

## **Step 2: Backup your Windows system drive**

Please use an arbitrary tool to backup the entire drive where your Windows system partition (normally drive C:) is located. Again, part-y backups all disk sectors it writes to so that you can use part-y’s command ‘*restore*’ to restore these disk sectors if anything goes wrong. Nevertheless, part-y is alpha software, so expect bugs. Please also carefully read this disclaimer:

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## **Step 3: Prepare the conversion**

On your running Windows 10 system, open an Administrator command prompt (cmd.exe). The USB stick with the Debian 11 Live system has to be attached, too.

First of all, you have to identify the ordinal number of your Window system drive. You can execute part-y.exe to enumerate all disks (the USB stick is E: in this example):

```

Administrator: Eingabeaufforderung

Microsoft Windows [Version 10.0.19043.1237]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Windows\system32>e:\part-y enumdisks
INFO: number of physical disks is 2
device file      capacity    number of sectors  log.sec.size  phy.sec.size
-----
\\.\PhysicalDrive0  128.00 GB        268.435.456   512          4096
\\.\PhysicalDrive1          0           0            512          512

volume name          volume GUID          drive
-----
-  \Device\HarddiskVolume1  9298254f-0000-0000-0000-100000000000  0
C  \Device\HarddiskVolume2  9298254f-0000-0000-0000-104000000000  0
-  \Device\HarddiskVolume3  9298254f-0000-0000-0000-10401e000000  0

volume number  drive letter  filesystem type
-----
1              -           NTFS
2              C           NTFS
3              -           NTFS

C:\Windows\system32>

```

The drive \\.\PhysicalDrive0 is the only harddisk in this demo (a virtual machine). \\.\PhysicalDrive is our Debian Live USB stick.

The three volumes \Device\HarddiskVolume1 thru 3 represent the Windows Boot partition (no drive letter), the Window system drive C:, and the Windows Recovery Environment partition (also no drive letter).

You can also display the MBR partition table of a disk. Please note that the Windows version of part-y just requires the number of the physical disk to be specified:

```

C:\Windows\system32>e:\part-y info --disk=0
INFO: device file is \\.\PhysicalDrive0
INFO: device file is a physical disk: yes
INFO: vendor is 'NVMe', product is 'VMware_Virtual_N'
INFO: device size is 128.00 GB (268.435.456 sectors)
INFO: physical sector size is 4096, logical sector size is 512
INFO: MBR partition table: yes; GUID partition table: no
STATUS: MBR exists and is HEALTHY.
INFO: MBR disk signature is 0x9298254F

B  TY  C/H/S start    C/H/S end    start sec    sec num     size     type string
-----
MASTER BOOT RECORD:
* 07    1/ 0/ 1    0/ 63/32      2048    2097152    1.00 GB  HPFS/NTFS/exFAT [NTFS]
  07    1/ 0/ 1    0/ 63/32    2099200  251658240  120.00 GB  HPFS/NTFS/exFAT [NTFS]
  27    1/ 0/ 1    0/ 63/32  253757440   6291456    3.00 GB  Hidden NTFS Windows RE [NTFS]
  00    0/ 0/ 0    0/ 0/ 0        0          0          0  empty

Disk Map (MBR):
Start LBA          End LBA          Number of LBAs      Size      Occupied?
-----             -----             -----           -----
          0 ..          0 =                  1 (      512) ALLOCATED (USED) 'Master Boot Record (MBR)'
          1 ..          2047 =               2047 (1023.50 KB) UNALLOCATED (FREE) 'unallocated (free) space'
         2048 ..        2099199 =           2097152 (    1.00 GB) ALLOCATED (USED) 'HPFS/NTFS/exFAT'
        2099200 ..      253757439 =           251658240 (   120.00 GB) ALLOCATED (USED) 'HPFS/NTFS/exFAT'
      253757440 ..      260048895 =           6291456 (     3.00 GB) ALLOCATED (USED) 'Hidden NTFS Windows RE'
     260048896 ..      268435455 =           8386560 (     4.00 GB) UNALLOCATED (FREE) 'unallocated (free) space'

INFO: no GPT found.

```

The MBR partition table dump is followed by the so called ‘disk map’, which shows you all areas of the disk including free (unallocated) space.

Here are some more (partitioning) prerequisites before you can continue:

1. The sectors 1..33 have to be free because the primary GUID partition table is stored in these sector range. Normally, the first partition starts at 1MB (sector number 2048).
2. The last 33 sectors of the disk have to be free because the backup GPT is stored here.

You can use either MS Windows diskpart.exe or any other (free) tool, e.g. Minitool Partition Wizard (<https://de.minitool.com/partition-manager/partition-wizard-startseite.html>) to make room for the two GPTs. part-y displays error messages if these preconditions cannot be verified.

You are now ready to perform a **dry run** (nothing gets modified):

```
E:\part-y.exe preparewin10 --disk=0 --win-sys-drive=C: --linux-stick-drive=E: --locale=de-DE
```

```
C:\Windows\system32>e:\part-y preparewin10 --disk=0 --win-sys-drive=C: --linux-stick-drive=E: --locale=de-DE
INFO: forcing dry-run because --yes-do-it not specified.
CHECKING: Master Boot Record and all partitions clean .....: OK
CHECKING: GUID Partition Table MUST NOT be there .....: OK
CHECKING: Logical sector size has to be 512 (physical can be 4096) : OK
CHECKING: Locate MBR-style Windows Recovery Environment partition : OK
CHECKING: Locate MBR-style Windows Boot Partition .....: OK
    Windows Boot Partition at index 0 (MBR), size: 1.00 GB
CHECKING: Scanning Windows boot EFI directory on drive C: .....: OK
INFO : estimated size of EFI partition content is 16.683.008 byte(s) or approx. 15.91 MB.
CHECKING: Ensure that Linux stick contains a live system .....: OK
CHECKING: Ensure that conversion folder is not already there .....: OK
CHECKING: (Linux) executables and shared objects available .....: OK
CHECKING: Computing disk space requirements (req. by conversion) ...: OK
WORKING : Ensure that we have the windows system drive GUID .....: OK

HINT: If the selected device is an SSD and you have established
      'over provisioning', then you should TEMPORARILY disable it
      (e.g. using 'Samsung Magician' or similar tool).

GOOD: You are ready to go!
```

Because I am German, I use the locale de-DE (default is en-US) – just use the locale that matches your country and language. The locale is required to build the Boot Configuration Data (BCD) store.

You can add the switch --verbose to get more output.

The preparation run of part-y establishes a folder named ‘MBR2GPT’ on the Linux Live USB stick. It stores a backup of all disk sectors it is going to write to, a conversion file with information about the MBR to GPT conversion and a newly created Boot Configuration Data store (BCD). Furthermore, it copies all binaries (Linux) into this directory and also copies all EFI boot files from the Windows system32 directory to the folder \MBR2GPT\WindowsEFIFiles on the stick.

Just re-run the above-mentioned command adding ‘--yes-do-it’ as the final command line parameter:

```
C:\Windows\system32>e:\part-y preparewin10 --disk=0 --win-sys-drive=C: --linux-stick-drive=E: --locale=de-DE --yes-do-it
CHECKING: Master Boot Record and all partitions clean .....: OK
CHECKING: GUID Partition Table MUST NOT be there .....: OK
CHECKING: Logical sector size has to be 512 (physical can be 4096) : OK
CHECKING: Locate MBR-style Windows Recovery Environment partition : OK
CHECKING: Locate MBR-style Windows Boot Partition .....: OK
    Windows Boot Partition at index 0 (MBR), size: 1.00 GB
CHECKING: Scanning Windows boot EFI directory on drive C: .....: OK
INFO  : estimated size of EFI partition content is 16.683.008 byte(s) or approx. 15.91 MB.
CHECKING: Ensure that Linux stick contains a live system .....: OK
CHECKING: Ensure that conversion folder is not already there .....: OK
CHECKING: (Linux) executables and shared objects available .....: OK
CHECKING: Computing disk space requirements (req. by conversion) ...: OK
WORKING : Ensure that we have the windows system drive GUID .....: OK
WORKING : Creating backup of all sectors to be written .....: OK
WORKING : Verifying just created backup .....: OK
WORKING : Creating conversion data file .....: OK
WORKING : Create Boot Configuration Data (BCD) .....: OK
WORKING : Copy all Windows EFI file to the Linux Live stick .....: OK
WORKING : Copy all Linux EFI executables to the Linux Live stick ..: OK

GOOD: You are ready to go!
Please reboot your machine, switch to UEFI boot mode in the EFI
firmware and boot into the Live Linux system.
Execute the Linux version from /run/live/medium in a root prompt.
```

C:\Windows\system32>

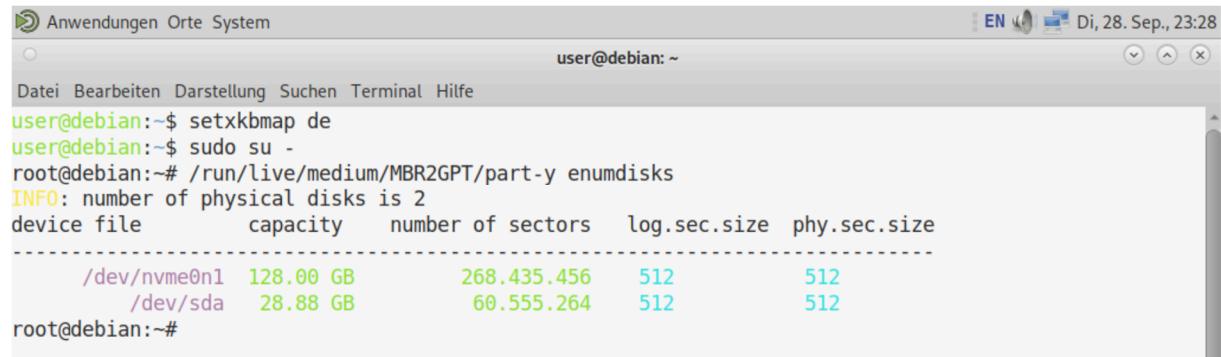
## Step 4: Switch to UEFI mode, boot the Linux live system

Please reboot your machine and enter the BIOS / UEFI firmware setup (e.g. hit Del or F2). Switch from BIOS boot mode (also named ‘CSM’ = ‘Compatibility Support Module’ to the UEFI boot mode.

Your Windows system will not boot anymore because it is still MBR-partitioned (still not prepared for the UEFI boot mode). Temporarily modify the boot order putting the USB stick (also ‘USB harddisk’ or similar) in front of all other boot options (UEFI jargon: ‘load options’).

Exit the BIOS setup / EFI firmware setup rebooting your machine. You should now see the GRUB2 bootloader screen coming from the Debian 11 Live system on the USB stick. You can either use the default boot entry or switch to the entry ‘with localization support’. This allows you to specify your favorite OS language before Debian continues to boot.

Once you see the desktop, open up a (MATE) terminal from the menu. You can use the Linux tool setxkbmap to set your favorite keyboard layout (default is US keyboard):



```
Anwendungen Orte System
EN Di, 28. Sep., 23:28
user@debian: ~
Datei Bearbeiten Darstellung Suchen Terminal Hilfe
user@debian:~$ setxkbmap de
user@debian:~$ sudo su -
root@debian:~# /run/live/medium/MBR2GPT/part-y enumdisks
INFO: number of physical disks is 2
device file      capacity   number of sectors  log.sec.size  phy.sec.size
-----
/dev/nvme0n1  128.00 GB       268.435.456      512           512
  /dev/sda    28.88 GB        60.555.264      512           512
root@debian:~#
```

Become root (`sudo su -`). The screenshot shows the location of our folder `MBR2GPT`, which was created in step 3. The screenshot also shows the difference between a Windows and a Linux system regarding the specification of a physical disk device.

On Windows, `\.\PhysicalDrive0`, `\.\PhysicalDrive1`, ... are used to specify a specific drive. On Linux, it is somewhat more complicated because you have to specify a block device starting with the

prefix /dev. SATA drives are named /dev/sda, /dev/sdb, etc. Nowadays SSDs are either connected to the SATA controller (also /dev/sdX) or to PCI express (NVM SSDs, M.2 slot). In the latter case, those devices are named /dev/nvme0n1, /dev/nvme0n2, etc.

In the screenshot above, /dev/nvme0n1 is the Windows disk drive that we want to convert, /dev/sda is the Linux live USB stick (here: 32GB). part-y stores the old Master Boot Record in the conversion file (folder MBR2GPT, file mbr2gpt.conversion) and compares this stored sector to the MBR of the disk drive that you specify on the command line. Because there is at least a specific 32bit disk signature in the MBR, it is very unlikely that part-y continues to run on the wrong disk device.

Nevertheless, if you are unsure about the correct disk device, you can use part-y with the command 'info' and the parameter '--disk=/dev/XXX' to dump the partition table of the device, e.g. (only top-half of the full output):

```
root@debian:~# /run/live/medium/MBR2GPT/part-y info --disk=/dev/nvme0n1
INFO: device file is /dev/nvme0n1
INFO: device file is a physical disk: yes
INFO: device size is 128.00 GB (268.435.456 sectors)
INFO: physical sector size is 512, logical sector size is 512
INFO: MBR partition table: yes; GUID partition table: no
STATUS: MBR exists and is HEALTHY.
INFO: MBR disk signature is 0x9298254F

B  TY  C/H/S start    C/H/S end      start sec      sec num      size      type string
-----
MASTER BOOT RECORD:
*  07    1/ 0/ 1      0/ 63/32        2048      2097152    1.00 GB  HPFS/NTFS/exFAT [NTFS]
   07    1/ 0/ 1      0/ 63/32        2099200   251658240  120.00 GB  HPFS/NTFS/exFAT [NTFS]
   27    1/ 0/ 1      0/ 63/32      253757440   6291456    3.00 GB  Hidden NTFS Windows RE [NTFS]
   00    0/ 0/ 0      0/ 0/ 0          0           0         0       empty

Disk Map (MBR):
  Start LBA             End LBA             Number of LBAs      Size      Occupied?
  -----
  0 ..                  0 =                   1 (      512) ALLOCATED (USED)
'Master Boot Record (MBR)'
  1 ..                  2047 =                2047 (1023.50 KB) UNALLOCATED (FREE)
'unallocated (free) space'
  2048 ..                2099199 =            2097152 ( 1.00 GB) ALLOCATED (USED)
'HPFS/NTFS/exFAT'
  2099200 ..              253757439 =        251658240 ( 120.00 GB) ALLOCATED (USED)

user@debian: ~
```

## Step 5: perform the conversion

After having found the correct disk device, execute:

```
/run/live/medium/MBR2GPT/part-y convertwin10 --disk=/dev/nvme0n1 --yes-do-it
```

(Please replace the disk device, you can also perform a dry run if you omit --yes-do-it). If you add --verbose, then more information is shown.

If you reboot your system, it should now boot into Windows 10 in UEFI mode. part-y finally called efibootmgr to establish a load option for MS Windows 10, which is put in front of all other boot options so that the USB Debian live stick can remain attached to the system.

If you execute part-y in info mode again (on your UEFI-booted Windows 10), you will see (part one of the screenshot):

```
C:\WINDOWS\system32>e:\part-y info --disk=0
INFO: device file is \\.\PhysicalDrive0
INFO: device file is a physical disk: yes
INFO: vendor is 'NVMe', product is 'VMware_Virtual_N'
INFO: device size is 128.00 GB (268.435.456 sectors)
INFO: physical sector size is 4096, logical sector size is 512
INFO: MBR partition table: yes; GUID partition table: yes
STATUS: PROTECTIVE MBR exists.
STATUS: primary GPT exists
STATUS: secondary/backup GPT exists
STATUS: primary and secondary/backup GPTs are consistent and HEALTHY.
INFO: MBR is a PROTECTIVE MBR.
INFO: MBR disk signature is 0x00000000

B  TY  C/H/S start    C/H/S end      start sec      sec num      size      type string
-----
MASTER BOOT RECORD:
EE  0/ 0/ 2 1023/255/63          1  268435455  128.00 GB  GPT (MBR followed by EFI header)
00  0/ 0/ 0   0/ 0/ 0           0   0           0           empty
00  0/ 0/ 0   0/ 0/ 0           0   0           0           empty
00  0/ 0/ 0   0/ 0/ 0           0   0           0           empty

Disk Map (MBR):
Start LBA          End LBA          Number of LBAs      Size      Occupied?
0 ..              0 =                  1 (      512) ALLOCATED (USED) 'Master Boot Record (MBR)'
1 ..          268435455 =            268435455 ( 128.00 GB) ALLOCATED (USED) 'GPT (MBR followed by EFI header)'

INFO: disk GUID is A47D0195-31B8-4661-B637-4E6DAFE495CC

revision .....: 0x00010000
header size .....: 0x0000005C
header CRC32 .....: 0xA77591F5
primary GPT at LBA .....: 1 -> is corrupt? no
backup GPT at LBA .....: 268435455 -> is corrupt? no
first usable LBA .....: 34
last usable LBA .....: 268435422
=> number of usable sectors ...: 268435389 is approx. 128.00 GB
part. entries CRC32 .....: 0xF9AC58C1

GPT partition entry 1 of 128:
Partition GUID .....: A47D0196-31B8-4661-B637-4E6DAFE495CC
Type GUID .....: C12A7328-F81F-11D2-BA4B-00A0C93EC93B => EFI System Partition (ESP)
Start and end LBA .....: 2048 to 2066431 (size approx. 1008.00 MB)
Partition attributes ....: BOOT| NOMOUNT
Partition name .....: ''
File system in partition : 'Windows FAT32'

GPT partition entry 2 of 128:
Partition GUID .....: A47D0197-31B8-4661-B637-4E6DAFE495CC
Type GUID .....: E3C9E316-0B5C-4DB8-817D-F92DF00215AE => Microsoft reserved
Start and end LBA .....: 2066432 to 2099199 (size approx. 16.00 MB)
Partition attributes ....: NONE (0x0)
Partition name .....: ''

GPT partition entry 3 of 128:
Partition GUID .....: A47D0198-31B8-4661-B637-4E6DAFE495CC
Type GUID .....: EBD0A0A2-B9E5-4433-87C0-68B6B72699C7 => Microsoft basic data
Start and end LBA .....: 2099200 to 253757439 (size approx. 120.00 GB)
Partition attributes ....: NONE (0x0)
Partition name .....: ''
```

A protective MBR was established followed by the GUID partition table (GPT). The legacy Windows boot partition was split and replaced by the EFI System Partition (ESP) and the 'Microsoft reserved' partition (16MB for Windows 10, was 128MB in Windows 8).

The third partition is the Window system drive C:. The remaining part of the screenshot:

```

GPT partition entry 4 of 128:
Partition GUID .....: A47D0199-31B8-4661-B637-4E6DAFE495CC
Type GUID .....: DE94BBA4-0601-4D40-A16A-BFD50179D6AC => Windows RE
Start and end LBA .....: 253757440 to 260048895 (size approx. 3.00 GB)
Partition attributes ....: NOMOUNT
Partition name .....: ''
File system in partition : 'Windows NTFS'

```

#### Disk Map (GPT):

Start LBA	End LBA	Number of LBAs	Size	Occupied?	
0 ..	0 =	1 ( 512)	ALLOCATED (USED)	'Master Boot Record (MBR)'	
1 ..	1 =	1 ( 512)	ALLOCATED (USED)	'GPT header (primary)'	
2 ..	33 =	32 ( 16.00 KB)	ALLOCATED (USED)	'GPT entries (primary)'	
34 ..	2047 =	2014 (1007.00 KB)	UNALLOCATED (FREE)	'unallocated (free) space'	
2048 ..	2066431 =	2064384 (1008.00 MB)	ALLOCATED (USED)	'EFI System Partition (ESP)'	
2066432 ..	2099199 =	32768 ( 16.00 MB)	ALLOCATED (USED)	'Microsoft reserved'	
2099200 ..	253757439 =	251658240 ( 120.00 GB)	ALLOCATED (USED)	'Microsoft basic data'	
253757440 ..	260048895 =	6291456 ( 3.00 GB)	ALLOCATED (USED)	'Windows RE'	
260048896 ..	268435422 =	8386527 ( 4.00 GB)	UNALLOCATED (FREE)	'unallocated (free) space'	
268435423 ..	268435454 =	32 ( 16.00 KB)	ALLOCATED (USED)	'GPT entries (secondary)'	
268435455 ..	268435455 =	1 ( 512)	ALLOCATED (USED)	'GPT header (backup)'	

The fourth partition contains the Windows Recovery Environment (WinRE).

Using the system information display tool, you can verify that the UEFI boot mode is active:

The screenshot shows the Windows System Information window. On the left, there's a navigation pane with links like 'Systemübersicht', 'Hardwareressource', 'Komponenten', and 'Softwareumgebung'. The main area has tabs for 'Systemübersicht' (selected), 'Datei', 'Bearbeiten', 'Ansicht', and '?'. Below these tabs is a table with two columns: 'Element' and 'Wert'. The table contains the following data:

Betriebssystemname	Microsoft Windows 10 Pro
Version	10.0.19043 Build 19043
BIOS-Version/-Datum	VMware, Inc. VMW71.00V.16221537.864.2005150253, 15.05.2020
SMBIOS-Version	2.7
Version des eingebetteten Controllers	255.255
BIOS-Modus	UEFI

## Step 6: manual post conversion steps

Although the conversion worked, some parts have to be manually adjusted after the conversion.

First of all, open up an Administrator command prompt. If you enter:

```
bcdedit /enum all
```

then the tool outputs (German Windows 10):

```
C:\WINDOWS\system32>bcdedit /enum all
Der Speicher für die Startkonfigurationsdaten konnte nicht geöffnet werden.
Der Datenträger einer Datei wurde extern so geändert, dass die geöffnete Datei nicht mehr gültig ist.

C:\WINDOWS\system32>
```

It says that the drive configuration was changed and it cannot open the BCD system store. Of course the BCD store is fine or your Windows 10 would not have booted correctly. `bcdedit.exe` can just not use it. Don't panic, this sequence fixes it:

```

diskpart.exe
list volume

C:\WINDOWS\system32>diskpart.exe

Microsoft DiskPart-Version 10.0.19041.964

Copyright (C) Microsoft Corporation.
Auf Computer: DESKTOP-HJKUPL4

DISKPART> list volume

  Volume ###  Bst  Bezeichnung  DS      Typ          Größe    Status     Info
  -----  -----  -----  -----  -----
  Volume 0    D                DVD-ROM        0 B  Kein Medi
  Volume 1    C                NTFS   Partition  120 GB Fehlerfre Startpar
  Volume 2          EFI-SYSTEM  FAT32  Partition  1008 MB Fehlerfre System
  Volume 3          NTFS   Partition  3072 MB Fehlerfre Versteck
  Volume 4    E    LINUX-MBR2G  FAT32 Wechselmed  28 GB  Fehlerfre

DISKPART>

```

The volume we are looking for is labeled ‘EFI-SYSTEM’ (here: number 2). Assign a drive letter to it, e.g. letter ‘T’:

```

select volume 2
assign letter=T
exit

DISKPART> select volume 2

Volume 2 ist jetzt das gewählte Volume.

DISKPART> assign letter=T

Der Laufwerkbuchstabe oder der Bereitstellungspunkt wurde zugewiesen.

DISKPART> exit

Datenträgerpartitionierung wird beendet...

C:\WINDOWS\system32>bcdedit /import T:\EFI\Microsoft\Recovery\BCD
Der Vorgang wurde erfolgreich beendet.

```

The final command:

```
bcdedit /import T:\EFI\Microsoft\Recovery\BCD
```

imports the identical Boot Configuration Store that part-y has created and stored in the above shown path again. But now bcdedit.exe can access the BCD store as its system store (you can check it with bcdedit /enum all).

If your Windows Recovery Environment does not work anymore, you can try to repair it using the following sequence (in an Administrator command prompt):

```
ReAgentc.exe /disable
ReAgentc.exe /enable
```

The Microsoft websites (on docs.microsoft.com) contain more repair options for Windows RE. Normally, it should work after you have executed the bcdedit steps above.

## Troubleshooting

If your system does not boot anymore after the conversion, then boot again into the Debian Live system on the USB stick.

To revert all changes applied to your drive, enter (as root):

```
/run/live/medium/MBR2GPT/part-y restore --disk=<your disk device> --backup-file ↴  
/run/live/medium/MBR2GPT/mbr2gpt.backup --yes-do-it
```

Reboot your machine, enter the BIOS setup and re-activate the legacy BIOS boot mode (or CSM mode). Your Windows machine should boot as before (using its MBR-partitioned drive).