Creating bootable USB from ISO in MacOS

I needed to reinstall one of my Windows PCs at home recently, and realised that I only have Windows 7 DVDs, but no Windows 10s.

Since it’s possible to boot and install Windows 10 from USB, that’s what I decided to do: download Windows 10 ISO and burn it on one of the USB sticks I have. This approach will be simpler for any other ISO image like Ubuntu or CentOS, so there’s a separate Steps 3-5 for them at the end of the post.

Here are the steps I had to take on my MacBook, originally procedure was written in macOS High Sierra, but I tested it in MacOS Mojave 10.14.2 as well. All of these need to be done in the Terminal app.

Windows 10 bootable USB vs Linux
ISO USB

Windows 10 is a very specific bootable media, so the process is a bit involved. For generic Linux installation ISOs this procedure would be simpler. So the first two steps are the same, then Windows and Linux have separate procedure.

Here’s what we will do for Windows ISO:

- erase USB stick and create MBR (master boot record)
- format USB stick with FAT-32 filesystem
- mount Windows 10 ISO image as a folder
- copy all the files from that folder onto the USB stick

Here’s what we would do for Linux ISO:

- use dd command to burn Linux installation ISO onto USB stick

Step 1: Confirm partitions before inserting USB

Use the diskutil command to list all disks and partitions on your Mac. This is simply so that you can run the same command again after inserting USB and easily spot the USB device.

```
greys@maverick:/ $ diskutil list
/dev/disk0 (internal):
#: TYPE NAME SIZE IDENTIFIER
0: GUID_partition_scheme 500.3 GB disk0
1: EFI EFI 314.6 MB disk0s1
2: Apple_APFS Container disk1 500.0 GB disk0s2

/dev/disk1 (synthesized):
#: TYPE NAME SIZE IDENTIFIER
0: APFS Container Scheme - +500.0 GB disk1
Physical Store disk0s2
1: APFS Volume Macintosh HD 466.2 GB disk1s1
2: APFS Volume Preboot 24.3 MB disk1s2
```
3: APFS Volume Recovery 519.0 MB disk1s3
4: APFS Volume VM 12.3 GB disk1s4

/dev/disk2 (disk image):
#: TYPE NAME SIZE IDENTIFIER
0: GUID_partition_scheme +30.0 GB disk2
1: EFI EFI 209.7 MB disk2s1
2: Apple_APFS Container disk3 29.8 GB disk2s2

/dev/disk3 (synthesized):
#: TYPE NAME SIZE IDENTIFIER
0: APFS Container Scheme - +29.8 GB disk3
Physical Store disk2s2
1: APFS Volume Photos 9.2 GB disk3s1

/dev/disk4 (disk image):
#: TYPE NAME SIZE IDENTIFIER
0: Apple_partition_scheme +173.1 MB disk4
1: Apple_partition_map 32.3 KB disk4s1
2: Apple_Driver_ATAPI 4.1 KB disk4s2
3: Apple_HFSX Firefox 173.0 MB disk4s3

/dev/disk5 (disk image):
#: TYPE NAME SIZE IDENTIFIER
0: GUID_partition_scheme +80.0 MB disk5
1: Apple_APFS Container disk6 80.0 MB disk5s1

/dev/disk6 (synthesized):
#: TYPE NAME SIZE IDENTIFIER
0: APFS Container Scheme - +80.0 MB disk6
Physical Store disk5s1
1: APFS Volume Raven-Core 70.1 MB disk6s1

/dev/disk7 (disk image):
#: TYPE NAME SIZE IDENTIFIER
0: GUID_partition_scheme +18.1 MB disk7
1: Apple_HFS F-Secure XFENCE 18.1 MB disk7s1

/dev/disk8 (external, physical):
#: TYPE NAME SIZE IDENTIFIER
0: GUID_partition_scheme *3.0 TB disk8
Step 2: Insert USB stick an re-run diskutil list

This time I can see there’s a disk9 showing up, with 64GB of size which matches my USB stick size:

greys@maverick:/ $ diskutil list
/dev/disk0 (internal):
#: TYPE NAME SIZE IDENTIFIER
0: GUID_partition_scheme 500.3 GB disk0  
1: EFI EFI 314.6 MB disk0s1  
2: Apple_APFS Container disk1 500.0 GB disk0s2

/dev/disk1 (synthesized):
#: TYPE NAME SIZE IDENTIFIER
0: APFS Container Scheme - +500.0 GB disk1  
Physical Store disk0s2  
1: APFS Volume Macintosh HD 466.2 GB disk1s1  
2: APFS Volume Preboot 24.3 MB disk1s2  
3: APFS Volume Recovery 519.0 MB disk1s3  
4: APFS Volume VM 12.3 GB disk1s4

/dev/disk2 (disk image):
#: TYPE NAME SIZE IDENTIFIER
0: GUID_partition_scheme +30.0 GB disk2  
1: EFI EFI 209.7 MB disk2s1  
2: Apple_APFS Container disk3 29.8 GB disk2s2

/dev/disk3 (synthesized):
#: TYPE NAME SIZE IDENTIFIER
0: APFS Container Scheme - +29.8 GB disk3  
Physical Store disk2s2  
1: APFS Volume Photos 9.2 GB disk3s1

/dev/disk4 (disk image):
#: TYPE NAME SIZE IDENTIFIER
0: Apple_partition_scheme +173.1 MB disk4
1: Apple_partition_map 32.3 KB disk4s1
2: Apple_Driver_ATAPI 4.1 KB disk4s2
3: Apple_HFSX Firefox 173.0 MB disk4s3

/dev/disk5 (disk image):
#: TYPE NAME SIZE IDENTIFIER
0: GUID_partition_scheme +80.0 MB disk5
1: Apple_APFS Container disk6 80.0 MB disk5s1

/dev/disk6 (synthesized):
#: TYPE NAME SIZE IDENTIFIER
0: APFS Container Scheme - +80.0 MB disk6
Physical Store disk5s1
1: APFS Volume Raven-Core 70.1 MB disk6s1

/dev/disk7 (disk image):
#: TYPE NAME SIZE IDENTIFIER
0: GUID_partition_scheme +18.1 MB disk7
1: Apple_HFS F-Secure XFENCE 18.1 MB disk7s1

/dev/disk8 (external, physical):
#: TYPE NAME SIZE IDENTIFIER
0: GUID_partition_scheme *3.0 TB disk8
1: EFI EFI 209.7 MB disk8s1
2: Apple_HFS Backups 3.0 TB disk8s2

/dev/disk9 (external, physical):
#: TYPE NAME SIZE IDENTIFIER
0: NONAME *62.1 GB disk9

Chances are, your USB stick has a PC standard FAT16/FAT32 filesytem, and will be mounted automatically. Let’s check:

greys@maverick:/ $ df -h | grep disk9
/dev/disk9  4.1Gi  4.1Gi  0Bi 100% 18446744073707414141 21385571704874683337099264% /Volumes/NONAME

Sure enough, it’s mounted as /Volumes/NONAME!
Creating Windows bootable USB

**IMPORTANT**: these steps are only needed for Windows ISOs. If you’re burning a Linux ISO, skip this and go to the Creating Linux bootable USB section below, it has its own steps.

**Step 3: format USB stick for Windows boot**

This single command will erase the USB stick, make it bootable by adding a Master Boot Record (MBR) and create a Windows-friendly filesystem (FAT) on it:

```
greys@maverick:~ $ diskutil eraseDisk MS-DOS "WIN10" MBR disk9
```

Started erase on disk9

Unmounting disk

Creating the partition map

Waiting for partitions to activate

Formatting disk9s1 as MS-DOS (FAT) with name WIN10

512 bytes per physical sector

/dev/rdisk9s1: 30000480 sectors in 1875030 FAT32 clusters
(8192 bytes/cluster)

bps=512 spc=16 res=32 nft=2 mid=0xf8 spt=32 hds=255 hid=2048
drv=0x80 bsec=30029824 bspf=14649 rdcl=2 infs=1 bkbs=6

Mounting disk

Finished erase on disk9

You’ll have the new filesystem mounted automatically, as you can see it’s empty (1.1MB used of 14GB):

```
greys@maverick:/ $ df -h | grep disk9
/dev/disk9s1 14Gi 1.1Mi 14Gi 1% 0 0 100% /Volumes/WIN10
```

Great stuff, now we’re moving to the next phase: we need to mount the Windows ISO image as a folder and then copy all the files onto the freshly formatted /Volumes/WIN10 USB stick.

**Step 4: Mount Windows ISO**

Simply go to the directory where you have the ISO image and run the open command for it:
You’ll have the MacOS Finder window with Windows ISO content pop-up, just ignore it and go back to the terminal.

You should now have it mounted under folder like this:

greys@maverick:/Volumes/SSD $ df -h | grep EN-US
/dev/disk8 4.1Gi 4.1Gi 0Bi 100% 18446744073707414141 2138557
1704874683337099264% /Volumes/CCSA_X64FRE_EN-US_DV5

Go into that folder and do ls to make sure you can see Windows DVD files there:

greys@maverick:/Volumes/SSD $ cd /Volumes/CCSA_X64FRE_EN-US_DV5/
greys@maverick:/Volumes/CCSA_X64FRE_EN-US_DV5 $ ls
autorun.inf boot bootmgr bootmgr.efi efi efi setup.exe sources support

Excellent!

**Step 5: Copy Windows files onto USB stick**

Right, now we have to copy all the files from DVD ISO image onto our new USB stick.

**IMPORTANT:** you should be in the /Volumes/CCSA_X64FRE_EN-US_DV5 directory when running this command below. That’s exactly where you are if you just completed Step 4.

The copy will take good 10-15min:

$ sudo cp -r . /Volumes/WIN10/

Password:

There should be no errors returned.

If you remember, in Step 3 we did the `df` command and the USB filesystem /Volumes/WIN10 was empty.

Just to get an idea that the copy was successful, let’s run
the **df command** again to make sure the USB stick has roughly the same amount taken up now – 4.1GB as per output fragments in previous step.

greys@maverick:/ $ df -h | grep WIN10
/dev/disk7s1 14Gi 4.1Gi 10Gi 29% 0 0 100% /Volumes/WIN10

Looks good! What’s left now is to eject the USB stick:

greys@maverick:/ $ diskutil unmountDisk /dev/disk9
Unmount of all volumes on disk9 was successful

and to boot it on the PC as shown below.

**Step 6: Boot USB stick on your PC**

**IMPORTANT:** recent enough PCs will likely offer you both legacy boot and UEFI boot. The rule of thumb is to boot UEFI if it’s available, especially if you’re booting Windows USB stick:
Creating Linux bootable USB

As I said earlier, non-Windows ISOs are much easier to deal with! So after the first 2 generic steps at the start of this post, we’ll be taking separate steps for Linux as shown below.

We need to unmount the USB disk we found in Step 2. It’s /dev/disk9 in our case:

greys@maverick:/ $ diskutil unmountDisk /dev/disk9
Unmount of all volumes on disk9 was successful

Step 3: convert ISO into bootable DMG

WARNING: this is an important step, without it your USB stick won’t boot.

So I have this DVD ISO of the CentOS 7.4, let’s convert it before burning onto the USB stick.

$ hdiutil convert -format UDRW -o centos ./CentOS-7-x86_64-DVD-1708.iso
Reading Master Boot Record (MBR : 0)...[
Reading CentOS 7 x86_64 (Apple_ISO : 1)...[
Reading (Type EF : 2)...[
Reading CentOS 7 x86_64 (Apple_ISO : 3...[

...............................................................

Elapsed Time: 10m 15.681s
Speed: 7.0Mbytes/sec
Savings: 0.0%
created: /Volume/Stuff/dist/centos.dmg

Step 4: write DMG image onto USB stick with dd

Exellent, now we are ready to burn the ISO image. Let’s cd into the directory with all the ISOs that you have and burn it using the dd_command.

dd_command runs with the help of 3 command line parameters:
- **if** – specifying source data, that’s our ISO image file
- **of** – specifying the destination file or device, /dev/disk9 in our case
- **bs** – block size, just needs to be something big enough like 1m (meaning 1 megabyte)

```
greys@maverick:/ $ cd /Volumes/Stuff/dist
greys@maverick:/Volumes/Stuff/dist $ sudo dd if=./centos.dmg of=/dev/disk9 bs=1m
```

Depending on the ISO size and the USB stick speed, you’ll have to wait 10-20min for this to complete.

**Useful tip (cheers Jim!):** if you want to confirm your current process, press Ctrl+t in the Terminal app and it should update output with dd progress (I highlighted the output below every time I pressed Ctrl+t):

```
load: 2.53 cmd: dd 11985 uninterruptible 0.00u 0.05s
83+0 records in
82+0 records out

85983232 bytes transferred in 7.441006 secs (11555324 bytes/sec)
load: 2.48 cmd: dd 11985 uninterruptible 0.01u 1.26s
2248+0 records in
2247+0 records out

2356150272 bytes transferred in 222.985555 secs (10566381 bytes/sec)
load: 2.24 cmd: dd 11985 uninterruptible 0.01u 1.94s
3422+0 records in
3421+0 records out

3587178496 bytes transferred in 339.752252 secs (10558218 bytes/sec)
```

finally, **dd** will complete a message like this:

```
4177+1 records in
4177+1 records out
4380387328 bytes transferred in 415.014562 secs (10554780 bytes/sec)
```
Step 5: Boot USB stick on your PC

That’s it, we’re done! Eject the USB stick (no need to run any diskutil commands as it wasn’t mounted automatically) and boot on your PC.

**IMPORTANT**: recent enough PCs will likely offer you both legacy boot and UEFI boot. The rule of thumb is to boot UEFI if it’s available:

**See Also**

- [ISO to USB in MacOS](#)
- [Bootable USB from ISO with Etcher](#)
- [MacOS](#)
- [dd command](#)