

---

## Setting up a UEFI triple-boot computer: Windows 10 + FreeBSD + Linux

### A preliminary word about UEFI

The UEFI implementation is fundamentally flawed in one respect: the choice of filesystem for the ESP (EFI system partition). For reasons of lowest common denominator as well as subservience to Microsoft, the UEFI implementers chose msdos as the filesystem, whereas the lowest common denominator itself had to be upped to ext2 – the Linux filesystem that every operating system can or at least should understand.

The ESP directory hierarchy lacks any meaningful standards, causing serious scope for confusion for multi-boot systems. The ideal ESP directory hierarchy should have been as described below.

( With the ESP mounted at `/boot/efi`, your EFI directory – or perhaps more correctly your EFI Loader directory, is accessible as `/boot/efi/EFI` )

```
/boot/efi/EFI
→      /boot/efi/EFI/Windows[/loader.efi]
→      /boot/efi/EFI/FreeBSD[/loader.efi]
→      /boot/efi/EFI/Linux[/loader.efi]
→      /boot/efi/EFI/boot.efi ->
```

The symbolic link `/boot/efi/EFI/boot.efi`, when present, would point to one of the OS loader `.efi` files and would establish the default operating system to `[chain]load`.

Since msdos does not support symbolic links, the filesystem of choice for the ESP should have been ext2 – a superior and Open Source filesystem. It is msdos instead.

All operating systems can and should use a single system-wide EFI partition. Multiple EFI partitions are accepted but deprecated. Usually, your first hard disk's first GPT partition should be your ESP, as well as the only EFI partition in the whole computer. This makes multi-booting easier.

As you live with UEFI, you will realize how nightmarish the situation is on account of lack of standards. It is almost as if the UEFI implementers went to their study tables bereft of any concepts. The cost of this oversight is that setting up multi-boot under UEFI is significantly more cumbersome (compared to Legacy BIOS mode), as well as the maintenance needed after each OS install/reinstall is significant.

Linux, typically and quite passably, creates its EFI loader subdirectory named after the distro. But currently, Windows and FreeBSD have the nasty habit of creating their EFI loader subdirectory named as `boot` (or `Boot` – it does not matter because msdos filesystem is case-insensitive). Since both Windows and FreeBSD want their EFI loader to be stored inside a subdirectory named `Boot`, we have to take special measures to install Windows 10 and FreeBSD side-by-side.

---

This write-up assumes the following non-standard definitions and meanings:

BIOS : Basic Input Output System (The motherboard firmware that regulates how your motherboard and attached hardware broadly work)

UEFI : Universally Extensible Firmware Interface (One of the two modes your BIOS can be placed in, the other being Legacy – or Legacy BIOS)

---

Let's assume your computer currently has a single internal hard disk (SATA/NVME; SATA assumed in this write-up), and you want to triple-boot Windows 10, FreeBSD and Linux. To set up triple-boot, your hard disk needs to be at least 120 GB.

This article covers single-disk triple-boot installation in GPT+UEFI mode. After all installation is done, you will have no dependency on your BIOS's Boot Menu for selecting which operating system to boot.

We are first going to install Windows 10. Windows 10 installation gets hinted as to which route to take (GPT+UEFI or MBR+BIOS) by the installation medium.

- (a) If you are installing the OS via DVD: did the DVD get booted in UEFI mode? If yes, the installation will be GPT+UEFI. Else MBR+BIOS.
- (b) If you are installing the OS via USB stick:
  - i) Was the USB stick partitioned GPT?, and
  - ii) Did the USB stick get booted in UEFI mode?

If either answer is yes, Windows 10 will gravitate towards GPT+UEFI mode installation. Else MBR+BIOS.

---

1. Our first step is to get the Windows 10 (hereon referred as W10) installation medium ready for GPT+UEFI mode installation. You can download the W10 ISO image at the following Media Creation Tool URL:

<https://www.microsoft.com/en-in/software-download/windows10ISO>

If you are using a DVD, things are simple: you just need to buy a dual-layer DVD, and then burn the W10 ISO image to the DVD on an operating system that understands the UDF filesystem – Windows 10 or Linux. (The normal 4.7 GB DVD won't do the job: the W10 ISO – between 5 and 6 GB – is too big for the normal DVD.)

If you are using a USB stick, you need to:

- (a) partition the stick as GPT with a single partition of type Microsoft basic data, and format the partition as NTFS (also see footnote 1 at the end of document); and
- (b) Then access a system running either Windows 10 or Linux (or any OS that supports the UDF filesystem). The one-time access is required to copy the contents of the Windows 10 ISO to the stick.

On Windows 10: right-click/double-click on the W10 ISO to mount it.

On Linux: use `'mount -t udf -o ro -o loop /path/to/w10.iso /mnt'`

Once the mount succeeds, copy the ISO's contents to your stick's NTFS partition.

Note that your USB stick is now already bootable – not intuitive (because no boot sector was ever written to the USB stick), but important to remember.

We are now ready for the big time adventure.

2. Enter your BIOS setup and fully enable UEFI – which means, disable CSM (Compatibility Support Module).
3. Boot with your Linux/GParted CD/DVD in UEFI mode. (Tip: prefer GParted because it boots much quicker than a Linux installer CD/DVD.) Use `fdisk` to partition your disk with 6 partitions:

```
# fdisk /dev/sda
```

(Use command `g` to create a new GPT table)

(Use command `n` to add a new partition)

(Use command `d` to delete an existing partition)

(Use command `t` to change the partition type)

(Use command `p` to print the partition table)

(Use command `w` to write out the final partition table)

```
sda1 : 560 MB : type 1 (EFI System)
sda2 : 34 GB/space for Windows 10 : type 11 (MS basic data)
sda3 : minimum 8 GB/swap space : type 19 (Linux swap)
sda4 : 34 GB/space for FreeBSD : type 42 (FreeBSD UFS)
sda5 : 34 GB/space for Linux : type 35 (Linux filesystem)
sda6 : space left/shared partition : type 35 (Linux filesystem)
```

Although `sda3` is of type `Linux swap`, FreeBSD can readily use it as swap space. `sda6` is your shared partition, readable and writeable by all operating systems.

After exiting `fdisk`, run the next commands:

```
mkswap /dev/sda3
mkfs.ext2 /dev/sda6
```

4. Boot with your Windows 10 DVD/USB stick. You'll have to press a Function Key to trigger the Boot Menu. Make sure the installation medium gets booted UEFI.

At the partitioning page, select `sda2` and then click `Next`. Install Windows 10.

Note that your hard disk's first partition is now your ESP: EFI system partition. We have to repeatedly manage this partition's data to enable multi-boot.

5. Reformat your USB stick as FAT32.
6. Boot with your FreeBSD CD/DVD (UEFI mode again) and get into Shell mode with your USB stick attached at the `/dev/da0` device node.

```
mount -t msdosfs /dev/da0p1 /mnt
mkdir /tmp/mnt2
mount -t msdosfs /dev/ada0p1 /tmp/mnt2
cd /tmp/mnt2/EFI
mv Boot Windows
cd ..
cp -R EFI /mnt/
umount /tmp/mnt2
umount /mnt
exit
```

What we did above is rename `Boot` (the EFI loader subdirectory created by Windows 10 installer) as `Windows`, and then backup the EFI directory itself to the USB stick.

Return to the FreeBSD installer and install FreeBSD – mounting the ESP at `/boot/efi` (partition type: `efi`), and using `/dev/ada0p4` as the `/` filesystem. If FreeBSD prompts you, as it likely will, for a separate boot partition ("This kind of partition requires a separate `/boot`. Would you like to create one now?"), choose `No`. This will force FreeBSD to reuse the ESP for EFI.

7. After FreeBSD installation is done, reboot the system into FreeBSD and then repeat the EFI exercise to backup the FreeBSD EFI code too:

```
mount -t msdosfs /dev/da0p1 /mnt
cd /boot/efi/EFI
mv boot FreeBSD
cp -R FreeBSD /mnt/EFI/
umount /mnt
```

You may optionally edit `/etc/fstab` to set `/dev/ada0p3` as your swap partition.

8. Now finally you can boot with your Linux CD/DVD (UEFI mode) to install Linux, using `/dev/sda5` as the `/` filesystem, `/dev/sda3` as the swap partition, and mounting the ESP at `/boot/efi` again – this time choosing to reformat all the three partitions, including the ESP partition.

---

Once Linux installation finishes, boot Linux and look up the `/boot/efi/EFI` directory. Restore the following three directories from your USB stick: `Microsoft` ; `FreeBSD` ; `Windows` – and then rename `Windows` as `Boot` :

```
mount -o ro /dev/sdb1 /mnt
cd /mnt/EFI
cp -R Windows FreeBSD Microsoft /boot/efi/EFI/
umount /mnt
cd /boot/efi/EFI
mv Windows Boot
```

The last frontier is getting the triple-boot to work. For this, just install the Linux package `refind`, and then run the command `'refind-install'`.

This sets up the `rEFInd` bootloader in the ESP, which at every boot will hereon give you the option to boot into `Windows 10` or `FreeBSD` or `Linux`.

It helps if you could instruct your BIOS to use `rEFInd` as the default boot option. It further helps to set the options below in `/boot/efi/EFI/refind/refind.conf` :

```
hideui all
showtools shutdown
scan_all_linux_kernels false
```

## A closing word about the `rEFInd` directory layout

Apart from the subdirectory `refind` itself and the subdirectory `Microsoft` (which should generally be ignored), each subdirectory in the `EFI` directory (`/boot/efi/EFI`) should reflect an installed operating system. The `EFI` loader filename can be anything ending with the extension `.efi` :

```
/boot/efi/EFI
→ /boot/efi/EFI/refind/refind64.efi
→ /boot/efi/EFI/Boot/bootx64.efi
→ /boot/efi/EFI/FreeBSD/loader64.efi
→ /boot/efi/EFI/Linux/grubx64.efi
→ /boot/efi/EFI/Microsoft[/IGNORE]
```

The key points to note are that `Windows 10`'s `EFI` loader is in a directory called `Boot`, while the directory `Microsoft` contains `Microsoft`-specific data (e.g., files for `BCD` – `Boot Configuration Data`) that should be ignored by other operating systems.

---

### Footnote 1:

`Windows 10` does not easily give the option to create a `GPT` partition schema on a USB stick. You can, for this purpose, download `Rufus` (`Windows`-only) from this URL: <https://rufus.ie/>

You can then burn the `Windows 10` ISO to your USB stick with `Rufus` using `GPT+UEFI` configuration.