GRUB2 & EFI recovery - Tutorial

Updated: April 15, 2015

This guide will mostly be useful to people well familiar with the GRUB/GRUB bootloader, and Linux in general. If you're a newbie, you're better off reading my original tutorials on this topic first, before trying anything written here. That said, you've probably reached this page because you have trouble recovering your GRUB bootloader on a UEFI system.

This can happen if you've had a Linux distro, e.g. Ubuntu or Mint happily installed on a machine, and then you added Windows, which ruined the bootloader. Now, you're trying to recover/restore GRUB, but the standard procedure that I've outlined in my GRUB2 guide is not working. Which is why we are here.

Problem

Let's take a real practical scenario. My G50 machine. Until recently, it hosted Windows 8.1, which comes preinstalled plus a bunch of hidden recovery partitions by the vendor, Ubuntu, Netrunner, and Mint. Then, I also added Windows 10 Technical Preview into the empty space left reserved for exactly this purpose, in between Windows 8 and Ubuntu.

You are booting in the UEFI mode, with or without Secure Boot enabled, and it worked flawlessly, until after the Windows 10 install. Now, to restore Ubuntu, you will need to boot into a live session, follow the instructions, and then fail, because they won't work. What you'll see is this, including double spacing and periods and all:

```
sudo grub-install --root-directory /mnt /dev/sda
Installing for i386-pc platform.
grub-install: warning: this GPT partition label contains no BIOS Boot Partition; embedding won't be possible.
grub-install: warning: Embedding is not possible.  GRUB can only be installed in this setup by using blocklists.  However, blocklists are UNRELIABLE and their use is discouraged..
grub-install: error: will not proceed with blocklists.
```

Solution 1: Bootloader install

This is going to be a lengthy procedure, but it will work. I will demonstrate with Ubuntu, but the basic logic applies for most distributions, although few of them support UEFI natively at the moment. But in the coming years, this will become more of an issue.

Basically, we have three possible ways of doing this. If Ubuntu is properly configured as an entry in the UEFI boot section, then we just need to install the EFI-supported version of GRUB2. Done.

```
sudo apt-get install grub-efi-amd64
```

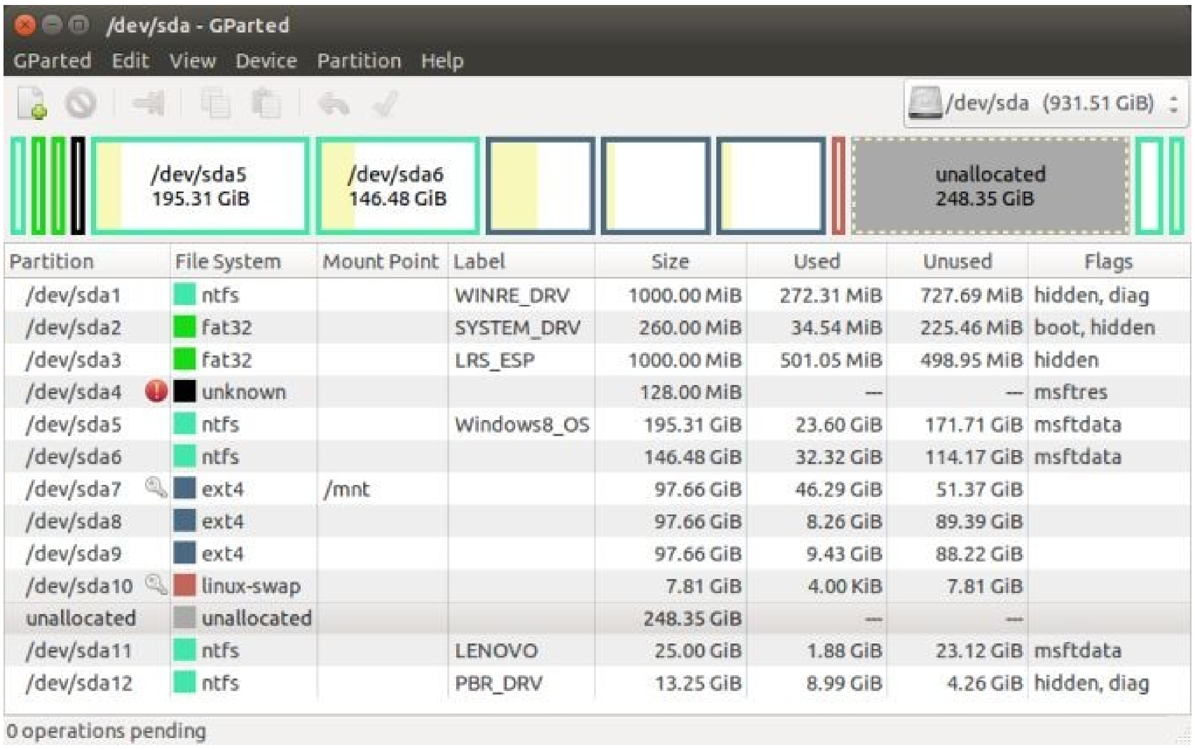
Solution 2: EFIBOOTMGR

Now, let's assume that we're not really quite sure what's happening. Let's first locate the EFI partition. You can treat it the way old /boot used to be, especially when dealing with LVM and such. Certain filesystems could not support advanced boot features, and vice versa, bootloaders could not support specific filesystem types, so the sensible compromise was to use a small /boot partition, formatted with the simplest filesystem available, thus working around any BIOS, filesystem and other limitations.

Here, the concept is the same. UEFI systems come with a small partition, usually around 200-300MB, formatted as FAT32. The technical details are a little more complex than what we had with GRUB Legacy and stage1 and stage2, but the idea is essentially identical.

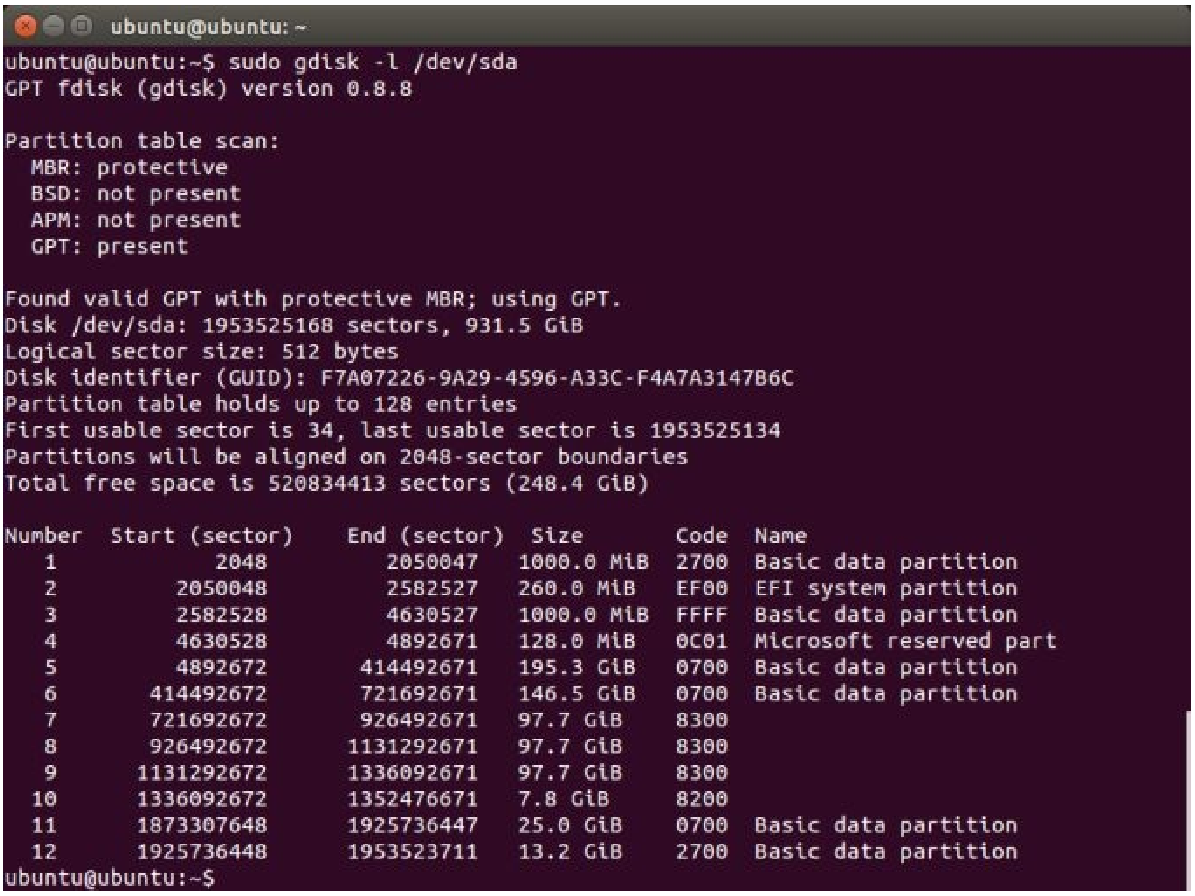
When the system starts, UEFI boot manager checks and accesses the EFI partition and loads the first marked image. In our case, this should be Ubuntu, and if there's no entry, we will add one. First, let's discover the partition.

There are two ways about it. We will use GParted as well as gdisk, which is the GPT-supported version of fdisk. If the program is not available in the live session, then you can install it with sudo apt-get install gdisk. We are looking for a FAT32 partition, with the boot flag. Indeed, we can see it is the second partition, or /dev/sda2. In GParted, we can see this based on the boot flag, and gdisk tells us it's EFI System Partition (ESP).



Partition	File System	Mount Point	Label	Size	Used	Unused	Flags
/dev/sda1	ntfs		WINRE_DRV	1000.00 MiB	272.31 MiB	727.69 MiB	hidden, diag
/dev/sda2	fat32		SYSTEM_DRV	260.00 MiB	34.54 MiB	225.46 MiB	boot, hidden
/dev/sda3	fat32		LRS_ESP	1000.00 MiB	501.05 MiB	498.95 MiB	hidden
/dev/sda4	unknown			128.00 MiB	---	---	msftres
/dev/sda5	ntfs		Windows8_OS	195.31 GiB	23.60 GiB	171.71 GiB	msftdata
/dev/sda6	ntfs			146.48 GiB	32.32 GiB	114.17 GiB	msftdata
/dev/sda7	ext4	/mnt		97.66 GiB	46.29 GiB	51.37 GiB	
/dev/sda8	ext4			97.66 GiB	8.26 GiB	89.39 GiB	
/dev/sda9	ext4			97.66 GiB	9.43 GiB	88.22 GiB	
/dev/sda10	linux-swap			7.81 GiB	4.00 KiB	7.81 GiB	
unallocated	unallocated			248.35 GiB	---	---	
/dev/sda11	ntfs		LENOVO	25.00 GiB	1.88 GiB	23.12 GiB	msftdata
/dev/sda12	ntfs		PBR_DRV	13.25 GiB	8.99 GiB	4.26 GiB	hidden, diag

0 operations pending



```
ubuntu@ubuntu:~$ sudo gdisk -l /dev/sda
GPT fdisk (gdisk) version 0.8.8

Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with protective MBR; using GPT.
Disk /dev/sda: 1953525168 sectors, 931.5 GiB
Logical sector size: 512 bytes
Disk identifier (GUID): F7A07226-9A29-4596-A33C-F4A7A3147B6C
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 1953525134
Partitions will be aligned on 2048-sector boundaries
Total free space is 520834413 sectors (248.4 GiB)

Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048         2050047   1000.0 MiB  2700  Basic data partition
   2         2050048         2582527   260.0 MiB   EF00  EFI system partition
   3         2582528         4630527   1000.0 MiB  FFFF  Basic data partition
   4         4630528         4892671   128.0 MiB   0C01  Microsoft reserved part
   5         4892672       414492671   195.3 GiB   0700  Basic data partition
   6       414492672       721692671   146.5 GiB   0700  Basic data partition
   7       721692672       926492671   97.7 GiB    8300
   8       926492672      1131292671   97.7 GiB    8300
   9      1131292672      1336092671   97.7 GiB    8300
  10      1336092672      1352476671   7.8 GiB     8200
  11      1873307648      1925736447   25.0 GiB    0700  Basic data partition
  12      1925736448      1953523711   13.2 GiB    2700  Basic data partition
ubuntu@ubuntu:~$
```

We will need to mount this partition and then access the EFI information stored there. This can be done using efibootmgr, a small utility that can read and edit the information stored there.

sudo apt-get install efibootmgr

Mount the Ubuntu partition (ours is /dev/sda7) and the EFI partition:

sudo mkdir -p /mnt/system
sudo mount /dev/sda7 /mnt/system
sudo mount /dev/sda2 /mnt/system/boot/efi

Now, we need to play with efibootmgr. The program is not very difficult to use. And if you make mistakes, you can easily remedy them. Indeed, let's do a silly exercise, to help you understand how this bootloader manager works.

WITHOUT fully reading the documentation, we will execute a "repair" command, which you can easily find online and copy & paste with delight. Indeed, let's assume that you don't know if there's a valid Ubuntu entry, and you just want to add one. The syntax is then:

sudo efibootmgr -c -d /dev/sda -p 2 -w -L ubuntu

The switches and options state: -c create new entry on -d disk, on -p partition, -w write unique signature into MBR if needed, and use -L label to mark the new entry. You can use any label you want. The partition number must match your ESP.

sudo efibootmgr -c -d /dev/sda -p 1 -w -L ubuntu
** Warning ** : Boot0002 has same label ubuntu

```
BootCurrent: 0005
Timeout: 0 seconds
BootOrder: 0006,2001,0003,0002,2003,0004,2002
Boot0000* EFI Network 0 for IPv4 (68-F7-28-4B-D1-A1)
Boot0001* EFI Network 0 for IPv6 (68-F7-28-4B-D1-A1)
Boot0002* ubuntu
Boot0003* Windows Boot Manager
Boot0004* Lenovo Recovery System
Boot0005* EFI USB Device (KingstonDataTraveler G2)
Boot2001* EFI USB Device
Boot2002* EFI DVD/CDROM
Boot2003* EFI Network
Boot0006* ubuntu
```

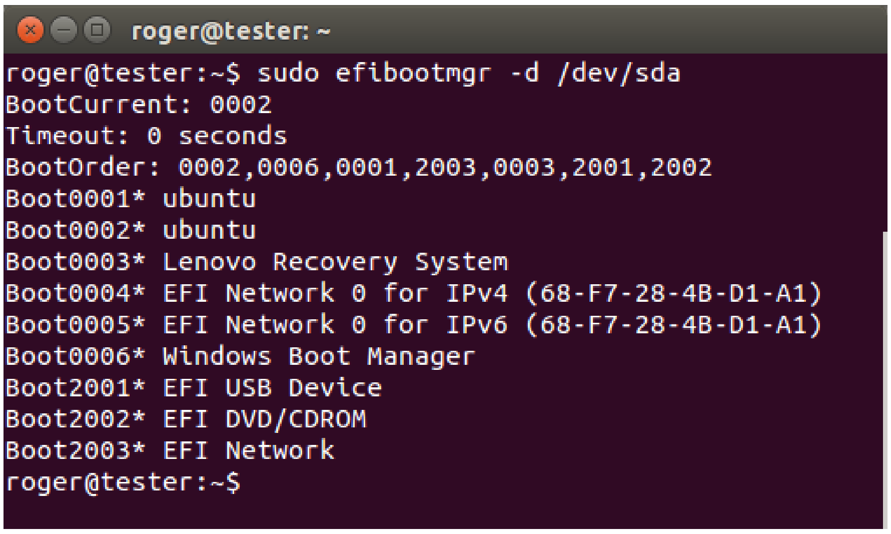
Let's examine the command output. As you can see, it is warning that there already is an entry for Ubuntu, it's just that Windows overrode the boot order. But no matter. Now, we can also see that we currently booted from a USB drive. The next boot order is going to be our new Ubuntu entry (0006), followed by USB devices if present, and then Windows. So we're good in that sense. Now, install GRUB-EFI, and we're ok.

You can continue playing and manipulating the EFI table if you want. For instance, you may want to delete extra entries if you like, to keep it clean. A crude example:

```
sudo efibootmgr -b 0008 -B
```

And you can also set active entry using the HEX notation. In fact, in our case, we only needed to mark the existing 0002 entry as active (-a), and that would have fixed the boot issue. But now, you understand what happened. It's as if someone changed the default stanza in the old GRUB menu. That's all.

Now if you don't want to make any modifications, you can list existing entries simply by using sudo efibootmgr -d /dev/sda. And this will print the existing boot order:



```
roger@tester:~$ sudo efibootmgr -d /dev/sda
BootCurrent: 0002
Timeout: 0 seconds
BootOrder: 0002,0006,0001,2003,0003,2001,2002
Boot0001* ubuntu
Boot0002* ubuntu
Boot0003* Lenovo Recovery System
Boot0004* EFI Network 0 for IPv4 (68-F7-28-4B-D1-A1)
Boot0005* EFI Network 0 for IPv6 (68-F7-28-4B-D1-A1)
Boot0006* Windows Boot Manager
Boot2001* EFI USB Device
Boot2002* EFI DVD/CDROM
Boot2003* EFI Network
roger@tester:~$
```

And modify accordingly, for instance, activate Ubuntu:

```
sudo efibootmgr -a 0002
```

This is the correct way of really doing it, and not writing a new entry. But I've found a bunch of online tutorials that blindly suggest the -c option. It's not dangerous, it just clutters the list. So be aware, and keep in mind everything is simple and fully reversible.

Solution 3: Manual method

If, for some reason, you cannot do the above, and you are not running Ubuntu, but a distro like Fedora perhaps, then you may want to try the manual recovery method, which is the same as what we did here, only more verbose.

```
(sudo) grub-install --boot-directory=/mnt/system/boot --bootloader-id=ubuntu  --target=x86_64-efi
--efi-directory=/mnt/system/boot/efi
```

What do we have here? Well, we're using the mounted boot directory, setting the correct bootloader entry ID (the same we did before), using the EFI target image, and writing into the EFI partition, which we have mounted earlier. That's all. You can also add debug to your output if you want, to see text flying.

```
...
grub-install: info: executing modprobe -q efivars.
grub-install: info: executing efibootmgr -b 0002 -B.
BootCurrent: 0005
Timeout: 0 seconds
BootOrder: 0006,0003
Boot0000* EFI Network 0 for IPv4 (68-F7-28-4B-D1-A1)
Boot0001* EFI Network 0 for IPv6 (68-F7-28-4B-D1-A1)
Boot0003* Windows Boot Manager
Boot0004* Lenovo Recovery System
Boot0005* EFI USB Device (KingstonDataTraveler G2)
Boot0006* ubuntu
Boot2001* EFI USB Device
Boot2002* EFI DVD/CDROM
Boot2003* EFI Network
...
```

Finally, we need to write the configuration into the grub.cfg file:

```
(sudo) grub-mkconfig -o /mnt/system/boot/efi/EFI/GRUB/grub.cfg
```

Solution 4: Alternative method (maybe)

If you also have Linux distributions installed in the legacy mode, like I did with Netrunner Prometheus, then you can also try this trick. Change the boot type in UEFI/BIOS to Legacy.
The jump instruction will now point to MBR, where you should have a bootloader for the legacy-supported systems installed. The boot menu could also include Ubuntu, depending on how you setup the system. But let's be practical.

In my example, Netrunner and Windows 8 were the only listed systems. I booted into Netrunner, updated the boot menu. On next reboot, Ubuntu was also available. Booting into it, you can now try the earlier step of installing the EFI-supported version of GRUB. This may not work, but you might want to play around, just for fun. Not on production systems, though.

Conclusion

There you go. This was a little lengthy, but I could not just give you silly commands to repeat without understanding what's happening. Knowledge is power, and the more you have, the less you will fear errors and mistakes when setting up complex multi-boot systems. Here, you get as much power as possible. Four methods.

You should work through the solutions slowly, one by one. Do not worry too much if all that's happened was an innocent Windows install. But you can consider backing up your data from a live session, and making sure you do not delete or overwrite anything in a bout of panic. There's no reason to reinstall or such. Just follow this tutorial and you'll be mighty fine. Another GRUB obstacle covered.

P.S. If you find this article useful, please support Dedoimedo.

Cheers.