

For me, booting Ubuntu in UEFI mode with the Nvidia drivers loaded, also always resulted in the well-known black screen when X was started.

That was until yesterday!

After running into some very promising info in [another thread](#) about installing Windows 7 in UEFI mode on a Mac. Folks over there struggled with Windows 7's required int 10h legacy support and found out that in order to work around that, one can perform an unattended installation (because display doesn't work during install). The **crucial information to successfully boot Ubuntu in UEFI mode even with Nvidia drivers** was that upon start of EFI boot (while handing over to grub), **Apple's firmware does not actually activate the VGA card as PCI-E bus master**. The Windows' guys explained how to circumvent this using an EFI shell which chainloads the Windows boot manager in order to at least run the setup in unattended mode. And here is the good news: **it's easy to do this in GRUB!**

In fact, I am right now typing this on a MacBook Pro 7,1 (mid-2010) running Ubuntu Vivid booted in UEFI mode ([Xorg.0.log](#)). However, it should be easy to run on or adapt this to any Linux distribution providing an EFI version of GRUB, e.g. Ubuntu 14.04.2 LTS.

```
~$ dmesg | grep -i efi
[ 0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-3.19.0-15-
generic.efi.signed root=UUID=7843c644-e6f4-4d64-9317-0b854cb524f2 ro quiet
splash intremap=off acpi_osi=! acpi_osi=Darwin nomodeset vt.handoff=7
[ 0.000000] efi: EFI v1.10 by Apple
[ 0.000000] efi: ACPI=0xbf96a000 ACPI 2.0=0xbf96a014 SMBIOS=0xbf71a000
```

To cut the long story short, we have to set two PCI-E registers: one that enables bus-mastering on the video card and the other one enabling VGA support on the PCI-E bridge of the video card. So it has actually nothing to do with the Nvidia drivers and depending on the viewpoint, not even with Apple's outdated/crippled/you name it EFI implementation.

This is how I solved it. Of course, the kudo's go primarily to the guys on the MacRumors forum. **DISCLAIMER** The following instructions are provided as is, without guarantees nor do I assume any liability. **DO THIS AT YOUR OWN RISK!**

1. Install Ubuntu in UEFI mode with the Nvidia drivers

I will not go into details here because there are lots of articles on the web showing you how to do this and end up with the nouveau drivers running your machine hot or with the black screen after installing the Nvidia drivers (e.g. www.rodsbooks.com/ubuntu-efi).

Please note however that the following instructions assume you have successfully installed and booted in UEFI mode. You can verify that by checking for existence of the directory `/sys/firmware` which is only created after booting in EFI mode.

2. Find the right PCI-E bus identifiers

We need both the IDs for the graphics card and the PCI-E bridge that it is connected to. Issue the following command in a shell:

```
~$ sudo lshw -businfo -class bridge -class display
pci@0000:00:00.0          bridge          MCP89 HOST Bridge
pci@0000:00:03.0          bridge          MCP89 LPC Bridge
pci@0000:00:0e.0          bridge          NVIDIA Corporation
pci@0000:00:15.0          bridge          NVIDIA Corporation
```

```
pci@0000:00:16.0      bridge      NVIDIA Corporation
pci@0000:00:17.0      >!!<      bridge      MCP89 PCI Express Bridge
pci@0000:04:00.0      >!!<      display     MCP89 GeForce 320M
```

Have a look at (1) the line saying display and (2) the line with bridge right before that display line. Write down the PCI-E bus ids (format XX:YY.Z) of the bridge device (here 00:17.0) and the display device (here 04:00.0) and remember which is which. **Note:** Those IDs may be different on your machine, depending on your Mac model and revision.

3. Create a GRUB script for setting the PCI-E registers during boot

Fire up a text editor with `sudo nano /etc/grub.d/01_enable_vga.conf` and copy/paste the content below. **Replace** `00:17.0` with the PCI-E ID of your bridge device noted in step 2. **Replace** `04:00.0` with the PCI-E ID of your display device noted in step 2.

```
cat << EOF
setpci -s "00:17.0" 3e.b=8
setpci -s "04:00.0" 04.b=7
EOF
```

Finally, make the created file executable and update your grub config files using the following **TWO** commands.

```
~$ sudo chmod 755 /etc/grub.d/01_enable_vga.conf
~$ sudo update-grub
```

4. Reboot and check

If, after rebooting, the register values have been set to 8 (bridge device) and 7 (display device), everything went fine:

```
~$ sudo setpci -s "00:17.0" 3e.b
08
~$ sudo setpci -s "04:00.0" 04.b
07
```

5. Install Nvidia drivers and enjoy!

Use Ubuntu's *Additional drivers* GUI to install the Nvidia drivers. I recommend following [this article](#) on how to enable brightness controls because it doesn't work out of the box.

edited Apr 13 '17 at 12:23



Community ♦

1

answered Apr 24 '15 at 21:17



Andreas

396 2 5