Hackintosh on a Dell T7500

Dr Ahmed Riza

1   Requirements

The machine I have is a Dell T7500 with an Nvidia Quadro 4000 GPU. The extra items I have added to the machine since was a second Nvidia graphics card (an old GeForce 7900GS) and a SAS 6Gb/s PCI card.

Firstly, I had to remove the GeForce 7900GS, since OS X would not boot with the two cards present.

2   Preparation

I followed the excellent article at the following URL: `http://lifehacker.com/5841604/the-always-up+to+date-guide-to-building-a-hackintosh`. I followed this guide, until the end of Step 2.

3   Specific Hardware

The specific hardware on the T7500 that did not work out of the box are given below, and I'm going to address how I got them working in this article.

1. Nvidia Quadro 4000

2. Broadcom 5754 On-board Ethernet

3. Enabling TRIM on the SSD drive

Note that I do not have sound working yet, and I'll update this article once I get that working.

4   Install OS X

After booting from the USB stick prepared as described in the article referenced in Section 2, I installed OS X on a new Intel SSD 480 GB drive. The installation went fine without any issues apart from my on-board Ethernet card not being recognized. I just asked the installer to ignore Ethernet and the rest went smoothly.

However, after rebooting, I ran into trouble straight away. There are a lot of forum posts from users who have had this experience, i.e. the machine starts booting with the Apple logo up and a spinning wheel. After a while, all you will get is a black screen!

I researched all the forums I possibly could, and thought long and hard about what could be causing this, since this card is definitely supposed to work, and there are people who have it working on their machines. So let's fix this.

5   Nvidia Quadro 4000

Confusingly, there are two varieties of this card, i.e.

1. Nvidia Quadro 4000

2. Nvidia Quadro 4000 for Mac

I have the first kind, i.e. a non-Mac version. I actually do not know the difference between the two cards. I believe it has something to do with the on-board ROM in the two boards. There are people on the net who falsely claim that the non-Mac version would not work with OS X. This is clearly wrong.

Anyway, the single source of information that finally lit the lightbulb in my head is the following, and I have to thank the author immensely for spending time to explain as he did. http://olarila.com/forum/viewtopic.php?f=18&t=154. Here are the steps I took to get the GPU working.

1. Boot into OS X from the USB stick in safe mode. That is, pass the "-x" parameter as a boot flag on the boot screen.

2. Install the Quadro 4000 driver from the Nvidia website. Note that you will have to select the Quadro 4000 Mac card in order to download the driver. This is fine, since the driver will work with the non-Mac card, after a bit of modification as described below. When the installation finishes, the installer will prompt you to reboot the machine. Do not reboot yet, until we fix the driver.

   On a side note, you may be wondering how I was able to install the driver from the Nvidia website, given that I do not have Ethernet working on this machine. I simply downloaded it from another machine, and transferred via a USB stick.

   I already knew the Vendor/Device ID of this card, by inspecting the hardware details from Windows. In any case, the details are as follows:

   - Vendor ID = 0x10de
   - Device ID = 0x06dd

   We need to modify the Info.plist of the graphics driver with these values. So, open up the Info.plist of the driver using your favorite editor. Please note that the file is owned by root, so you'll need to *sudo* to modify the file. I assume you know all about that. I used vi to edit the file:

   ```
   sudo vi /System/Library/Extensions/NVDAGF100Hal.kext/Contents/Info.plist
   ```

   Look for the key **IOPCIMatch**, and modify the string as follows:

   ```
   <key>IOPCIMatch</key>
   <string>0x000010de&amp;0x000006dd</string>
   ```

   Save the file and "touch" the Extensions folder:

   ```
   sudo touch /System/Library/Extensions
   ```

3. Now, we need to figure out the right value to pass to the **PCIRootUID** boot flag. Open a terminal and type the following command:

   ```
   ioreg -l | grep -15 "AppleACPIPCI\ " | grep UID
   ```

   On my machine, I got two values, i.e.

   ```
   UID = 4
   UID = 38
   ```

   After a bit of trial and error, what worked for me was the first value, i.e. *PCIRootUID = 4*.

We are now ready to reboot, so reboot the machine. I am still booting from the USB stick, so pass the following boot flags:

```
PCIRootUID=4
```

The actual value you pass here depends on what was discovered in the step above. That should get the Quadro 4000 working and finally, have a working machine with a nice GPU!
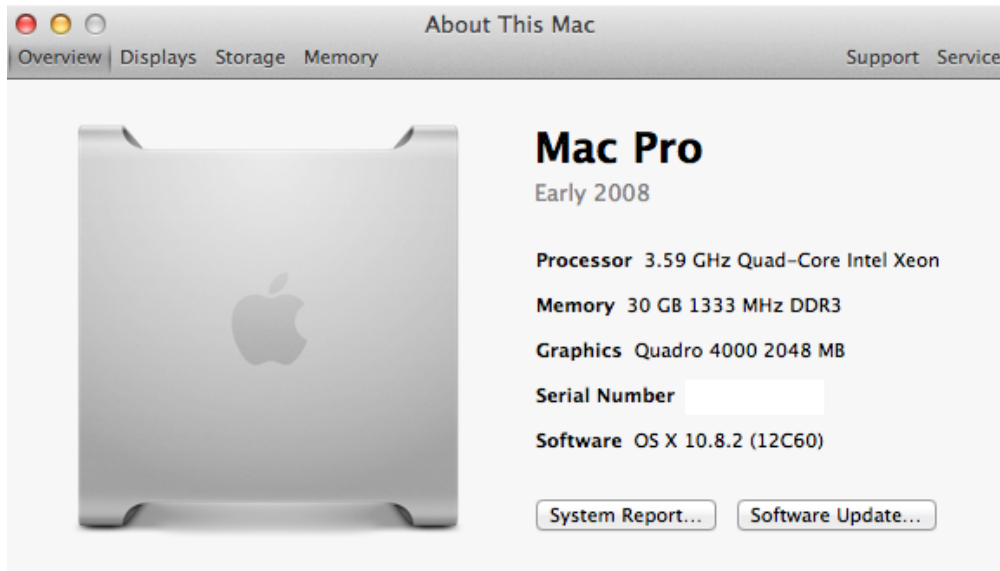
6   Broadcom 5754 On-board Ethernet

This was the next challenge. Some Googling revealed that a very nice person has actually written a driver for OS X.

1. You can either download the precompiled driver from the following site: `http://www.insanelymac.com/forum/topic/247470-bcm5722-bcm5754m-bcm5755m-bcm5787m-and-bcm5906m-nic-driver-3264-bit/`

2. Or compile it yourself with Xcode by downloading the source from git: `https://github.com/adlan/BCM5722D/`

Now, the driver may work straight out of the box for you, but chances are that it won't as was the case with me. So, I booted into Windows, and found out the Device ID of the Ethernet card. My card details were:

1. Vendor ID: 14e4

2. Device ID: 1681

It is the Device ID, you'll have to enter into the Info.plist of the driver. I was also a bit confused about where to install the driver. I could not find this mentioned in the driver documentation or source code. Some Googling revealed that it has to be installed under the **/System/Library/Extensions/IONetworkingFamily.kext/Contents/PlugIns/** directory.

So, I edited the Info.plist as follows:

```
sudo vi /System/Library/Extensions/IONetworkingFamily.kext/Contents/PlugIns/BCM5722D.kext/Contents/In
```

I added an extra line to the dictionary containing the Device IDs:

```
<array>
        <string>pci14e4,165a</string>
        <string>pci14e4,167a</string>
        <string>pci14e4,1672</string>
        <string>pci14e4,167b</string>
        <string>pci14e4,1673</string>
        <string>pci14e4,1681</string>
```

```
                    <string>pci14e4,169b</string>
                    <string>pci14e4,1693</string>
                    <string>pci14e4,1712</string>
                    <string>pci14e4,1713</string>
            </array>
```

The entry with **pci14e4,1681** is what I added. After this, "touch" the Extensions directory as usual, reboot and Ethernet worked flawlessly. Once again, many thanks to the author of the driver.

7   Enabling TRIM on the SSD Drive

This was the easiest job of all. Once again, a very nice person has detailed the steps precisely and all I did was follow those steps to the letter. The following page helped in that regard: `http://digitaldj.net/2011/07/21/ trim-enabler-for-lion`. I will detail the steps from this article, here for reference:

1. Backup the files wet are patching.

   ```
   sudo cp /System/Library/Extensions/IOAHCIFamily.kext/Contents/PlugIns/IOAHCIBlockStorage.kext/
   Contents/MacOS/IOAHCIBlockStorage /System/Library/Extensions/IOAHCIFamily.kext/Contents/PlugIns/
   IOAHCIBlockStorage.kext/Contents/MacOS/IOAHCIBlockStorage.original
   ```

2. Patch the file to enable TRIM support.

   ```
   FOR ML 10.8.1 AND LION 10.7.5 OR NEWER

   sudo perl -pi -e 's|(\x52\x6F\x74\x61\x74\x69\x6F\x6E\x61\x6C\x00{1,20})
   [^\x00]{9}(\x00{1,20}\x4D)|$1\x00\x00\x00\x00\x00\x00\x00\x00$2|sg'
   /System/Library/Extensions/IOAHCIFamily.kext/Contents/PlugIns/IOAHCIBlockStorage.kext/Contents/MacOS/IOAHCIBlockStorage

   FOR ML 10.8.0 AND LION 10.7.4 BELOW

   sudo perl -pi -e 's|(\x52\x6F\x74\x61\x74\x69\x6F\x6E\x61\x6C\x00{1,20})
   [^\x00]{9}(\x00{1,20}\x51)|$1\x00\x00\x00\x00\x00\x00\x00\x00$2|sg'
   /System/Library/Extensions/IOAHCIFamily.kext/Contents/PlugIns/IOAHCIBlockStorage.kext/Contents/MacOS/IOAHCIBlockStorage
   ```

3. Force a refresh of the system's kernel extension cache

   ```
   sudo touch /System/Library/Extensions/
   ```

Now, reboot and TRIM should be working, and you can check this from the hardware information about your drive. There should be an entry for TRIM. As ever, a picture is worth a thousand words (the drive information should show "TRIM Support : Yes"):

Mac Pro

Hardware
- ATA
- Audio
- Bluetooth
- Card Reader
- Diagnostics
- Disc Burning
- Ethernet Cards
- Fibre Channel
- FireWire
- Graphics/Displays
- Hardware RAID
- Memory
- PCI Cards
- Parallel SCSI
- Power
- Printers
- SAS
- **Serial-ATA**
- Thunderbolt
- USB

Network
- Firewall
- Locations
- Modems
- Volumes
- WWAN
- Wi-Fi

Software
- Accessibility
- Applications
- Components
- Developer
- Extensions
- Fonts
- Frameworks
- Installations
- Logs
- Managed Client
- Preference Panes
- Printer Software
- Profiles
- Startup Items
- Sync Services

Serial-ATA Device Tree

▼ Intel ICH10 AHCI
    TSSTcorpDVD-ROM TS-H353C
▼ Intel ICH10 AHCI
    PLDS DVD+/-RW DH-16ABS
▼ Intel ICH10 AHCI
    INTEL SSDSA2CW160G3
Intel ICH10 AHCI
Marvell 88SE9123
▼ Marvell 88SE9123
    INTEL SSDSC2CW480A3
Marvell 88SE9123
Marvell 88SE9123
Marvell 88SE9123
Marvell 88SE9123
▼ Marvell 88SE9123
    MARVELL VIRTUALL

INTEL SSDSC2CW480A3:

Capacity:	480.1 GB (480,103,981,056 bytes)
Model:	INTEL SSDSC2CW480A3
Revision:	400i
Serial Number:	CVCV207204CB480DGN
Native Command Queuing:	Yes
Queue Depth:	32
Removable Media:	No
Detachable Drive:	No
BSD Name:	disk1
Medium Type:	Solid State
TRIM Support:	Yes
Partition Map Type:	GPT (GUID Partition Table)
S.M.A.R.T. status:	Verified
Volumes:	

  disk1s1:

Capacity:	209.7 MB (209,715,200 bytes)
BSD Name:	disk1s1
Content:	EFI

  OS X:

Capacity:	479.76 GB (479,760,007,168 bytes)
Available:	420.19 GB (420,190,314,496 bytes)
Writable:	Yes
File System:	Journaled HFS+
BSD Name:	disk1s2
Mount Point:	/
Content:	Apple_HFS
Volume UUID:	22DF36BA-42DC-3801-A6E0-0C2953520C8D