Overview

The purpose of this guide is to provide a step-by-step guide to installing El Capitan/Yosemite on the ProBook/EliteBook/Zbook equipped with Intel graphics.

Note: For Broadwell and Skylake hardware, you must install 10.11. Broadwell support in 10.10 is not stable (previous versions no Broadwell support). And Skylake support was only introduced in 10.11 (.?), and only stable enough for use in 10.11.4.

Note: This guide will also work with Mavericks (for laptops with supporting hardware), although that is not my primary testing target. In fact, I'm using the same setup on my 4530s with OS X as old as 10.6.8. Of course, many of the latest kexts are not compatible with the older versions of OS X and the WiFi card supports is not the same as the later versions. You may have to dig up old kexts or use different WiFi hardware.

I have both a ProBook 4530s, and ProBook 4540s and have been using the hotpatch techique that this guide enables for quite a while. The files and procedures here have been tested on other models, although you should still consider this guide beta.

Since the ProBook Installer is not being maintained anymore (and neither is the guide), this is meant as a replacement for the guide by nguyenmac: http://www.tonymacx86.com/el-capita...-hp-elitebook-probook-others-clover-uefi.html

What you need

Compatible laptop:
- 6-series laptop (Sandy Bridge): HP Probook 4x30s, 6x60b, Elitebook 2x60p, 8x60p, 8x60w
- 7-series laptop (Ivy Bridge or Sandy Bridge): HP Probook 4x40s, 4x0 G0, 6x70b, Elitebook 8x70p, 2x70p, 9470m
- 8-series laptop (Haswell): HP 3x0 G1, Probook 4x0 G1/G2, Elitebook 8x0 G1, ZBook G1, Folio 1040 G1
- 9-series laptop (Broadwell): HP Probook 4x0/6x0/8x0/ZBook G2
- 100-series laptop (Skylake): HP ProBook 4x0 G3 is experimental (probably incomplete), but reported as working.
- 100-series laptop (Kabylake): HP ProBook 4x0 G4 is experimental (probably incomplete), supported with the 'beta' branch and is currently reported as working.

- OS X Yosemite or El Capitan downloaded from the Mac App Store
- 8GB USB stick
- compatible WiFi hardware
- BIOS updated to latest as available from hp.com

At the moment, laptops without Intel graphics are not supported. Although many of the procedures and techniques here may be applicable, these laptops require a different graphics configuration (config.plist changes, other kext choices, slightly different ACPI setup).

Note: This guide supports only those laptops with Synaptics PS2 trackpads. Other trackpads will be mouse emulation only and you will need to exclude/delete the SSDT-KEY*.aml files from ACPI/patched.

Compatible WiFi

Here is the list of recommended compatible wifi cards:
- HP Probook 4x30s: the laptop has whitelist BIOS, so only usable card is HP-branded Atheros

AR5B195 (HP part# 593127-001). You can also use a rebranded AR5BHB92 (AR9280).
- HP Probook 6x60b/5330m, HP Elitebook 8x60p/2x60p: the laptop has whitelist BIOS, so only usable card is HP-branded Broadcom 43224HMS (582564-001/582564-002)
- For laptops with NGFF/M.2 WiFi slot: BCM94352Z or BCM94350ZAE (? not tested) or BCM943602BAED (? not tested)
- Other laptop:
WIFI ONLY: Atheros AR9280/9285, Broadcom 4322/43224 (better OS X feature support)
WIFI+BT: AR5B195 (not recommended), Broadcom 43225HMB/4352HMB (better OS X feature support)

Note: Some laptops covered by this guide have a built-in bluetooth controller and do not use the one present on the WiFi card. For those laptops you do not need a WiFi/bluetooth combination card since the bluetooth component on it will not be connected and you can use the onboard bluetooth controller instead (usually Broadcom). It is up to you to investigate properly to discover how your bluetooth hardware is configured.

Further information regarding OS X compatible WiFi is in the FAQ: http://www.tonymacx86.com/el-capita...faq-read-first-laptop-frequent-questions.html

The patch provided for 5ghz (BCM94352HMB) is specific to US/FCC (North America). If you reside in another part of the world, the patch (by @the-darkvoid) must be customized for your country code. Read here: http://www.tonymacx86.com/network/104850-guide-airport-pcie-half-mini-v2.html

For those laptops with half-mini PCIe, BCM94360HMB is also a good option, but you will likely need to route new/additional antennas as this card uses (3) smaller coax connectors than most half-mini PCIe WiFi cards. The advantage is that it is recognized natively and no 5ghz patch is needed (AirPort branding still provided by FakePCIID).

Note: USB WiFi is not recommended. The user interface is poor, and it causes instability, especially with sleep.

BIOS settings

To start, set BIOS to defaults.

Then insure:
- UEFI boot is enabled (hybrid/with CSM for best result)
- secure boot is disabled
- disable fast boot
- IGPU graphics memory set to 64mb (Broadwell and Skylake)
- disable the serial port via BIOS option if available
- disable "LAN/WLAN switching" if available
- disable "Extended Idle Power States" if you find it under "Power Management Options"
- disable "Wake on LAN" and "Wake on USB"
- disable Firewire/IEEE 1394 if your laptop has it

Note: If you have a laptop with switchable graphics, leave it enabled. You can still use it on Windows, although the discrete card will be disabled when running OS X by the ACPI patches provided here. If you want to also disable it in Windows (via BIOS option), make sure you read about the DGPU option in SSDT-HACK.dsl, as in the scenario where the DGPU is disabled by BIOS, DGPU should be set to zero.

Preparing USB and initial Installation

Prior to installing OS X, it is a good idea to create an OEM recovery USB from Windows. If anything goes wrong and you want/need to get back to Windows, you can restore it via the USB. Use the utility provided by HP to accomplish this.

You can also leave Windows intact, but it can get tricky. Read here for more information: http://www.tonymacx86.com/multi-booting/133940-mavericks-windows-8-same-drive-without-erasing.html

This guide for creating USB and installing using Clover UEFI works well for this laptop: http://www.tonymacx86.com/el-capita...de-booting-os-x-installer-laptops-clover.html

Special notes:

- Definitely copy RealtekRTL8111.kext and IntelMausiEthernet.kext to Clover/kexts/Other as having network support during post-install is required (the rest of this guide depends on it). An alternative is to copy FakePCIID.kext, FakePCIID_Broadcom_WiFi.kext, and ProBookAtheros.kext to Clover/kexts/Other. This will enable WiFi, provided you have compatible WiFi hardware already installed.

Note: FakePCIID.kext and FakePCIID_Broadcom_WiFi.kext are available here: https://github.com/RehabMan/OS-X-Fake-PCI-ID. ProBookAtheros.kext is available in the 4x30s DSDT patch github: https://github.com/RehabMan/HP-ProBook-4x30s-DSDT-Patch (download the ZIP of the entire github project).

- GenericUSBXHCI.kext is not necessary with most laptops covered by this guide since AppleUSBXCHI.kext appears to work well natively. The exception is those laptops with 6-series chipsets. For these laptops you should use a USB2 port and USB2 flash drive. There is no support in the BIOS for booting from the NEC/Renesas ports anyway.

- Also copy HPFanReset.efi to EFI/Clover/drivers64UEFI. HPFanReset.efi is available here: https://bitbucket.org/RehabMan/hp-probook-4x30s-fan-reset/downloads

- Use the 'createinstallmedia' approach. It works well, and there is little chance for pilot error. This method also gives you an OS X recovery partition.

- It is best to use a USB2 stick. USB3 on 6-series ProBook/EliteBook will have issues.

- Some laptops will need to use the alternate DropTables as mentioned in the guide. Likely required for Sandy Bridge models with turbo boost enabled CPUs (eg. Core i5-2xxx, Core i7-2xxx, but not Core i3-2xxx).

Post Installation

Install Clover UEFI as described in the guide linked by the previous section (post #2). After installing Clover, and configuring it correctly (config.plist, kexts, etc, just as you did for USB) you should be able to boot from the HDD/SSD. The configuration at this point should be exactly the same as USB. Don't forget both HFSPlus.efi and HPFanReset.efi.

But there are still many issues and devices that won't work correctly. For that, we need to patch DSDT, provide a proper config.plist, and install the kexts that are required.

Since you have RealtekRTL8111.kext/IntelMausiEthernet.kext already injected by Clover, you should have internet access simply by using an Ethernet cable to your router. Plug it in and make sure you have internet access before continuing. Or if you're using WiFi, you can connect to your WiFi router before continuing.

Installation of the tools and patching is easy provided the scripts and tools at the ProBook repository: https://github.com/RehabMan/HP-ProBook-4x30s-DSDT-Patch

To start, the developer tools must be installed. Run Terminal, and type:

Code (Text):

```
git
```

You will be prompted to install the developer tools. Since you have internet working, you can choose to have it download and install them automatically. Do that before continuing.

After the developer tools are installed, we need to get a couple of projects from my github.

In Terminal:

Code (Text):

```
mkdir ~/Projects
cd ~/Projects
git clone https://github.com/RehabMan/HP-ProBook-4x30s-DSDT-Patch probook.git
```

The beta branch may have updated (but untested) code. Experimental code for new laptops, etc, may only be available in the beta branch. You can use the beta branch by switching to it after you clone the project.

In Terminal:

Code (Text):

```
cd ~/Projects/probook.git
git checkout beta
```

Now it is time to install some more tools and all the kexts that are required...

In Terminal:

Code (Text):

```
cd ~/Projects/probook.git
./download.sh
./install_downloads.sh
```

The download.sh script will automatically gather the latest version of all tools (patchmatic, iasl, MaciASL) and all the kexts (FakeSMC.kext, IntelBacklight.kext, ACPIBatteryManager.kext, etc) from bitbucket. The install_downloads.sh will automatically install them to the proper locations.

If you wish, you can reboot to verify a few more items are working (many only partially). Booting may be a bit slower at this point because of the kexts installed but with incorrect ACPI setup.

To finish the setup, we need a correctly patched DSDT/SSDT.

With the current project, no patched DSDT/SSDTs are used. Instead, this guide uses Clover hotpatches and a set of "add-on" SSDTs. The advantage of hotpatching is that hardware and BIOS can be changed without re-extract/re-patch. It is also a bit easier to setup as the normal extract/disassemble/patch/compile process is not needed (that process was formerly automated via the ProBook Installer and patchmatic).

In Terminal:

Code (Text):

```
cd ~/Projects/probook.git
./build.sh
```

The build.sh script causes the "add-on" SSDT files to be compiled (with iasl), the results placed in ./build. It also builds the system specific plist files to be used later for Clover's config.plist.

You can see what is there:

Code (Text):

```
cd ~/Projects/probook.git
ls build
```

As I write this, current content:

Code (Text):

```
SSDT-1020-G1-Broadwell.aml    SSDT-9x70.aml            SSDT-USB-1020-G1.aml
SSDT-1040-G1-Haswell.aml    SSDT-BATC.aml            SSDT-USB-4x0-G2.aml
SSDT-2x70.aml            SSDT-BATT-G2.aml        SSDT-USB-4x0-G3.aml
SSDT-3x0-G1.aml            SSDT-BATT-G3.aml        SSDT-USB-4x30s.aml
SSDT-4x0-G0.aml            SSDT-BATT.aml            SSDT-USB-4x40s.aml
SSDT-4x0-G1-Haswell.aml        SSDT-EH01.aml            SSDT-USB-6x0-G1.aml
SSDT-4x0-G1-Ivy.aml        SSDT-EH02.aml            SSDT-USB-6x60.aml
SSDT-4x0-G2-Broadwell.aml    SSDT-FAN-MOD.aml        SSDT-USB-6x70.aml
SSDT-4x0-G2-Haswell.aml        SSDT-FAN-ORIG.aml        SSDT-USB-820-G2.aml
SSDT-4x0-G3-Skylake.aml        SSDT-FAN-QUIET.aml        SSDT-USB-840-G2.aml
SSDT-4x30s.aml            SSDT-FAN-READ.aml        SSDT-USB-850-G2.aml
SSDT-4x40s.aml            SSDT-FAN-SMOOTH.aml        SSDT-USB-8x0-G1.aml
SSDT-5x30.aml            SSDT-HACK.aml            SSDT-USB-8x0-G3.aml
SSDT-6x0-G1-Haswell.aml        SSDT-IGPU-HIRES.aml        SSDT-USB-8x60.aml
SSDT-6x60.aml            SSDT-IGPU.aml            SSDT-USB-9x70.aml
SSDT-6x70.aml            SSDT-KEY102.aml            SSDT-USB-ZBook-G1.aml
SSDT-8x0-G1-Haswell.aml        SSDT-KEY87.aml            SSDT-XHC.aml
SSDT-8x0-G1-Ivy.aml        SSDT-LANC_PRW.aml        SSDT-XOSI.aml
SSDT-8x0-G2-Broadwell.aml    SSDT-LPC.aml            SSDT-ZBook-G2-Broadwell.aml
SSDT-8x0-G2-Haswell.aml        SSDT-PNLF.aml            SSDT-ZBook-G2-Haswell.aml
SSDT-8x0-G3-Skylake.aml        SSDT-PRW.aml            readme.txt
SSDT-8x60.aml            SSDT-SATA.aml
SSDT-8x70.aml            SSDT-SMBUS.aml
```

```
SSDT-8x70.aml                SSDT-SMBUS.aml
```

The files that need to be installed depend on personal preferences and your laptop model. The (appropriate set of) SSDTs in ./build must be installed to the EFI partiton at EFI/Clover/ACPI/patched.

To mount your EFI partition, use the mount_efi.sh shell script from the project:

Code (Text):

```
cd ~/Projects/probook.git
sudo ./mount_efi.sh /
```

All models will use the following SSDTs:

Code (Text):

```
SSDT-HACK.aml
SSDT-XOSI.aml
SSDT-EH01.aml
SSDT-EH02.aml
SSDT-XHC.aml
SSDT-LPC.aml
SSDT-SATA.aml
SSDT-SMBUS.aml
SSDT-PNLF.aml
SSDT-PRW.aml
SSDT-LANC_PRW.aml
SSDT-BATC.aml
```

Note: This list of files may change in the github without a change right away in the guide text above. You should always use install_acpi.sh to install the correct set of files (see below).

Then depending on model...

One of the model specific SSDTs such as SSDT-4x30s.aml or SSDT-4x40s.aml, etc.
One of SSDT-IGPU.aml or SSDT-IGPU-HIRES.aml
One of SSDT-BATT.aml, SSDT-BATT-G2.aml, or SSDT-BATT-G3.aml.
One of SSDT-KEY102.aml or SSDT-KEY97.aml
One of the USB customizations SSDT-USB-*.aml

And depending on your preference for fan control...

One of SSDT-FAN-READ.aml, SSDT-FAN-SMOOTH.aml, SSDT-FAN-ORIG.aml, or SSDT-FAN-QUIET.aml.

If there is not a USB customization SSDT, do not use one for another model. One can be created if you provide the materials listed in "USB customization".

If there is not a system specific SSDT, you can try to use one that is close, but it is better if you provide the materials requested in "Unsupported laptops", so a correct SSDT can be created.

For example, I use these SSDTs with my ProBook 4530s with 1366x768 display:

Code (Text):

```
SSDT-HACK.aml
SSDT-XOSI.aml
SSDT-EH01.aml
SSDT-EH02.aml
SSDT-XHC.aml
SSDT-LPC.aml
SSDT-SATA.aml
SSDT-SMBUS.aml
SSDT-PNLF.aml
SSDT-PRW.aml
SSDT-LANC_PRW.aml
SSDT-BATC.aml
SSDT-4x30s.aml
SSDT-IGPU.aml
SSDT-BATT.aml
SSDT-KEY102.aml
SSDT-USB-4x30s.aml
SSDT-FAN-MOD.aml
```

On my ProBook 4540s with 1080p display, I use:

Code (Text):

```
SSDT-HACK.aml
SSDT-XOSI.aml
SSDT-EH01.aml
SSDT-EH02.aml
SSDT-XHC.aml
SSDT-LPC.aml
SSDT-SATA.aml
SSDT-SMBUS.aml
SSDT-PNLF.aml
SSDT-PRW.aml
SSDT-LANC_PRW.aml
SSDT-BATC.aml
SSDT-IGPU-HIRES.aml
SSDT-BATT.aml
SSDT-KEY102.aml
SSDT-USB-4x40s.aml
SSDT-FAN-MOD.aml
```

And as a newer example, the Skylake ProBook 4x0 G3:

Code (Text):

```
SSDT-HACK.aml
SSDT-XOSI.aml
SSDT-EH01.aml
SSDT-EH02.aml
SSDT-XHC.aml
SSDT-LPC.aml
SSDT-SATA.aml
SSDT-SMBUS.aml
```

```
SSDT-PNLF.aml
SSDT-PRW.aml
SSDT-LANC_PRW.aml
SSDT-BATC.aml
SSDT-4x0-G3-Skylake.aml
SSDT-IGPU.aml
SSDT-BATT-G3.aml
SSDT-KEY102.aml
SSDT-USB-4x0-G3.aml
SSDT-FAN-MOD.aml
```

Choose the set of SSDTs that match your hardware and your personal choices (as far as fan control).

You could copy them with Finder or you can use Terminal.

There are a number of install scripts provided in install_acpi.sh to make it easier to accomplish.

For example, my 4530s with 1366x768 display:

Code (Text):

```
./install_acpi.sh install_4x30s
```

Or my 4540s with 1080p display:

Code (Text):

```
./install_acpi.sh install_4x40s_hires
```

You can list the current install scripts available with:

Code (Text):

```
./install_acpi.sh help
```

All of the scripts default to copying SSDT-FAN-READ.aml, which provides BIOS fan control and readings only. If you want a different fan patch, you can specify it as the second argument to install_acpi.sh. The possible values are READ, QUIET, SMOOTH, and MOD.

On my 4540s, I like to use the MOD variant:

Code (Text):

```
./install_acpi.sh install_4x40s_hires MOD
```

If your model does not have an install script it is because the model is not directly supported. See "Unsupported laptops" later in this guide.

Power Management

Use the ssdtPRGen.sh script by Pike R. Alpha: https://github.com/Piker-Alpha/ssdtPRGen.sh

Instructions for using it are provided in the README.

The generated SSDT.aml should be placed on the EFI partition at /EFI/Clover/ACPI/patched/SSDT.aml.

Based on the current instructions at the README, what follows is a synopsis...

To prepare, you only need to do the first two commands:

Code (Text):

```
cd ~/Projects/probook.git
curl --fail -o ./ssdtPRGen.sh https://raw.githubusercontent.com/Piker-Alpha/ssdtPRGen.sh/m
chmod +x ./ssdtPRGen.sh
```

With newer computers (Broadwell, Skylake, and future), make sure you use the beta branch:

Code (Text):

```
cd ~
curl --fail -o ./ssdtPRGen.sh https://raw.githubusercontent.com/Piker-Alpha/ssdtPRGen.sh/b


chmod +x ./ssdtPRGen.sh
```

Then you can run the script:

Code (Text):

```
./ssdtPRGen.sh
```

When it asks if you want to copy to /Extra just respond 'n'. Same for opening ssdt.dsl... respond 'n'.

The results are at ~/Library/ssdtPRGen/SSDT.aml.

Note: For certain Ivy Bridge CPUs (if you find your pstates stuck at x8), you may need to use the "-w 3" flag:

Code (Text):

```
./ssdtPRGen.sh -w 3
```

Copy that file to EFI partition, /EFI/Clover/ACPI/patched/SSDT.aml

Code (Text):

```
cd ~/Projects/probook.git
sudo ./mount_efi.sh /
cp ~/Library/ssdtPRGen/ssdt.aml /Volumes/EFI/EFI/Clover/ACPI/patched/SSDT.aml
```

Also, be aware that hibernation (suspend to disk or S4 sleep) is not supported on hackintosh.

You should disable it:

Code (Text):

```
sudo pmset -a hibernatemode 0
sudo rm /var/vm/sleepimage
sudo mkdir /var/vm/sleepimage
```

Always check your hibernatemode after updates and disable it. System updates tend to re-enable it, although the trick above (making sleepimage a directory) tends to help.

Note: It is common with certain Ivy computers with ULV CPUs and MacBookAir5,x SMBIOS to need kernel flag -xcpm. If you get a panic upon boot, add the -xcpm kernel flag to config.plist/Boot/Arguments.

Final config.plist

Up to now, you've been using the same config.plist we were using for installation. After all the ACPI files are in place (previous two steps), you're ready to use the final config.plist from the probook repo.

First, mount the EFI partition:

Code (Text):

```
cd ~/Projects/probook.git
sudo ./mount_efi.sh /
```

The config.plist you use from the project depends on your hardware. All plist files are located in the ./config folder. You can see what is there:

Code (Text):

```
cd ~/Projects/probook.git
ls config
```

As I write this, current content:

Code (Text):

```
config_1020_G1_Broadwell.plist    config_4x30s.plist        config_8x0s_G2_Haswell.plist
config_1040_G1_Haswell.plist    config_4x40s.plist        config_8x60p.plist
config_2x70p.plist        config_5x30m.plist        config_8x70p.plist
config_3x0_G1.plist        config_6x0s_G1_Haswell.plist    config_9x70m.plist
config_4x0s_G0.plist        config_6x60p.plist        config_ZBook_G0.plist
config_4x0s_G1_Haswell.plist    config_6x70p.plist        config_ZBook_G1_Haswell.plist
config_4x0s_G1_Ivy.plist    config_8x0_G3_Skylake.plist    config_ZBook_G2_Broadwell.plis
config_4x0s_G2_Broadwell.plist    config_8x0s_G1_Haswell.plist    config_ZBook_G2_Haswell
config_4x0s_G2_Haswell.plist    config_8x0s_G1_Ivy.plist    readme.txt
config_4x0s_G3_Skylake.plist    config_8x0s_G2_Broadwell.plist
```

Copy the appropriate plist for your hardware to EFI/Clover/config.plist on your EFI partition.

Note: If you needed to use the alternate DropTables during installation, you may also need it in your final config.plist.

For example, my ProBook 4530s:

Code (Text):

```
cd ~/Projects/probook.git
cp ./config/config_4x30s.plist /Volumes/EFI/EFI/Clover/config.plist
```

You could also copy the file using Finder.

After copying the config.plist from the repo to EFI/Clover/config.plist, you should customize the SMBIOS so you have a unique serial. You can use Clover Configurator to do this (use google to find/download it). DO NOT use Clover Configurator to edit your actual config.plist. Instead edit a "dummy" config.plist to create the SMBIOS data and then use copy/paste with a plist editor (I use Xcode) to copy the SMBIOS section into my active config.plist. Clover Configurator is too buggy and cannot be trusted with edits to your real config.plist. Make sure you use the same SMBIOS model as the original plist used.

Repeat: **DO NOT edit your config.plist with Clover Configurator**. Clover Configurator will erase important settings from the config.plist, and as a result, it will not work.

Do not stop reading

Although most of the post-install tasks are done, continue to read this guide. It it has important information you should know about.

Be sure to visit "Known Problems", especially the instructions in the "Audio:" section. Typically, even after completing all the above tasks, audio will not work on the first reboot. You must follow the specific instructions in that section in order to get the audio kexts in cache.

In the case of a problem, don't bother asking about with without all files requested in "Problem Reporting".

Updates to the patch repositories

From time to time, updates may become available to the project repository. In the event of such updates, you may want to update your copies, and re-patch DSDT/SSDT with the updates.

Since you're using git, it is easy...

In Terminal:

Code (Text):

```
cd ~/Projects/probook.git
git stash
git pull
./download.sh
./install_downloads.sh
./build.sh
```

In the case any of the SSDTs are updated, you will need to copy them to your EFI partition (as when you originally installed). Same goes for updates to the plist files. It is a good idea to update to the latest files before contemplating applying major updates to the system.

What works

I have tested the following features:
- UEFI booting via Clover
- built-in keyboard (with special function keys)
- built-in trackpad (basic gestures)
- HDMI video/audio with hotplug
- AirPlay mirroring to AppleTV
- WiFi, provided you have compatible hardware
- Bluetooth (with handoff) using BCM94352HMB
- native USB3 with AppleUSBXHCI (USB2 works also) (except 10.11 with 6-series laptops with NEC/Renesas)
- native audio with AppleHDA, including headphone
- built-in mic
- built-in camera (if you are lucky)
- native power management
- battery status
- backlight controls with smooth transitions, save/restore across restart
- accelerated graphics
- wired Ethernet
- Mac App Store working
- Messages/FaceTime (working on 10.10.3 even though I didn't run through the guide: http://www.tonymacx86.com/general-help/110471-how-fix-imessage.html)
- laptops with JMicron card reader work (ProBook/EliteBook 6-series and 7-series laptops)

Not tested/not working

The following features have issues, or have not been tested:
- you may notice the screen flicker on certain websites/apps with gray gradients
- card reader is not working on some models (would require significant development effort, not

important to me)

Known Problems

Fade to black after initial installation: Press your brightness up key.

Find My Mac/Locking: Find My Mac does not work properly. Don't lock your mac because it's difficult (or impossible) to unlock again.

Slow WiFi after sleep/wake cycle: Disable "Wake for network access" in SysPrefs->Energy Saver.

Audio: After installing new/updated kexts or after the initial installation, you may lose audio.

To fix:
- reboot without caches (hit space at Clover, select without caches)
- rebuild cache

Code (Text):

```
sudo touch /System/Library/Extensions && sudo kextcache -u /
```

- reboot normally (if needed, two times)

Alternate/experimental fix: https://www.tonymacx86.com/threads/...i-hotpatch-10-11.189416/page-499#post-1410398

Audio: External mic via requires mono-stereo adapter

Audio: If your laptop has a "combo jack" (combo headphones, external mic), the external mic will not work. Note: This is except for laptops with ALC280. The needed XML is in the project to enable full headset switching (4 pin) for ALC280 only equipped laptops such as the 8x0 G2, ZBook G2. Tested on the EliteBook 850 G2 Broadwell.

WiFi not working: Please realize you must install compatible hardware. Most of these laptops do not come with supported WiFi cards. If you have installed a compatible WiFi card, test it on Windows before concluding there is a problem with OS X. Finally, make sure the device is powered on. There is button on your keyboard deck for WiFi radio power. The LED often does not reflect the true state of power to the device, so don't mash it like a maniac. Press it once, and wait for the networks to populate... it is not instant. If it doesn't work, press again, and again... wait for the networks. If it is not working, check antenna connections and test on Windows.

Camera after sleep: On some laptops (4540s, others?) the Camera may intermittently not be detected after sleep (seems longer sleeps more than shorter sleeps). It is alleviated somewhat by portType=4 for the port associated with the camera, but still may happen occasionally. Workaround: sleep the laptop again, wake.

Another work around for camera after sleep was suggested by @kusakata: http://www.tonymacx86.com/threads/s...oard-id-airport-behavior.198094/#post-1290781:

> Also, for the webcam issues after sleep, mentioned in the post (I didn't have these in Yosemite, but do in El cap), I found a solution that doesn't require resleeping the laptop:
>
> Code (Text):

Click to expand...

Slow boot and constant fsck with certain SSDs: If you have fsck always running on boot, disable TRIM. The final config.plist used in this guide always enables the TRIM patch. You can easily disable it by setting Disabled=true. You will notice your boot is extremely slow and you will notice the fsck in verbose boot or system.log. I have this issue with my desktop and Mushkin SSDs, and has been reported here with Intel SSDs. I have no issue with Crucial (M4) or Samsung (EVO).

NVRAM in Skylake models: As is common with Skylake, native NVRAM via UEFI services often does not work. So make sure you install EmuVariableUefi-64.efi and the "RC scripts" when you install Clover.

USB3 on 6-series laptops with NEC/Renasas XHCI controller: This controller is supported with patches on 10.10.x and earlier. For 10.11.x and later, you can use GenericUSBXHCI.kext for partial support (not very reliable). Since there are detection issues with that kext, it helps to plug the device in prior to booting. For 10.11.x and later, 6-series laptops make a poor choice if you need reliable USB3.

USB customization

There are several USB SSDTs (build/SSDT-USB*.aml) provided for a few models. But you will notice that not all models are accounted for. Since USBInjectAll.kext is being used, all ports will be injected for models which lack a custom USB SSDT, but it is optimal to only inject the ports needed.

To assist in creating a new USB SSDT, the following procedure should be followed:

- enable the port limit patch if necessary (XHC 8086:8xxx and 100-series XHC).
(you will find the port limit patch in the config.plist but disabled by default)
- run IORegistryExplorer v2.1 (see "Problem Reporting" below for link)
- plug a USB3 device into each USB port (does not need to be all ports simultaneously)
- plug a USB2 device into each USB port (does not need to be all ports simultaneously)
- go back to the running instance of IORegistryExplorer and save the resulting ioreg file

Reply with an a note regarding your laptop hardware details and the resulting ioreg file. From the data in the ioreg a custom SSDT for your USB ports can be created.

The same procedure can be used if your USB ports/devices are not working (or some of them not working) even if you have a custom USB SSDT. Just boot with kernel flag -uia_ignore_rmcf so that the current custom configuration (for USBInjectAll.kext) is not in effect.

Other post-install tasks

Trackpad: Be sure to visit the options in SysPrefs->Trackpad and change them to your liking.

Trackpad three finger support: You can configure three finger swipes in SysPrefs->Keyboard->Shortcuts. Instead of pressing keys for a given function, do the three finger swipe (up/down/left/right).

Disable trackpad when using an external mouse: The latest script installs the VoodooPS2Daemon. It allows you to disable the built-in trackpad when a USB mouse is plugged in. Just check the box in SysPrefs->Accessibility->"Mouse & Trackpad".

Bluetooth: If you get the Bluetooth Setup Assistant popup, go to SysPrefs->Bluetooth->Advanced, uncheck the boxes.

Major updates

For example... 10.10 to 10.11.

First step should be to update to the latest repository as described in "Updates to the patch repositories".

Also update Clover to the latest using the Clover installer.

Also update config.plist at EFI/Clover/config.plist to the latest content from the repo. Be sure to retain your own SMBIOS data at config.plist/SMBIOS.

Now you can update via the App Store. Just boot the installer/updater upon restart.

After updating, run ./install_downloads.sh again:

Code (Text):

```
cd ~/Projects/probook.git
./install_downloads.sh
```

The reason for running ./install_downloads.sh both before and after the update is because the kexts installed by install_downloads.sh may depend on the version of OS X currently running.

To transition from PBI-CE based install

If you used the ProBook Installer-Clover Edition (PBI-CE), you can transition to this hotpatch setup:

- remove all files in EFI/Clover/ACPI/patched except for SSDT.aml (it came from ssdtPRGen.sh)
- then follow all instructions for post-install in the guide (ACPI/patched, config.plist)
- you can skip the part about "Power Management" unless you accidentally deleted SSDT.aml
- ./downloads.sh and ./install_downloads.sh can be used to update your kexts

If you have a "mixed" system (HD3000 on 7-series), the PBI-CE installed MacBookPro9,2 SMBIOS and two kexts, either AppleGraphicsPowerManagement_MacBookPro9_2.kext and ACPI_SMC_PlatformPlugin_MacBookPro9_2.kext or AppleGraphicsPowerManagement_MacBookPro10_1.kext and ACPI_SMC_PlatformPlugin_MacBookPro10_1.kext. With the new method for USB patching, these hacks are not needed. You should use an SMBIOS appropriate for your CPU (MacBookPro8,2) and delete these kexts.

Unsupported laptops

If you think your laptop might be similar to the ones listed as supported, but needs only small tweaks, please provide the following information:

- native DSDT/SSDT files extracted via Clover F4 (ACPI/origin)

- audio codec dump from Linux
- full hardware details (CPU, chipset, graphics device, screen resolution, etc)
- output from 'lspci -nn' in Linux Terminal

Notes regarding HDMI vs. DP

The files in this guide contain patches to change external ports from DP to HDMI. This may not be appropriate if your laptop has a DP port instead of HDMI.

If your laptop has DP, please provide the files requested in "Problem Reporting". Capture ioreg when the external monitor is connected. Please make sure your profile provides complete and accurate details.

Other Notes

This guide (and recent PBI-CE) does not install the FakeSMC sensor plugins. They are inefficient and waste CPU cycles even when HwMonitor is not running. As a result, various CPU status, heat sink temperatures, fan speeds will not show in HwMonitor.

You are welcome to install the plugins manually: https://github.com/RehabMan/OS-X-FakeSMC-kozlek/

After you install them manually, the ./download.sh and ./install_downloads.sh scripts will continue to update them (if FakeSMC is updated) even though those scripts won't install them initially.

Problem reporting

All problem reports should be accompanied by various files that allow your progress to be accounted for...

In terminal,

Code (Text):

```
rm -Rf ~/Downloads/RehabMan
mkdir ~/Downloads/RehabMan
cd ~/Downloads/RehabMan
patchmatic -extract
```

Attach contents of Downloads/RehabMan directory (as ZIP).

Also, post ioreg **as ZIP**: http://www.tonymacx86.com/audio/58368-guide-how-make-copy-ioreg.html. Please, use the IORegistryExplorer v2.1 attached to the post! DO NOT reply with an ioreg from any other version of IORegistryExplorer.app.

And output from:

Code (Text):

```
kextstat|grep -y acpiplat
kextstat|grep -y appleintelcpu
kextstat|grep -y applelpc
```

```
kextstat|grep -y applehda
```

Also output from:

Code (Text):

```
ls -l /System/Library/Extensions/AppleHDA.kext/Contents/Resources/*.zml*
```

Also, press F4 at the main Clover screen, then compress and attach the EFI/CLOVER folder in use. And please, in order to reduce the size, omit EFI/Clover/themes.

Also post output of:

Code (Text):

```
sudo touch /System/Library/Extensions && sudo kextcache -u /
```