[GUIDE] Yosemite install on Asus UX305FA

Before we begin, make a **backup of all data** or better yet a disk image to safe guard your data!
You have been warned! A backup is the only safeguard if something goes wrong.

Please read this install guide completely prior to any action, since it is a synopsis of required actions, You will be using other guides as well, by knowing this guide you will understand the interaction between various guides.

This guide is made possible by the efforts of a great many people who have worked hard to make such a guide for the UX305FA possible, we stand on their shoulders today.

Resources:

Asus Zenbook UX305FA compatability
[Guide] Booting the OS X installer on LAPTOPS with Clover UEFI
RehabMan/OS-X-Clover-Laptop-Config · GitHub
New RehabMan kexts/builds/binaries for Yosemite
[Guide] Patching LAPTOP DSDT/SSDTs
[Guide] Patching DSDT/SSDT for LAPTOP backlight control
RehabMan/ OS-X ACPI-Debug
Clover V2.3K Special Edition
Configuration SMBIOS

.
Please note we use “step (3)”, “stage (2)” and “phase (3)” they are distinctly different.
Steps, are the steps I took in the install process. Stages, are the Apple installers stages.
Phases, are the three phases of Clover EFI install process..

The **FIRST STEP** is to make an install DISK/USB from which the install might take place.

[Guide] Booting the OS X installer on LAPTOPS with Clover UEFI

The above guide is the definitive source for OS X install. Below you will find the steps I took.

Please make sure you installed HFSPlus.efi in /EFI/Clover/drivers64UEFI

The **SECOND STEP**, we need to make some changes to the BIOS/EFI. When the computer starts hit the escape key “esc”. We need to “Enter Setup” usually the bottom item.

REQUIRED CHANGES:

Video /Graphics “MUST” be set to “64”

Section Advanced:

Intel Virtualization	Disable
VT-d	Disable
Graphics	64

Section Boot:

Fast Boot	Disable
Launch GSM	Enable

Section Security:

Security Boot Menu	Enter
Secure Boot	Disable

Section Save & Exit:

Save Changes and Exit	Yes

The **THIRD STEP**, boot into installer.
Note: *You will need a USB keyboard and mouse for initial install*.
We are just looking for the system to boot up to the installer screen and this begins **PHASE ONE.** Select "Disk Utils and partition drive you will need at least one partition for install, default value is fine. The name you give this disk will be the "i**nstallei_disk_name**" used later for booting the system. Once completed exit Disk Utils. Choose "Install" and install OS X This will be the *first stage* of a *two stage* process. The computer will reboot. This ends Install **PHASE ONE** and stage one.

The computer will boot into "install_osx" this will be the start of install **PHASE TWO**.
The installer will automatically start *stage two* of install process.
Once completed the computer will reboot ending **PHASE TWO** and *stage two.*

Booting to "**installed_disk_name**".begins **PHASE THREE**.
The user will be prompted for information required for initial user setup etc. Upon completion the installers work is complete. However, we still have to install the Clover Boot Loader as described in Booting the OS X installer on Laptops with Clover EFI. Once Clover Boot loader is installed and post configuration completed, **PHASE THREE** is completed**.**

From this point forward we boot from the **"installed_disk_name"**. Keep the INSTALL DISK for emergency access to system. Maintaining the original INSTALL DISK/USB will ensure you have a copy of original "Clover Version Used during install". It is a good practice to also maintain the original Clover Package or zip for archival purposes.

At this point all "step", "stages" and "phases" are completed. Post install starts with a clean slate.

See POST INSTALL Asus UX305FA

[GUIDE] POST INSTALL Asus UX305FA

Items needed, refer to resources in Install Guide for assistance and Google.

Either Kext Wizard or Kext Utilities to install additional kexts.
MacIASL and associated files: iasl, patchmatic used for patching DSDT/SSDT
MacIASL must be configure and path to patch files added. See
[Guide] Patching LAPTOP DSDT/SSDTs
Optional but useful IOregistry-Explorer, EFI-Mounter, FileMerge and ACPI-Debug

Current Kexts List:

- VoodooHDA.kext
- ApplePS2SmartTouchPad.kext
- AsusNBFnKeys.kext
- ACPIBacklight.kext
- ACPIBatteryManager.kext
- FakeSMC.kext

Current List of patches:

- Fix _WAK Arg0 v2
- HPET Fix
- SMBUS Fix
- IRQ Fix
- RTC Fix
- OS Check Fix
- Fix Mutex with non-zero SyncLevel
- Fix PNOT/PPNT
- Add IMEI
- Fix ADBG Error
- Brightness_haswell
- Asus_N55SL/VivoBook
- There is one manual patch to apply for Fn F5 and Fn F6, simply copy patch and paste in patch window then apply (not in list of patches to select) FnKeyPatch.zip
- “Optional big patch over multiple files” Rename GFXO IGPU

RehabMan an excellent source of knowledge on all things Hackintosh, strongly believes that you should patch your own extracted DSDT/SSDT files to insure that the hardware for your specific machine is used in the patching process. Plus it is a good thing to learn for the next time. That said, sarge999 simply uses my builds since we have identical hardware. However UX305FA can have either Synaptics or Elan Touch Pad in this case the ApplePS2SmartTouchPad.kext supports both, but the DSDT could have subtle differences. As always the decision is made by the end user on how they proceed. I received an email that ApplePS2SmartTouchPad will be updated end of June/July.

Please note that once a DSDT is patched and binary used to boot the machine all previous patches will

be included in the extracted DSDT at this point in time.

You might say we have a fast lane and slow lane from here forward. Option fast lane is the use of DSDT/SSDT and config.plist I will post on finalization of this guide. Option slow lane is you build your own set of DSDT/SSDT and config.plist as preferred by RehabMan and learn a little something along the journey to our goal of a working OS X install.

Fast Lane:

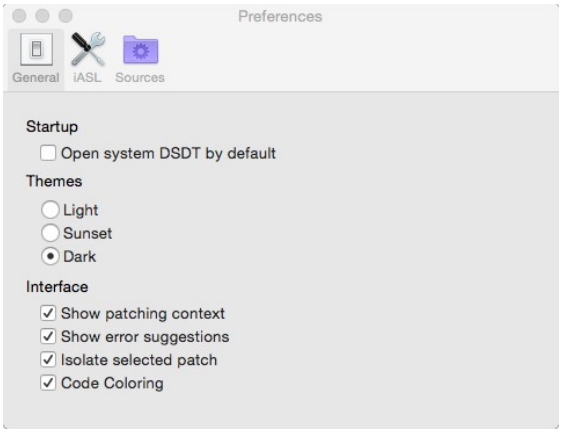
Fast lane requires you to download “FastLanePackage.zip”. Un-compress the files to a directory of your choice. You will find config.plist, DSDT and all SSDT binaries in .aml format. Do Not RENAME any of these files. Pick your option to mount the EFI partition so we can add or replace existing files, I use EFI Mounter for this. Copy config.plist to /EFI/CLOVER. Copy all DSDT/SSDT to /EFI/Clover/ACPI/patched. (e.g. sudo copy *.aml /EFI/CLOVER/ACPI/patched). To access the boot options hit space bar in Clover to display options menu. At this point reboot in SAFE MODE first time to ensure compatibility, some things don't work the same in Safe Mode graphics will not be accelerated and may blink and flash. If all goes well boot in normal mode, occasionally you might need to boot with no-caches to repopulate the cache if things like sound, battery etc fail to populate. You are now in the working OS X. environment.

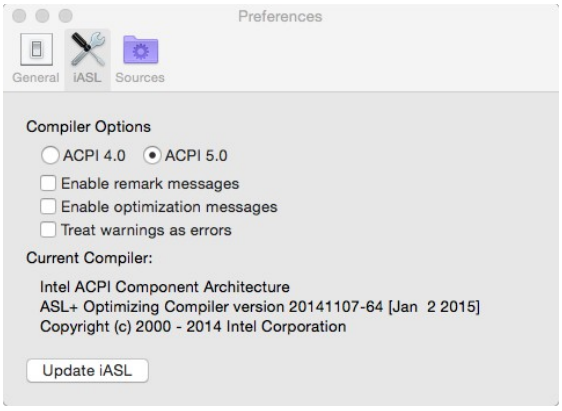
SLOW LANE:

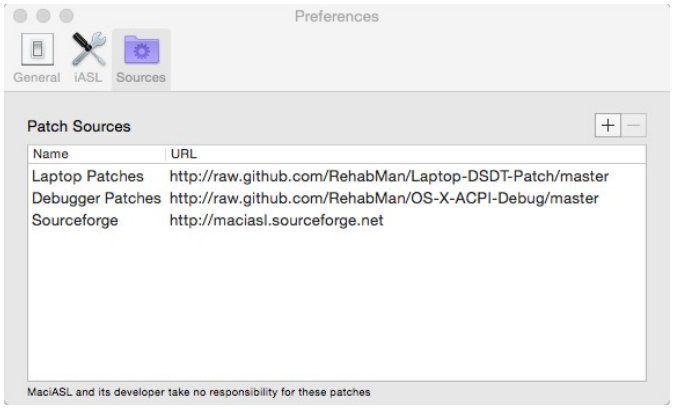
From this point on we are in the slow lane mode building a working DSDT and SSDT set of binaries for our particular machine. It is a good idea to become friendly with the terminal if you want to build your own DSDT/SSDT. Example requires components from MacIASL package. Example to populate a directory named “work-dsdt” on the desktop. In the terminal we would type “patchmatic -extract /Desktop/work-dsdt” this would extract a full set of DSDT/SSDT .aml (binaries) files. Then to disassemble the *.aml files we would “cd /Desktop/work-dsdt” now we are in the directory of all *.aml files. Disassemble with “iasl -da -dl *.aml” now the directory contains DSDT.dsl and DSDT.aml the same holds true with SSDT's as well. This would be your working set of files to patch. We work on the .dsl files in MacIASL and when we are finished patching compile the file to produce new .aml (binaries). One cautionary note it is very easy to forget to “apply” a patch, since you are concentrating on display of [patches changes rejects].

It is time to get MacIASL configured although the above example used iasl and patchmatic which are part of the MacIASL package. If you download RehabMan's MacIASLzip. You will have everything you need, once extracted in Downloads open a terminal and cd to /Downloads/Mac*. Next we want to put iasl and patchmatic in a directory contained in our execution path. In terminal you can confirm location via “pwd” or simply “ls” to make sure iasl and patchmatic are in working directory. In terminal type the following “sudo cp iasl patchmatic /usr/local/bin/” you will be prompted for root password. Once completed type “cd” this will return you to “home directory” type “iasl” then “patchmatic” to confirm they are both in the execute path. In Finder copy MacIASL to Applications.

With the files in place we will Open MacIASL and choose Preferences, a dialog window will appear.





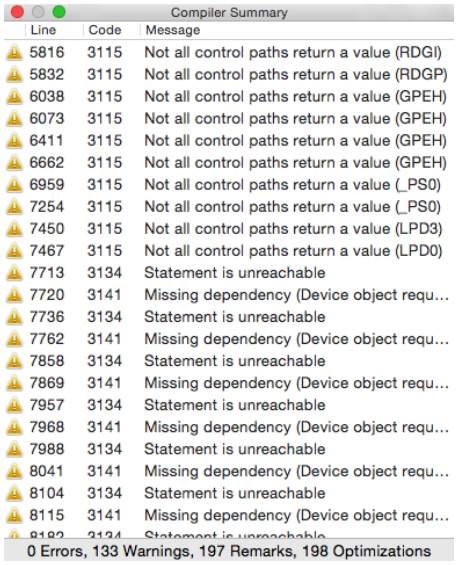


Simply match the settings above. Note: 1st image Startup un-checked this is important!
Worth noting MacIASL will not save until you actually use the save command or save as.
This allows us to patch and compile safely since nothing is saved util you specifically do so.
Thus if a patch goes horribly wrong, no harm no foul just reload and start over. It is good practice to compile before you patch to make sure the file is error free from the beginning. For fixing errors see [Guide] Patching LAPTOP DSDT/SSDTs .

You will see an error _DOS in this process the answer of how to fix this error is located one line above the error . and requires removal of comment "//". If my few last brain cells are correct you will also see an unreachable error you must correct.
Example:{

```
        return arg0
        arg1    you can never get here you returned above, comment it out with "//" infront //arg1
}
```

When patching make sure to compile at least every second patch so the problem is easier to identify and correct, secondly on compile you will get a dialog popup with a large amount of information, pay attention to lower window frame left side it will list errors so "0 Errors" means that we need not be concerned with warnings etc, since compile was successful.



We have the basics covered pretty well, so that leaves us with rename GFXO IGPU. This is a special case situation and RehabMan helped me out with resolution (see post 189). I will do a quick recap since I understand the anxiety level increased 10 fold. It really isn't as bad as it might seem and you must have gained some degree of confidence since you got this far, so take a deep breath and get a drink and we will start.

First lets save our work so far by copying the work-dsdt folder to safe-work-dsdt, we can always go back if need be now. Part of this process is a new disassembly of all files plus a new “ref.txt”. So we need our patched DSDT compiled and saved prior to this step. If you haven't done so please compile and save DSDT.aml (binary).

The “ref.txt” is a simple text file with the code provided pasted in and saved as “ref.txt”

Post 189 by RehabMan:

Create refs.txt:

Code:

```
External (_SB_.PCI0.PEG0.PEGP.SGPO, MethodObj, 2)
External (_SB_.PCI0.LPCB.H_EC.ECWT, MethodObj, 2)
External (_SB_.PCI0.LPCB.H_EC.ECRD, MethodObj, 1)
```

Then disassemble with it:

Code:

```
iasl -da -dl -fe refs.txt *.aml
```

Now open SSDT-8.dsl and move the Externals created by refs.txt to after the other externals:

Code:

```
...
    External (SCIS, FieldUnitObj)

    /*
     * External declarations that were imported from
     * the reference file [refs.txt]
```

```
    */
    External (_SB_.PCI0.PEG0.PEGP.SGPO, MethodObj, 2)
    External (_SB_.PCI0.LPCB.H_EC.ECWT, MethodObj, 2)
    External (_SB_.PCI0.LPCB.H_EC.ECRD, MethodObj, 1)

    OperationRegion (SANV, SystemMemory, 0xD9763E18, 0x018A)
...
```

Results: DSDT.dsl, SSDT-1.dsl, SSDT-3.dsl, SSDT-4.dsl, SSDT-6.dsl, SSDT-7.dsl all compile clean.

SSDT-2.dsl: needs "Remove _PSS placeholders"

SSDT-5.dsl: needs same "Move External from refs.txt" as for SSDT-8.dsl.

For some reason I found that I also required externals declared in SSDT-1.dsl for it to compile.

Now for all files *.dsl we must patch them with “rename GFXO IGPU” if you watch the [patches changes rejects] if none are applied(changes) we can move on to next file. Remeber to “apply patch” and compile then save both as *dsl and *.aml (binary).

If all has gone well we have our final set of DSDT/SSDT's now we have to make them usable by Clover. A config.plist **required** change to use the DSDT/SSDT package is, Root | ACPI | SSDT | Drop Oem = Yes. If you don't make the change it will simply hang during boot.

Now we need to mount the EFI partition to accomplish this task I use “EFI Mounter” , side note EFI Mounter sometimes has problem in multiple disks environments it fails, the fix in terminal “sudo mkdir /Volumes/EFI” then run it again and all is well. Next, all *.aml (DSDT/SSDT) files must be copied to /EFI/CLOVER/ACPI/patched/. Finally copy config.plist /EFI/CLOVER/.

Good Luck!

Jhawk

Revision:
-06 fixed hyperlinks, added FnKeyPatch.zip