u/CruxFail's Dell 7060 Build Notes

OpenCore 0.5.8, macOS Catalina 10.15.4

PC Summary

Make/Model	Dell OptiPlex 7060 Micro Form Factor
CPU	Intel Core i5-8500T @ 2.10GHz; 8th Gen, Coffee Lake S
GPU	UHD Graphics 630
RAM	SK Hynix 1x16 GB, 2666 MHz, DDR4 SDRAM
Motherboard	Dell Inc. 0KYJ8C (U3E1)
Chipset	Intel 300 Series LPC (Q370) – A306
Audio	Realtek ALC255/ALC3234 (Vendor Id: 0x10ec0255)
Ethernet	Intel I219-LM
WiFi/BT	Fenvi M.2 BCM94360NG
Storage	Toshiba 512GB NVMe
BIOS	v1.4.2

Legend for Highlighted Text Colors

Outstanding Questions

Answered Questions

Broken with Problem

Fixed with Solution

Want to Do Later

`Code or a Command`

Link Within This Document

System Successes

- iGPU
  - Video from both DisplayPort ports and one HDMI port work.
  - Dual-monitor output works.
- FileVault
- Time Machine
- Disabling CFG Lock
- Audio
- iServices
- USB Ports
- Sleep/Wake
- Wireless/Bluetooth

Outstanding Issues

Time in BIOS

- Issue: The Time in my BIOS will not stay set correctly. After a reboot and checking, it will always be +5 hours.
- Resolution:

Ejected USB Drives

- Issue: USB drives are ejected unexpectedly when in Sleep. I've read this is normal macOS behavior, but that seems hard to believe. I tried Jettison.app but that didn't consistently work.
- Resolution:

Bluetooth Keyboard in BIOS

- Issue: I have a Logitech MX Keys and MX Master 3 Mouse that support Bluetooth. I use them without Bluetooth (using the Logitech dongle

instead) because I have not figured out how to use them in the BIOS or Boot Picker. It seems that Bluetooth is not activated until after I login.
- Resolution:

Solved Issues

Default Boot Option

- Issue: Using “CTRL + Enter” and “CTRL + Index” to set a default boot device in the picker does not work for me. Nothing seems to change. It will always go back to the first choice in the Boot Picker. I am using the GUI, so I don’t know if that changes anything.
- Resolution: I do not think this works with “OpenCanopy.efi” (the GUI Boot Picker) yet. I turned off the GUI and now I can select a default. You can tell the default choice is selected by the asterisk that is listed before the number.

CPUFriend

- Issue: I was not able to figure out CPUFriend and CPUFriendFriend to customize power management. Specifically, I could now figure out how to use “CPUFriendFriend.py” properly. I’m hoping I will be able to find some help on that one.
- Resolution: I kept playing around and I figured out that opening Terminal, then dragging “CPUFriendFriend.py” into the Terminal window, will initiate the script properly. It now produces all the files shown in the OCD Guide.

Intermittent Slow Performance

- Issue: Overall, something doesn’t seem optimized in CPU management or Power management. The system runs great most of the time, but certain tasks and processes really slow down the PC. The problem is, it seems a bit random. The same thing that slows everything down one time will work fine next time I try. For instance, all of a sudden Word for Mac takes 70% CPU, but then after a reboot it’s fine. I’m hoping that once I figure out CPUFriend and CPUFriendFriend this will be resolved.
- Resolution: As hoped, this has improved since adding CPUFriend and CPUFriendFriend.

Intermittent Slow Boot

- Issue: Sometimes the Dell splash-screen will stay for upwards of 5 minutes before the OC Boot Picker shows up.
- Resolution: This turned out to be caused partly by my "Patriot Spark" 64GB USB 3.0 drives. I still need to test each port, as it seems some ports in combination with the USB drives cause the slowness, while others do not.

My Preliminary Steps

Previously done when installing via OpenCore v0.5.7:

- I created a Windows 10 installation USB from another working PC using the MediaCreationTool.
- I downloaded the following and placed them on the installation USB:
    - Speccy installer (v1.32.740)
    - Python installer (v3.8.2)
    - iasl (20180105)
    - SSDTTime (57c9377)
- I installed Windows 10 on the destination PC.
- I installed Speccy on destination PC.
- I installed Python on destination PC.
    - I selected “Add Python 3.8 to PATH”.
    - Installed for all users.
    - Disabled path length limit at the end of installation.
- I unzipped “SSDTTime-master.zip” to the desktop on the destination PC.
- I unzipped the contents of “iasl-win-20180105.zip” and moved them to the \SSDTTime-master\Scripts folder.
    - https://github.com/corpnewt/SSDTTime/issues/9

Preliminary steps this time:

- I downloaded OpenCorePkg (v0.5.8).

OpenCore Desktop Guide

Lovingly referred to as the OCD Guide from here on out.

https://dortania.github.io/OpenCore-Desktop-Guide/

Creating the USB

macOS Install

Making the installer in macOS

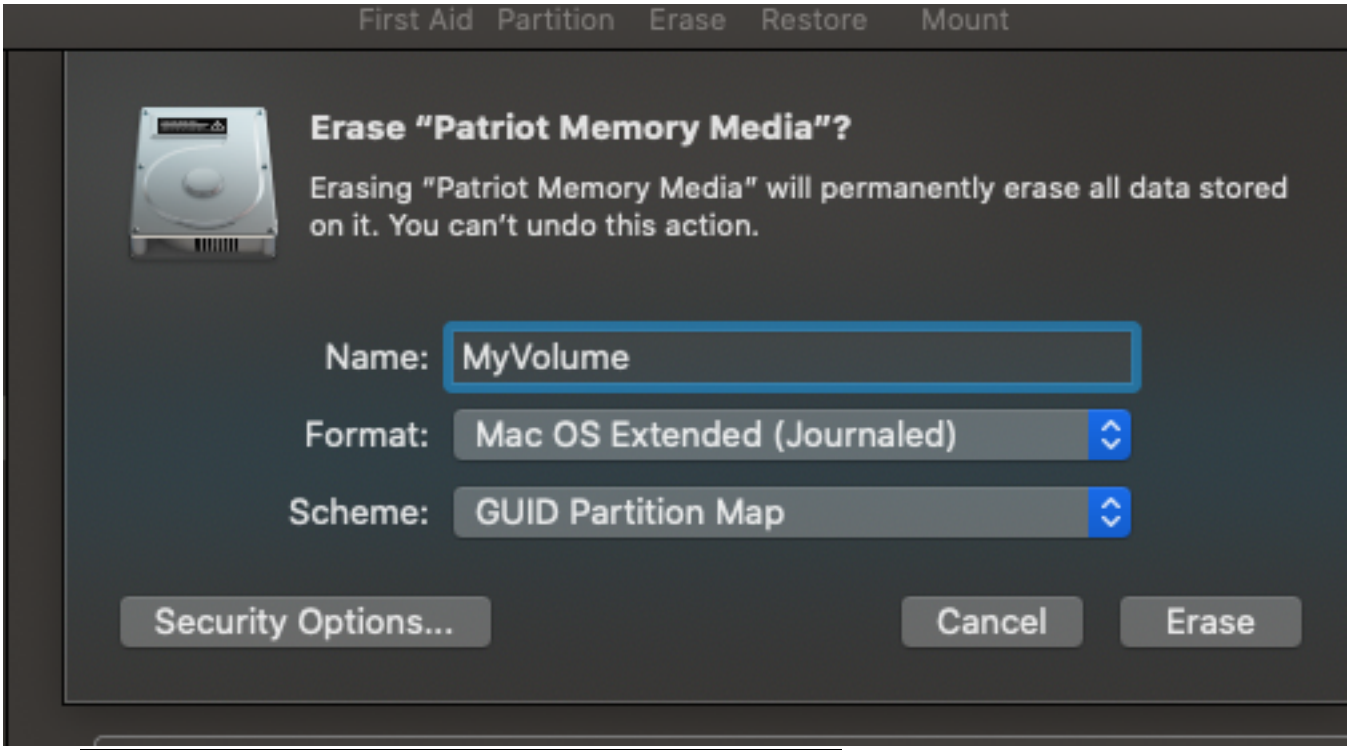
- This time I decided to recreate the bootable USB with “Install macOS Catalina.app” from the App Store.

Downloading macOS

- I downloaded “Install macOS Catalina.app” from the App Store, but instructions on how I did it in Windows with gibMacOS can be found below.

Setting up the installer

- I formatted my drive following the directions in the OCD Guide.



- I ran `sudo /Applications/Install\ macOS\ Catalina.app/Contents/Resources/createinstallmedia --volume /Volumes/MyVolume` in Terminal.

Setting up OpenCore's EFI environment

- I mounted the EFI of my new USB drive using MountEFI and copied the EFI folder from “OpenCore-0.5.8-RELEASE” into the EFI partition on my USB.

Windows Install

- I set this up during my first attempt. I will keep the notes here in case they are needed.

Making the installer in Windows

- I created the USB using gibMacOS in Windows.

Downloading macOS

- gibMacOS would not run.
    - I had to make sure gibMacOS was on the system drive and not on another drive.
- I downloaded "061-96006 - 10.15.4 macOS Catalina"

Making the installer

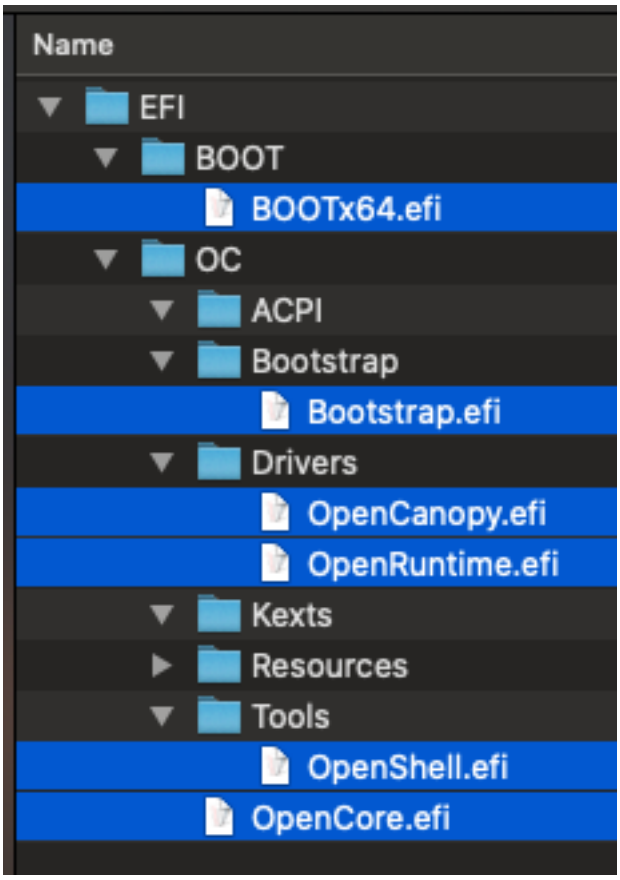
- When formatting the drive, I ran into the error "DiskPart exited with non-zero status (1). Aborting."
    - https://github.com/corpnewt/gibMacOS/issues/62
    - For me, I had to temporarily remove the "macOS Downloads" folder out of gibMacOS, delete gibMacOS, re-extract gibMacOS from the .zip, move the "macOS Downloads" folder back, and then try the "MakeInstall.bat" again. This happens every time I go back to make another USB and I'm not sure why. But hey, it works.

Linux Install

- I have not tried this.

Adding the Base OpenCore Files

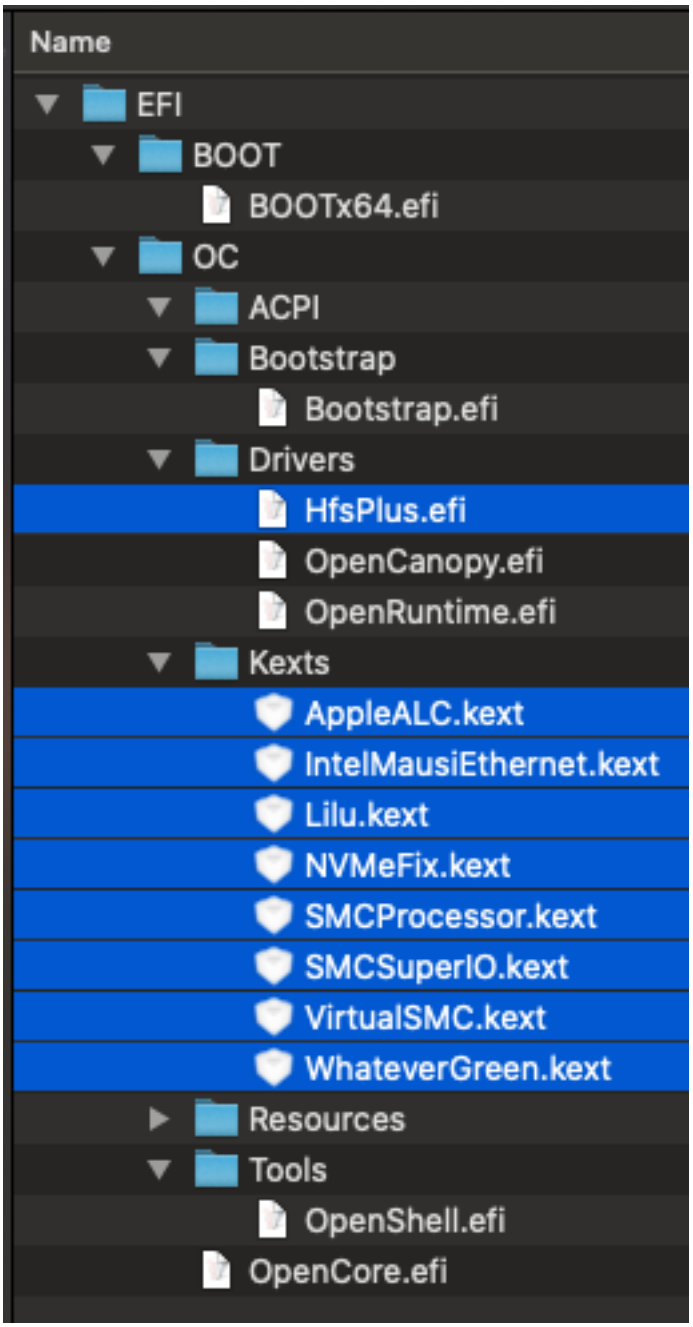
- I followed the directions. My EFI folder now looks like this:



- I left “OpenCanopy.efi” because I am using the GUI Boot Picker.
  - I later removed this so that I could pick a default boot option and hide the picker altogether for a cleaner boot.

Gathering Files

- I followed the directions in the OCD Guide. My EFI folder now looks like this:



Getting started with ACPI

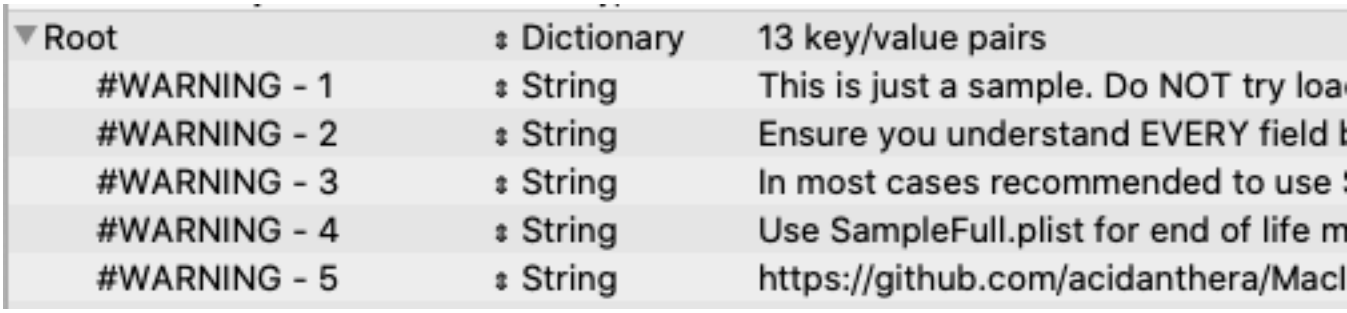
This portion of the OCD Guide has been branched out into a separate page. Click the link below to be taken to that section in this document:
Link => Getting Started with ACPI <= Link

INTEL CONFIG.PLIST

Coffee Lake

Starting Point

- I already copied the “Sample.plist” from the \OpenCore-0.5.8-RELEASE\Docs folder to the \EFI\OC folder and renamed it to “Config.plist” back when I was using SSDTTime.
- I opened my “Config.plist” with ProperTree and ran the “OC Clean Snapshot” option to add all of my SSDTs, Kexts, and Drivers.
- I deleted the 5 warning entries at the top of the “Config.plist” from the sample.



ACPI

Add:

- I have the following SSDTs listed and set to True:
  - SSDT-EC-USBX.aml
    - For Embedded Controller support and USB power management.
  - SSDT-PLUG.aml
    - For CPU power management.
  - SSDT-SBUS-MCHC.aml
    - For fixing AppleSMBus support in macOS.
  - SSDT-HPET.aml
    - For fixing IRQ and specifically audio on this PC.

Block:

- As recommended in the OCD Guide, I did not change anything.

Patch:

- In order to get “SSDT-HPET.aml” (and therefore audio) to work, the patches from “patches_OC.plist” must replace the patches here.
- The “patches_OC.plist” was created using SSDTTime in Windows. I could not get this to work in macOS.

Quirk:

- I followed all OCD Guide recommendations:
    - FadtEnableReset: FALSE
    - NormalizeHeaders: FALSE
    - RebaseRegions: FALSE
    - ResetHwSig: FALSE
    - ResetLogoStatus: FALSE

Booter

MmioWhitelist:

- As recommended in the OCD Guide, I did not change anything.

Quirks:

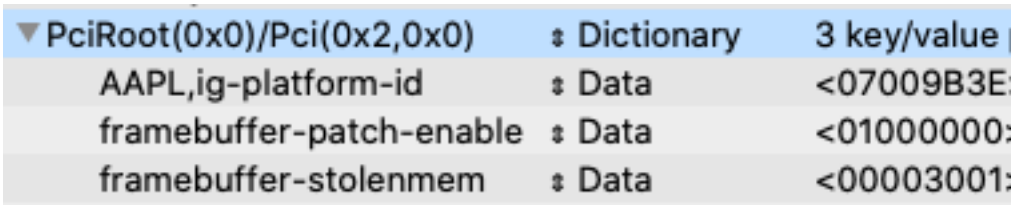
- I followed all OCD Guide recommendations:
    - AvoidRuntimeDefrag: TRUE
    - DevirtualiseMmio: TRUE
    - DisableSingleUser: FALSE
    - DisableVariableWrite: FALSE
    - DiscardHibernateMap: FALSE
    - EnableSafeModeSlide: TRUE
    - EnableWriteUnprotector: TRUE
    - ForceExitBootServices: FALSE
    - ProtectMemoryRegions: FALSE
        - Would this help with USBs disconnecting when macOS goes into Sleep?
    - ProtectSecureBoot: FALSE
    - ProtectUefiServices: FALSE
    - ProvideCustomSlide: TRUE
    - RebuildAppleMemoryMap: TRUE
    - SetupVirtualMap: FALSE

- SignalAppleOS: FALSE
- SyncRuntimePermissions: TRUE

DeviceProperties

Add:

- This section has been customized for my iGPU and Audio Controller:
  - **PciRoot(0x0)/Pci(0x2,0x0):**
    - First Install (used example):
      - I have the UHD 630 iGPU. That card is listed in the OCD Guide as an example. I added the settings from the guide to resolve an initial kernel panic and they worked. Next time around I checked the Framebuffer Patching Guide.

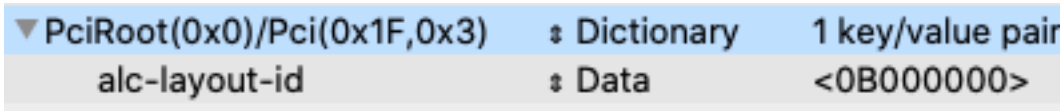


      - Value 00003001 = 19MB
    - Second Install (used Framebuffer Patching Guide):
      - I read through the framebuffer guide and noted the following:
        - AAPL,ig-platform-id
          - "0x3E9B0007" is the recommended platform ID for the UHD 630 on Coffee Lake, so it makes sense to try this one first. The swapped hexadecimal bits make it "07009B3E".
        - framebuffer-patch-enable
          - "01000000" simply enables "WhateverGreen.kext".
        - framebuffer-stolenmem
          - This sets the minimum stolen memory amount for the GPU. I do not have the option in the Dell BIOS to set Memory

Allocation for the iGPU in this PC, so the guide recommends setting this to 19MB.

- 19 Megabytes converted to Bytes (binary format) = 19922944
- 19922944 Decimal converted to Hexadecimal (signed 2's compliment) = 01300000
- 01300000 split up into pairs of hex bytes (01 30 00 00) and reversed = 00 00 30 01
- Yeah, makes total sense to me…

- **PciRoot(0x0)/Pci(0x1b,0x0):**
  - I fixed this post-install on my first attempt. Details are in the section "Post-Install => Fixing Audio". The gist of the effort is:
    - "PciRoot(0x0)/Pci(0x1b,0x0)" needs to be changed to "PciRoot(0x0)/Pci(0x1F,0x3)" because that is the location of my audio controller.
    - "layout-id = 01000000" needs to be changed to "alc-layout-id = 0B000000". This is Layout ID 11, first tested as a boot argument, then converted to Hex and using the "alc-layout-id" for a more permanent layout fix.



Block:

- As recommended in the OCD Guide, I did not change anything.

Kernel

Add:

- All Kexts should be listed here and Enabled. Lilu should be at the top. The order matters; with dependencies being listed first and dependents below. The Clean OC Snapshot in ProperTree placed them in this order. I have 8 in this order:
  - Lilu.kext
  - VirtualSMC.kext
  - AppleALC.kext

    - IntelMausiEthernet.kext
    - NVMeFix.kext
    - SMCProcessor.kext
    - SMCSuperIO.kext
    - WhateverGreen.kext
        - I am not sure if WEG needs to be in a higher priority location. It seems to be working fine.

Block:

- As recommended in the OCD Guide, I did not change anything.

Emulate:

- As recommended in the OCD Guide, I did not change anything.

Patch:

- As recommended in the OCD Guide, I did not change anything.

Quirks:

- I followed all OCD Guide recommendations, except AppleCpuPmCfgLock, AppleXcpmCfgLock, DisableIoMapper, and XhciPortLimit:
    - AppleCpuPmCfgLock: FALSE
        - This is usually needed because there is no access to CFG-LOCK in the Dell BIOS. However, I later found out how to disable the option by reading the “Post-Install => Fixing CFG Lock” section of the OCD Guide.
    - AppleXcpmCfgLock: FALSE
        - This is usually needed because there is no access to CFG-LOCK in the Dell BIOS. However, I later found out how to disable the option by reading the “Post-Install => Fixing CFG Lock” section of the OCD Guide.
    - AppleXcpmExtraMsrs: FALSE
    - AppleXcpmForceBoost: FALSE
    - CustomSMBIOSGuid: FALSE
    - DisableIoMapper: FALSE
        - I turned this off because I can turn off VT-D in the BIOS. It’s listed in the Dell BIOS as “Virtualization Support => VT for Direct I/O”.
    - DisableRtcChecksum: FALSE
    - DummyPowerManagement: FALSE

    - ExternalDiskIcons: FALSE
    - IncreasePciBarSize: FALSE
    - LapicKernelPanic: FALSE
    - PanicNoKextDump: TRUE
    - PowerTimeoutKernelPanic: TRUE
    - ThirdPartyDrives: FALSE
        - "Enables TRIM, not needed for NVMe but AHCI based drives may require this. Please check under system report to see if your drive supports TRIM."
        - I might need this when I install my secondary internal SSD.
    - XhciPortLimit: FALSE
        - I turned this off because making a USB Map was recommended instead.

Bless Override:

- This is not mentioned in the guide. I looked it up in the "Configuration.pdf" and it looks like I do not need this.

Boot:

- I followed all OCD Guide recommendations, except PollAppleHotKeys and ShowPicker:
    - ConsoleAttributes: 0
    - HibernateMode: None
    - HideAuxiliary: FALSE
    - HideSelf: TRUE
    - PickerAttributes: 0
    - PickerAudioAssist: FALSE
    - PickerMode: Builtin
    - PollAppleHotKeys: TRUE
        - I set this post-install during my first install. It is for FileVault support. Do not do this if FileVault is not needed.
    - ShowPicker: FALSE
        - This is not mentioned in the guide, but I looked it up in the "Configuration.pdf". It is to show/hide the boot picker altogether. I set this to "False" after I was confident that I could select a default boot option (Misc => Security =>

AllowSetDefault: TRUE) and access the boot picker with the Option or Escape key (Misc => Boot => PollAppleHotKeys).
    - TakeoffDelay: 0
    - Timeout: 5

Debug:

- I followed all OCD Guide recommendations, except AppleDebug:
    - AppleDebug: FALSE
        - I will turn this on if I run into issues.
    - DisableWatchDog: TRUE
    - DisplayDelay: 0
        - This is not mentioned in the guide, but I looked it up in the "Configuration.pdf" and it looks like I can leave it at 0.
    - DisplayLevel: 2147483650
        - It will be useless as I am not using the DEBUG version of OC.
    - Target: 67
        - It will be useless as I am not using the DEBUG version of OC.

Entries:

- This is not mentioned in the guide. I looked it up in the "Configuration.pdf" and it looks like I do not need this. It's used to manually add boot entries.

Security:

- I followed all OCD Guide recommendations:
    - AllowNvramReset: TRUE
    - AllowSetDefault: TRUE
        - Allow CTRL+Enter and CTRL+Index to set default boot device in the picker. I will set this once my config is final.
        - This does not seem to work in the GUI Boot Picker (OpenCanopy.efi).
    - AuthRestart: TRUE
        - I turned this on as the guide recommended for FileVault. It's less secure when this is on. I would like to turn this off later to see what kind of inconvenience it causes.
    - BootProtect: None
    - ExposeSensitiveData: 6
        - It will be useless as I am not using the DEBUG version of OC.

    - HaltLevel: 2147483648
        - This is not mentioned in the guide. I looked it up in the “Configuration.pdf” but I do not understand if I need to alter anything. I left it alone.
    - ScanPolicy: 0
    - Vault: Optional

Tools:

- As recommended in the OCD Guide, I did not change anything. The Tools I have in the \EFI\OC\Tools folder show here.
    - OpenShell.efi

NVRAM

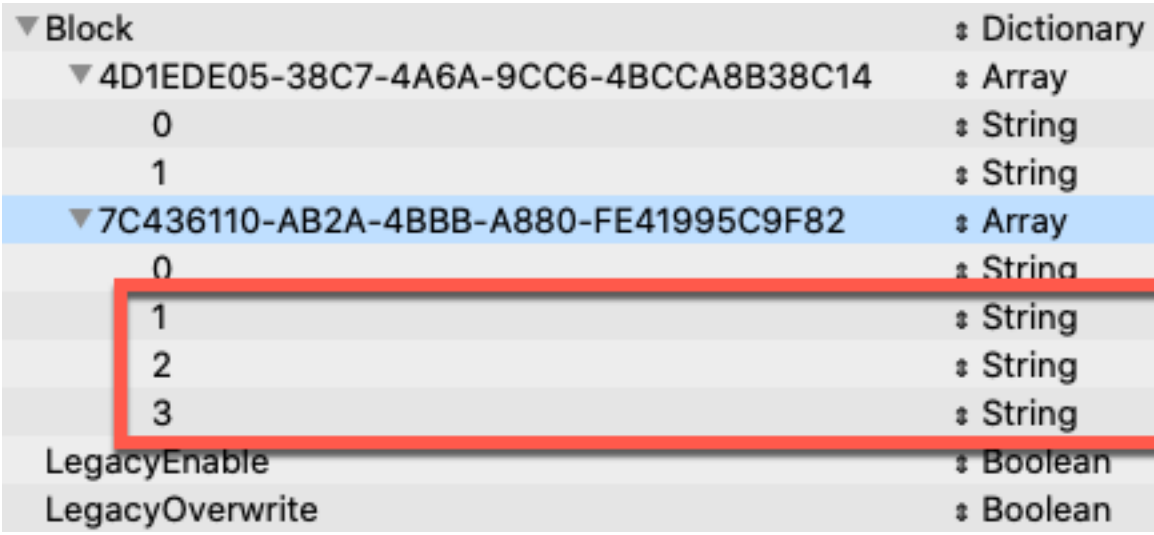
Add

- This section has been customized for my PC:
    - 4D1EDE05-38C7-4A6A-9CC6-4BCCA8B38C14 (Booter Path, mainly used for UI Scaling)
        - DefaultBackgroundColor: 00000000
            - This is used to select Black as the background color during boot.
        - UIScale: 1
            - I left this as 01 even though it is suggested to change to 02 for FileVault. Setting this to 02 produced a large font in the OC Picker and a large Apple logo on boot.
    - 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102
        - This is not mentioned in the guide. I looked it up in the “Configuration.pdf” but I do not understand if I need to alter anything. I left it alone.
            - I noticed a reference to this shows during every boot right before the Boot Picker GUI shows up. I’m wondering if I can safely remove it.
            - I do not need an rtc-blacklist (pg. 60 in the OpenCore “Configuration.pdf”). I deleted this section from the "Configuration.plist". This was added to the "Sample.plist" in 0.5.8.

    - 7C436110-AB2A-4BBB-A880-FE41995C9F82 (System Integrity Protection bitmask)
        - SystemAudioVolume: 46
            - This is the default system audio level when booting.
        - Boot-args: debug=0x100 keepsyms=1
            - These are the only boot arguments I wanted.
        - Csr-active-config: 00000000
            - This enables SIP (System Integrity Protection).
        - Prev-lang:kbd: 656E2D55 533A30
            - This is to select English as the language.

Block

- 7C436110-AB2A-4BBB-A880-FE41995C9F82
    - I added a string for “SystemAudioVolume”, “csr-active-config”, and “prev-lang:kbd” so that these variables would always be added to NVRAM.



- 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102
    - I do not need an rtc-blacklist (pg. 60 in the OpenCore “Configuration.pdf”). I deleted this section from the "Configuration.plist". This was added to the "Sample.plist" in 0.5.8.

LegacyEnable

- As recommended in the OCD Guide, I set this to False.

LegacyOverwrite

- As recommended in the OCD Guide, I set this to False.

LegacySchema

- I did not change any of these variables since it is only used if LegacyEnable is True.

WriteFlash

- As recommended in the OCD Guide, I set this to True.

PlatformInfo (SMBIOS)

- I used “GenSMBIOS.bat” to generate everything I need. I saved the text to a file I called “GenSMBIOS Results.txt” for later reference.

Automatic

- As recommended in the OCD Guide, I set this to True.

Generic

- I used the following settings:
    - AdviseWindows: FALSE
    - MLB: Board Serial produced by GenSMBIOS.
    - ROM: As recommended, I set this to my NICs MAC address.
    - SpoofVendor: TRUE
    - SystemProductName: iMac19,1
        - Was already set to this.
    - SystemSerialNumber: Serial produced by GenSMBIOS.
    - SystemUUID: SmUUID produced by GenSMBIOS.

UpdateDataHub

- As recommended in the OCD Guide, I set this to True.

UpdateNVRAM

- As recommended in the OCD Guide, I set this to True.

UpdateSMBIOS

- As recommended in the OCD Guide, I set this to True.

UdateSMBIOSMode

- As recommended in the OCD Guide, I set this to Create.

UEFI

AFPS

- I followed all OCD Guide recommendations:

    - EnableJumpstart: TRUE
    - HideVerbose: TRUE
    - JumpstartHotPlug: FALSE
    - MinDate: 0
    - MinVersion: 0

Audio

- As recommended in the OCD Guide, I did not change anything. I also do not want the Bootchime.

ConnectedDrivers

- As recommended in the OCD Guide, I set this to True.

Drivers

- All the drivers were added during the initial OC Clean Snapshot done in OpenTree.
    - HfsPlus.efi
    - OpenCanopy.efi
    - OpenRuntime.efi

Input

- I followed all OCD Guide recommendations:
    - KeyFiltering: FALSE
    - KeyForgetThreshold: 5
    - KeyMergeThreshold: 2
    - KeySupport: TRUE
    - KeySupportMode: Auto
    - KeySwap: FALSE
    - PointerSupport: FALSE
    - PointerSupportMode: BLANK
    - TimerResolution: 50000

Output

- I followed all OCD Guide recommendations:
    - ClearScreenOnModeSwitch: FALSE
    - ConsoleMode: BLANK
    - DirectGopCacheMode: BLANK
        - This is not mentioned in the Guide and I left it as is.

    - DirectGopRendering: FALSE
    - IgnoreTextInGraphics: FALSE
    - ProvideConsoleGop: TRUE
    - ReconnectOnResChange: FALSE
    - ReplaceTabWithSpace: FALSE
    - Resolution: Max
    - SanitiseClearScreen: FALSE
    - TextRenderer: BuiltinGraphics

ProtocolOverrides

- I set the following variables post-install my first time for FileVault support. Do not do this if FileVault is not needed.
    - AppleSmcIo: TRUE
    - FirmwareVolume: TRUE
    - HashServices: TRUE

Quirks

- I followed all OCD Guide recommendations:
    - ExitBootServicesDelay: 0
    - IgnoreInvalidFlexRatio: FALSE
    - ReleaseUsbOwnership: FALSE
    - RequestBootVarFallback: TRUE
    - RequestBootVarRouting: TRUE
    - UnblockFsConnect: FALSE

ReservedMemory

- As recommended in the OCD Guide, I did not change anything.

Cleaning up

- I ran the Sanity Checker. Everything checked out as expected.

Intel BIOS settings

Disable:

- Fast Boot
    - This can be found in the Dell BIOS: Post Behavior => Fast Boot => set to “Thorough”
- VT-d (can be enabled if you set DisableIoMapper to TRUE)

    - This can be found in the Dell BIOS: Virtualization Support => VT for Direct I/O => leave Unchecked
- CSM (Compatibility Support Module)
    - My Dell BIOS does not have this option.
        - It may be the same as Legacy Boot options and I disabled those.
- Thunderbolt (For initial install, as Thunderbolt can cause issues if not setup correctly)
    - My system does not have Thunderbolt.
- Intel SGX
    - This can be found in the Dell BIOS: Intel Software Guard Extensions => Intel SGX Enable => set to “Disabled”
- Intel Platform Trust
    - This can be found in the Dell BIOS: Security => TPM 2.0 Security => leave Unchecked
- CFG Lock (MSR 0xE2 write protection)
    - My Dell BIOS does not have this option.
        - I permanently fixed this using the “CFG Lock” section of this Guide.

Enable:

- VT-x
    - This can be found in the Dell BIOS: Virtualization Support => Virtualization => leave Checked
- Above 4G decoding
    - My Dell BIOS does not have this option.
- Hyper-Threading
    - My Dell BIOS does not have this option.
- Execute Disable Bit
    - My Dell BIOS does not have this option.
- EHCI/XHCI Hand-off
    - My Dell BIOS does not have this option.
- OS type: Windows 8.1/10 UEFI Mode
    - My Dell BIOS does not have this option.
- DVMT Pre-Allocated (iGPU Memory): 64MB
    - My Dell BIOS does not have this option. It’s taken care by “WhateverGreen.kext” and DeviceProperties => Add = > framebuffer-stolenmem in the “Config.plist”.

INSTALATION

Installation Process

TROUBLESHOOTING

General Troubleshooting

OpenCore Debugging

macOS Boot Process

POST INSTALL

Security and File Vault

FileVault

- I already moved “OpenRuntime.efi” to the \EFI\OC\Drivers folder on my USB drive in the “Adding The Base OpenCore Files” section of the OCD Guide.
- I left the \NVRAM\ADD\4D1EDE05-38C7-4A6A-9CC6-4BCCA8B38C14 UIScale set to 01. Setting this to 02 produced a large font in the OC Picker and a large Apple logo on boot.
- I enabled FileVault in macOS System Settings. The drive completed encryption and it remained working after a reboot.

Vault

- I did not set this up.

Fixing Audio

Finding your layout ID

- My audio device was listed on a Dell website as Realtek ALC3234.

    - This is also known as the Realtek ALC255, which can be seen in the VendorID. See the "Add GUI and Bootchime" section of the OCD Guide for a screenshot.
- I looked up AppleALC Supported Codecs and got the following layout IDs:
    - 3, 11, 13, 15, 17, 18, 21, 27, 28, 30, 31, 99
    - 7, 71, 100 – these were mentioned elsewhere so they might work

Testing your layout

ID	Hex Value	Result
3	03000000	static
7	07000000	boot sound
11	0B000000	boot sound
13		static
15		boot sound
17		static
18		static
21		static
27		boot sound
28		stopped testing

- Layout 11 worked after creating "SSDT-HPET.aml" in SSDTTime and patching my "Config.plist".

Making Layout ID more permanent

- I opened Hackintool and navigated to the PCIe section to find my PC's audio contoller. Look for "HDEF" in the IOReg Name column:

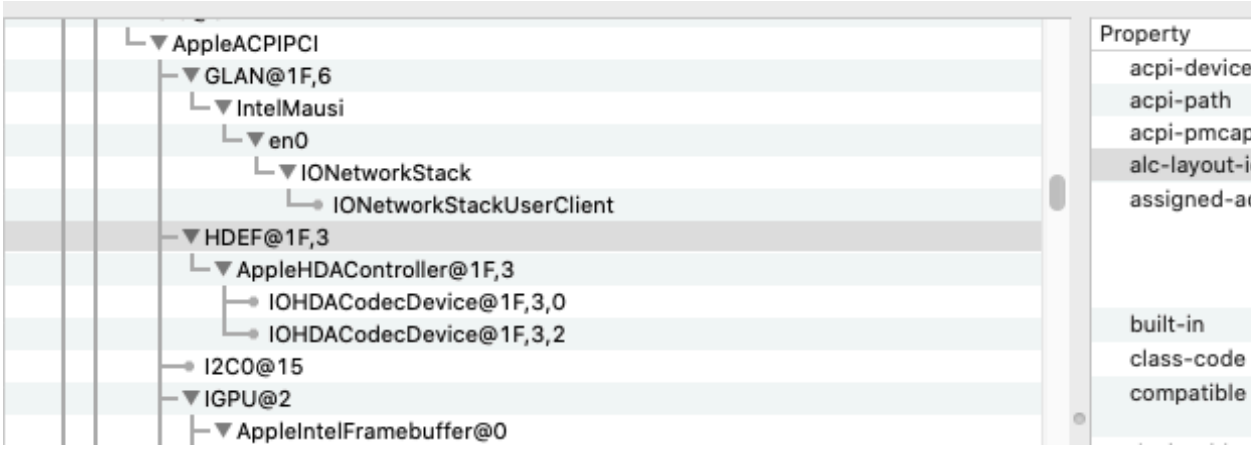
ISA bridge	/PCI0@0/LPCB@1F	pci8086,a306	PciRoot(0x0)/Pci(0x1F,0
Audio device	/PCI0@0/HDEF@1F,3	pci8086,a348	PciRoot(0x0)/Pci(0x1F,0
SMBus	/PCI0@0/SBUS@1F,4	pci8086,a323	PciRoot(0x0)/Pci(0x1F,0

- I updated DeviceProperties => Add => PciRoot(0x0)/Pci(0x1b,0x0) in the "Config.plist" to be "PciRoot(0x0)/Pci(0x1F,0x3)".
- I changed DeviceProperties => Add => PciRoot(0x0)/Pci(0x1b,0x0) => layout-id to be "alc-layout-id" with a value of "0B000000".

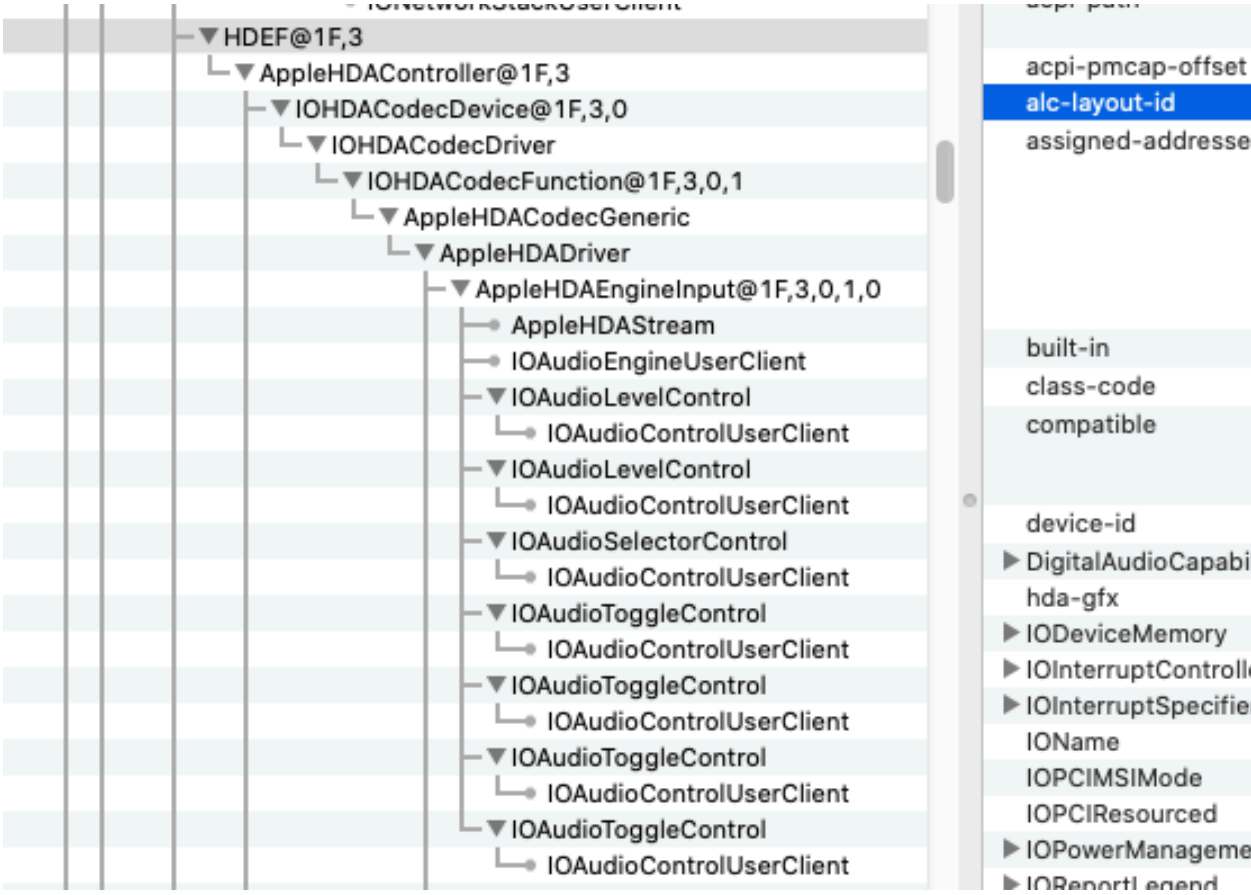
▼PciRoot(0x0)/Pci(0x1F,0x3)	⇕ Dictionary	1 key/value
alc-layout-id	⇕ Data	<0B00000

    - This is an example of my audio not working properly. In IORegistryExplorer you can see the HDEF audio controller loaded and

the AppleHDAController did, too. But there are many missing devices underneath.



- This is the fixed version once I applied “SSDT-HPET.aml” and the patch it made.



Booting without USB

- I used MountEFI to mount the EFI from my USB.
- I copied the EFI folder to my desktop.

- I dismounted the USB EFI and the USB itself.
- I used MountEFI to mount the EFI of my internal drive.
- I deleted the current internal EFI folder and logs, then emptied the trash (needed of that partition will run out of space).
- I moved the EFI folder from my desktop to the internal EFI folder and rebooted.
- Once I verified everything was working, I made a copy of that EFI folder from my USB drive and saved it as a "snapshot" for future use if needed.

Updating OpenCore, kexts and macOS

Updating OpenCore

- Will investigate another time.

Updating Kexts

- Will investigate another time.

Updating macOS

- Will investigate another time.

Fixing CFG Lock

- I don't understand why the OCD Guide asks you to change the "Config.plist" without first knowing if CFG-Lock can be turned off. So, I performed the following steps first:
    - I copied "VerifyMsrE2.efi" from the \OpenCore-0.5.8-RELEASE\EFI\OC\Tools folder to the \EFI\OC\Tools folder on my USB.
    - I added this tool to Misc => Tools in my "Config.plist" by doing an OC Snapshot in ProperTree.
    - I restarted my PC and selected "VerifyMsrE2.efi" from the Boot Picker.

- This is the output:

```
Looking up EFI_MP_SERVICES_PROTOCOL...
Checking MSR 0xE2 on all CPUs. Values must be SAME!!!
CPU00 has MSR 0xE2: 0x0000000001E008008
Starting All APs to verify 0xE2 register...
CPU01 has MSR 0xE2: 0x0000000001E008008
CPU02 has MSR 0xE2: 0x0000000001E008008
CPU03 has MSR 0xE2: 0x0000000001E008008
CPU04 has MSR 0xE2: 0x0000000001E008008
CPU05 has MSR 0xE2: 0x0000000001E008008
Done checking MSR 0xE2 register, compare the values printed!
This firmware has LOCKED MSR 0xE2 register!
```

- This tells me that the CFG Lock option is locked in my firmware. :(

Extracting the Dell BIOS

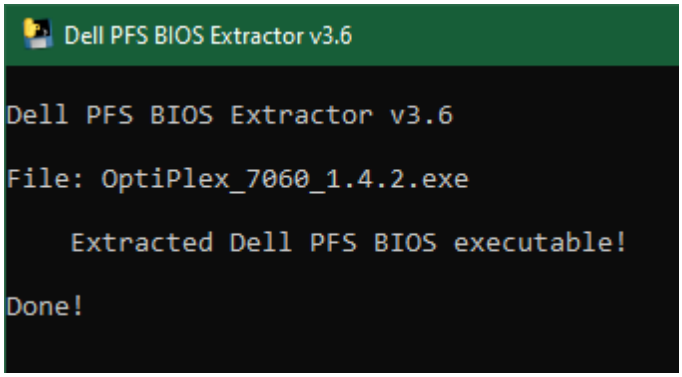
- Dell does not provide the proper BIOS files to use, so I had to dump the BIOS from the Dell installation file.
- I downloaded the latest Dell BIOS (OptiPlex_7060_1.4.2.exe) from Dell's support site and placed it into my Downloads folder.

From macOS

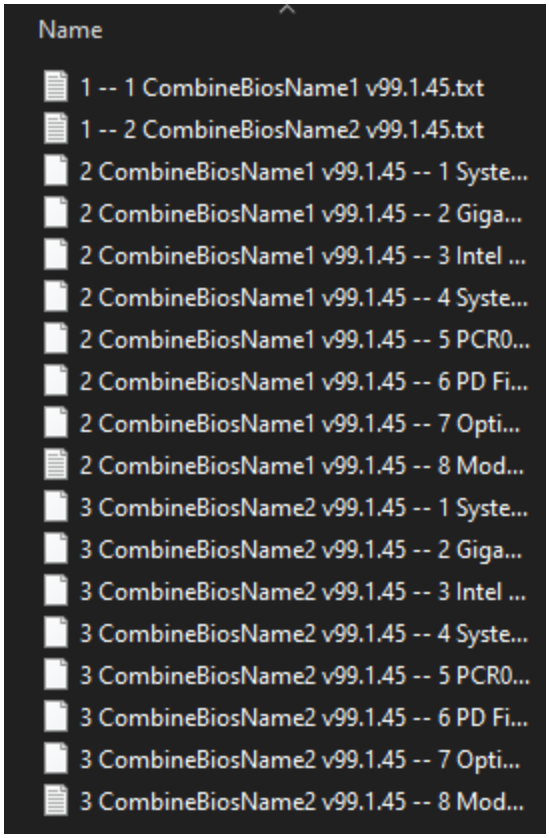
- I did this in Windows, but it can be done in macOS with Dell PFS BIOS Extractor, found in BIOSUtilities.

From Windows

- I downloaded Dell PFS Extract v3.6 and used 7-Zip to un-rar the file.
- I dragged “OptiPlex_7060_1.4.2.exe” onto “Dell_PFS_Extract.exe”.



- This created a folder called “OptiPlex_7060_1.4.2.exe_extracted” with all my needed BIOS files.



Disabling CFG Lock

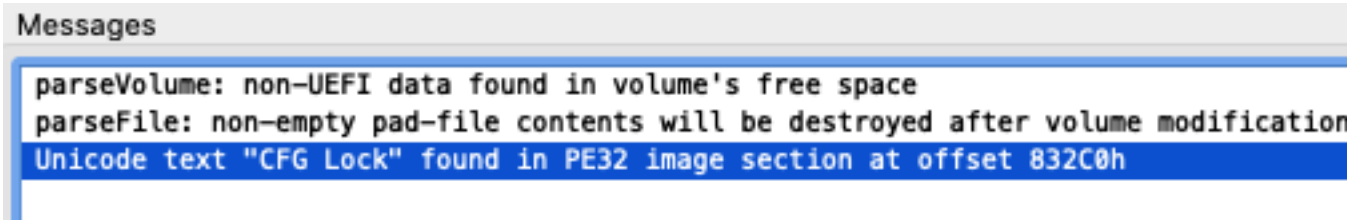
- I already setup “VerifyMsrE2.efi” in the steps listed above.
- I downloaded Modified GRUB Shell and moved it to the \EFI\OC\Tools folder on my USB.
- I added the tool to Misc => Tools in my “Config.plist” by doing an OC Snapshot in PorperTree.
- I downloaded UEFITool (UEFITool_0.28.0_mac.zip) and Universal-IFR-Extractor (ifrextract_v0.3.6.osx.zip).
- I will not disable AppleCpuPmCfgLock or AppleXcpmCfgLock in “Config.plist” until later.

Checking if CFG-Lock can be turned off

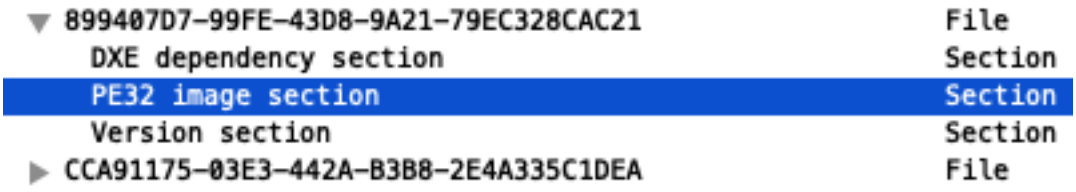
- I already completed these steps above.

Turning off CFG-Lock manually

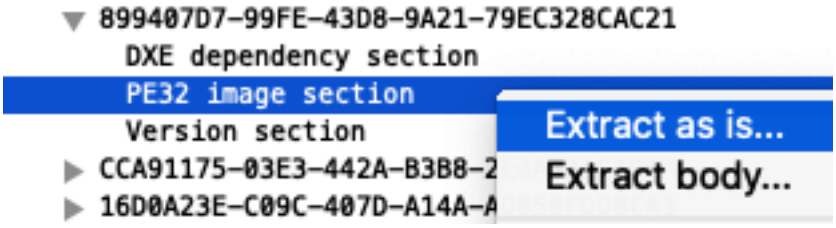
- I opened the UEFITool.
- I selected File => Open Image File… and opened “2 CombineBiosName1 v99.1.45 -- 1 System BIOS with BIOS Guard [V2] v1.4.2.bin”.
- I searched the top-level “UEFI image” using File => Search. I chose the Text option to search and entered “CFG Lock”. I received the following result:



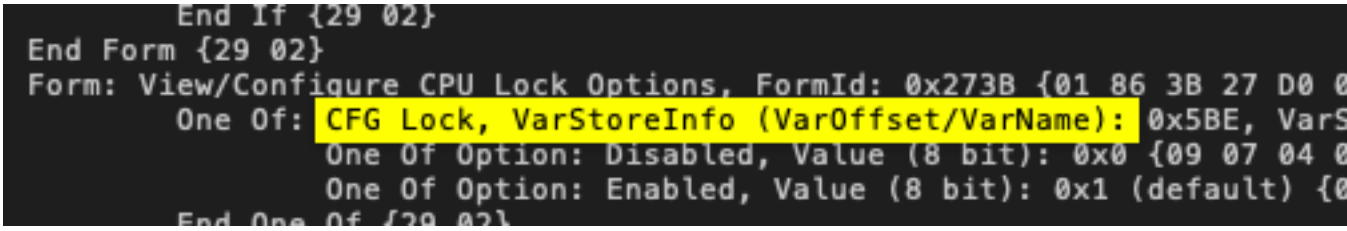
- I double-clicked the text “Unicode text ‘CFG Lock’ found in PE32 image section at offset 832C0h” and it took me to the appropriate PE32 image section:



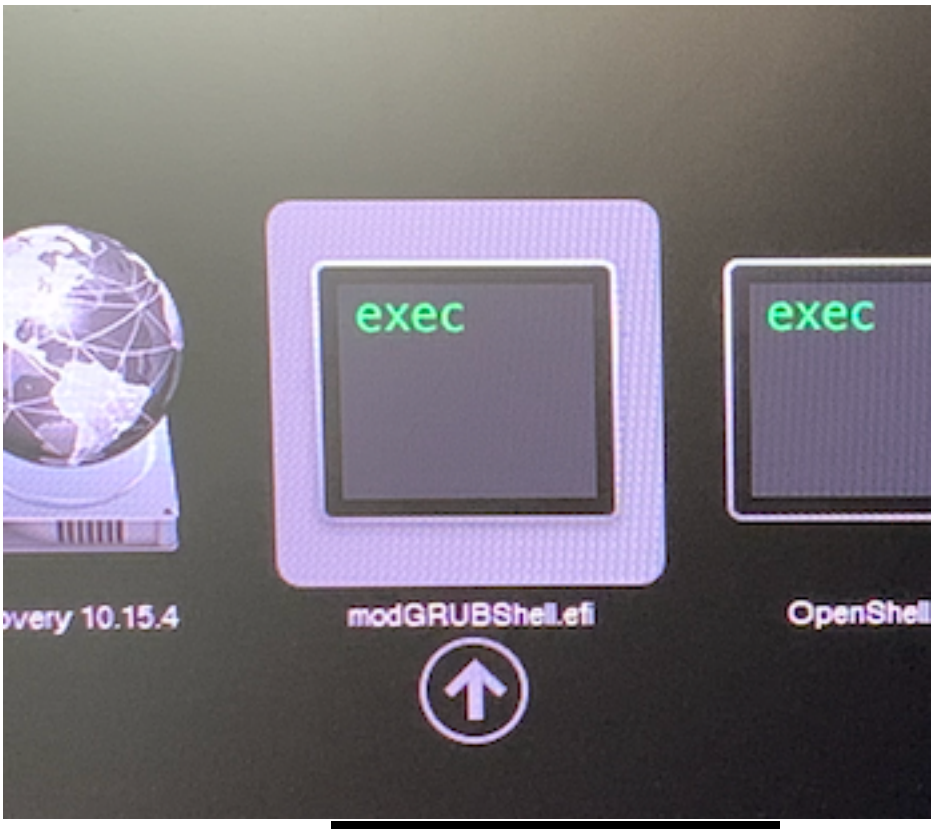
- I extracted the “PE32 image section” into the same folder as “ifrextract” and named it “CFG-Lock.bin”:



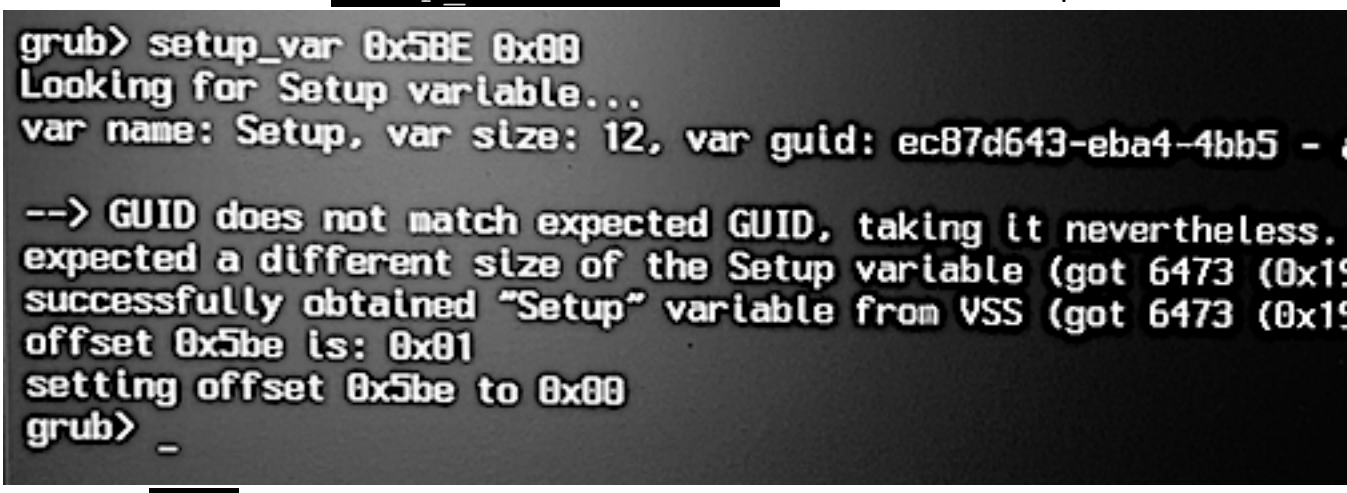
- Right-click on “ifrextract”, then hold Option, then select Open. Close the window. This has to be done ahead of time to give permission to open this program in macOS. Otherwise Terminal will prompt you with no way to give it proper permission.
- I opened Terminal and did the following:
    - Dragged and dropped “ifrextract” into the Terminal window to provide the Tool path.
    - Dragged and dropped “CF-Lock.bin” into the Terminal window to provide the Input path.
    - Dragged and dropped “CF-Lock.bin” into the Terminal window, then changed “.bin” to “.txt” to provide the Output path.
- I opened the resulting “CF-Lock.txt” file and searched for “CFG Lock, VarStoreInfo (VarOffset/VarName):”.



- My offset for CFG Lock for Dell BIOS v1.4.2 is “0x5BE”. This is entirely unique to this specific system. Read the warning in the OCD Guide:
    - “Do note that variable offsets are unique not just to each motherboard but even to its firmware version. **Never try to use an offset without checking.”**
- I opened my “Config.plist” and set AppleCpuPmCfgLock and AppleXcpmCfgLock to FALSE in Kernel => Quirks.
- I restarted my PC and selected “modGRUBShell.efi” from the Boot Picker.



- I enter the command `setup_var 0x5BE 0x00` and this is the output:



- I entered `exit` to return to the Boot Picker and selected “Reset NVRAM”.

- When the Boot Picker returned, I selected “VerifyMsrE2.efi” and this is the output:

```
Looking up EFI_MP_SERVICES_PROTOCOL...
Checking MSR 0xE2 on all CPUs. Values must be SA
CPU00 has MSR 0xE2: 0x000000001E000008
Starting All APs to verify 0xE2 register...
CPU01 has MSR 0xE2: 0x000000001E000008
CPU02 has MSR 0xE2: 0x000000001E000008
CPU03 has MSR 0xE2: 0x000000001E000008
CPU04 has MSR 0xE2: 0x000000001E000008
CPU05 has MSR 0xE2: 0x000000001E000008
Done checking MSR 0xE2 register, compare the val
This firmware has UNLOCKED MSR 0xE2 register!
```

- This tells me that the CFG Lock option is unlocked in my firmware. SUCCESS!

Cleanup

- After a successful boot I removed “VerifyMsrE2.efi” and “modGRUBShell.efi” from \EFI\OC\Tools on the USB.

- I updated my “Config.plist” by doing an OC Snapshot in PorperTree.

Fixing Resolution and Verbose

Recommended Configuration

- I followed all OCD Guide recommendations.

Fixing DRM

Fixing DRM support and iGPU performance

- “DRM is broken for iGPU-only systems”

Fixing DRM

- Not possible for this PC.

Fixing iGPU performance

- I’m not sure if this is only related to DRM, because it was said iGPUs are not supported.
- I tried adding “igfxfw” with the values suggested to “PciRoot(0x0)/Pci(0x2,0x0)”, but it didn’t seem to make a difference and so I removed it.

Fixing iServices

Generate a new Serial

- I completed this in the Intel Config.plist => Coffee Lake => Platforminfo section of the OCD Guide.
- After looking this section over, I decided to check the Apple Check Coverage page and see what my serial showed up as. My generated serial is listed as invalid. This is the safest option.

Fixing En0

- I checked my NIC with Hackingtool. I did not need this section.

Fixing ROM

- I completed this in the Intel Config.plist => Coffee Lake => Platforminfo section of the OCD Guide.

Verifying NVRAM

- I completed this in the Getting Started with ACPI => UNIVERAL => NVRAM PMC section of the OCD Guide.

Clean out old attempts

- I did not need this section.

Verifying your work one last time

- I did not need this section.

Cleaning up your AppleID

- I did not need this section.

Customer Code error

- I did not need this section.

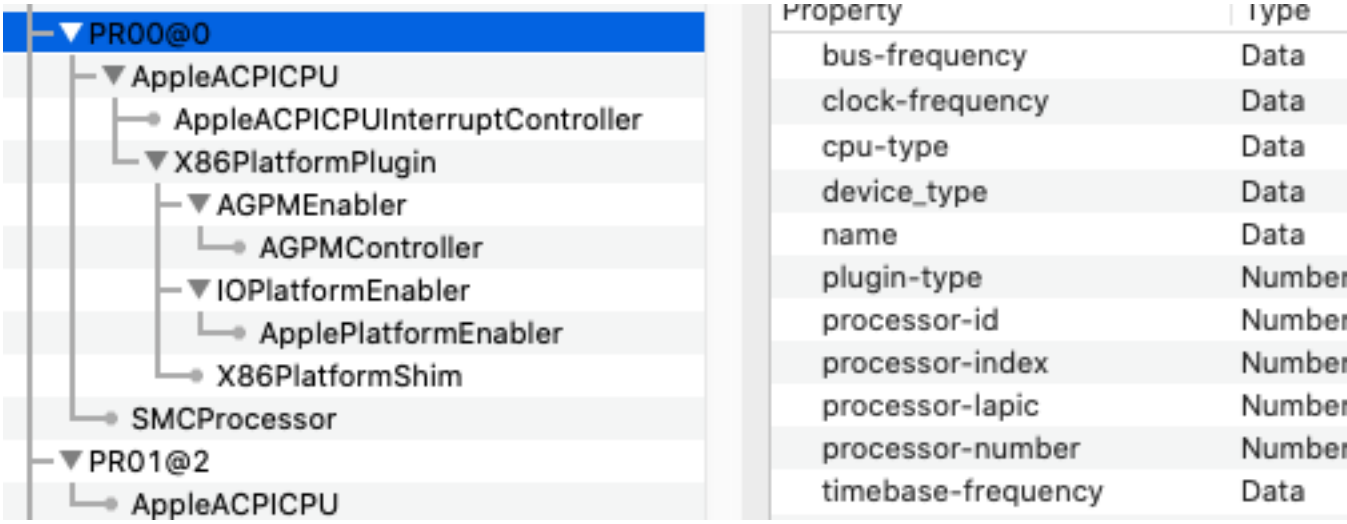
Emulating NVRAM

- I did not need this section.

Fixing Power Management

Enabling X86PlatformPlugin

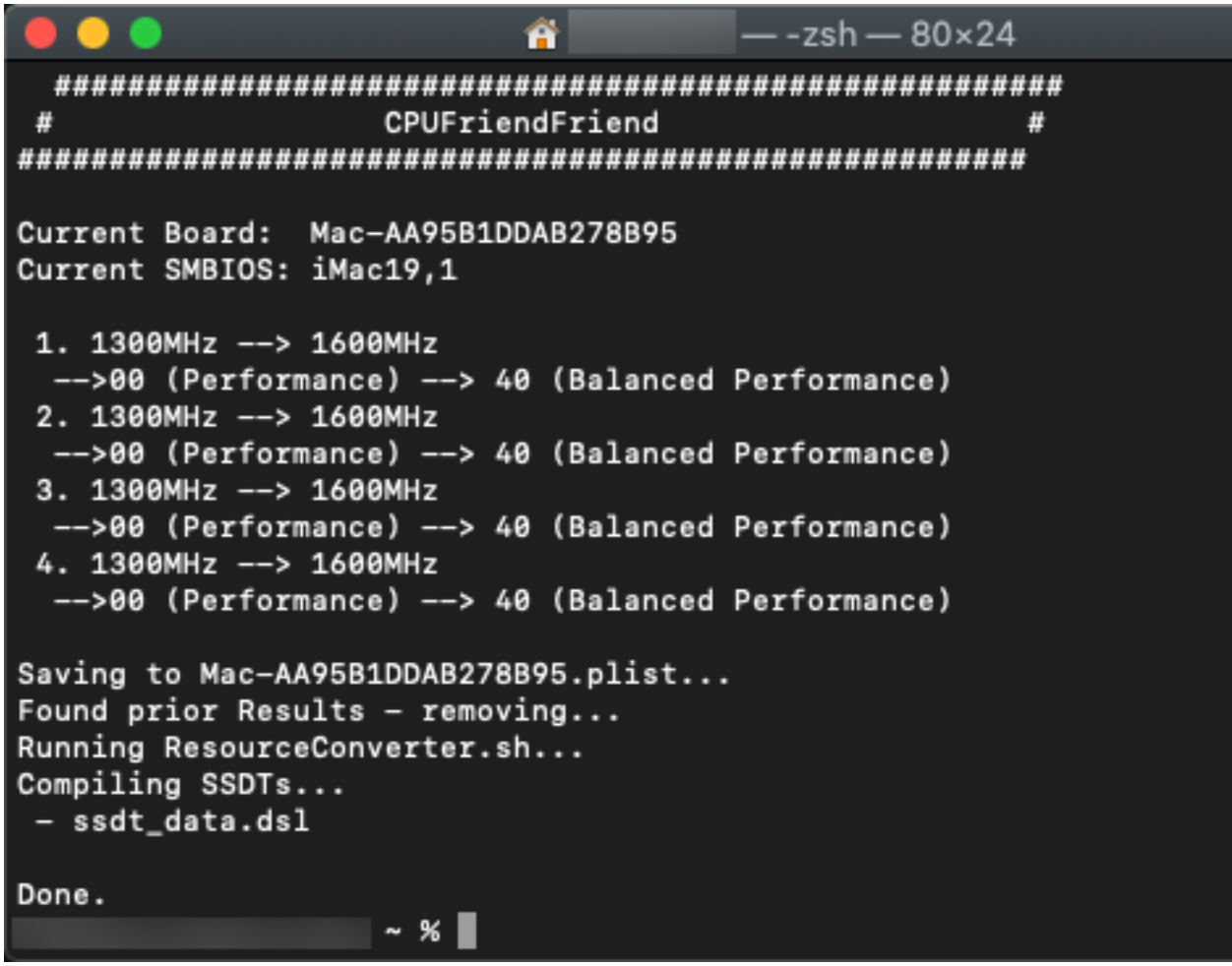
- My Power Management appears to be working correctly:

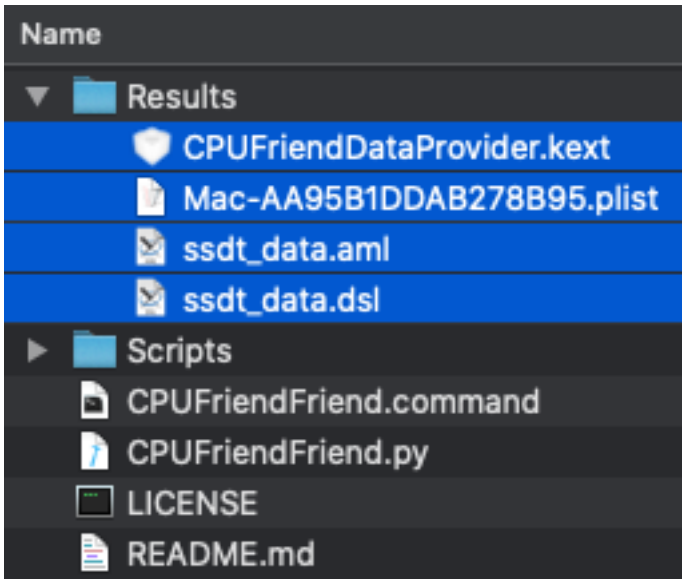


Using CPU Friend

- I downloaded CPUFriend-1.2.0-RELEASE.zip and CPUFriendFriend-master.zip.

- I went to Intel's ARK site and found my processor for reference.
- I opened Terminal and dropped “CPUFriendFriend.py” into the window.
    - Min hex freq: 10
        - Min hex freq / TDP-down Frequency = 1.6 GHz / 1600 MHz / (First 2 digits converted to Hex) 10
    - EPP: 0x40
        - I chose 0x40 for the top end of Balance Performance.
    - I repeated these steps four times, once for each frequency vector.
        - I was confused because the Guide does not mention multiple vectors, but this was done at the recommendation of this post. I cannot say I completely understand what each vector is.
    - My results:





- I moved “CPUFriend.kext” from the “CPUFriend-1.2.0-RELEASE” folder into my \EFI\OC\Kexts folder.
- I moved “CPUFriendDataProvider.kext” from the \CPUFriendFriend-master\Results folder into my \EFI\OC\Kexts folder.
- I updated my “Config.plist” by doing an OC Snapshot in ProperTree.

Fixing USB

This portion of the OCD Guide has been branched out into a separate page. Click the link below to be taken to that section in this document:
Link =>
Fixing USB <= Link

EXTRAS

Legacy Install

- I did not need this section.

Add GUI and Bootchime

Setting up OpenCore's GUI

I decided to undo these changes after discovering that I could not choose a default boot option from the GUI boot picker.

- I downloaded the latest Binary Resources, unzipped, and replaced the Resources folder on my USB with the folder from it.
- "OpenCanopy.efi" is already in place from the "Adding The Base OpenCore Files" section of the OCD Guide.
- I updated Misc => Boot => "PickerMode" and "PickerAttribute" while initially configuring the "Config.plist".
- The \UEFI\Drivers section already contains "OpenCanopy.efi" from the initial OC Clean Snapshot done in ProperTree.

Setting up Boot-chime with AudioDxe

- I did not want the Bootchime enabled.

iGPU Patching

- I did not need this section.

Fixing KASLR slide values

- I did not need this section.

Disabling unsupported GPUs

- I did not need this section.

Clover Conversion

- I did not need this section.

Getting Started with ACPI

https://dortania.github.io/Getting-Started-With-ACPI/

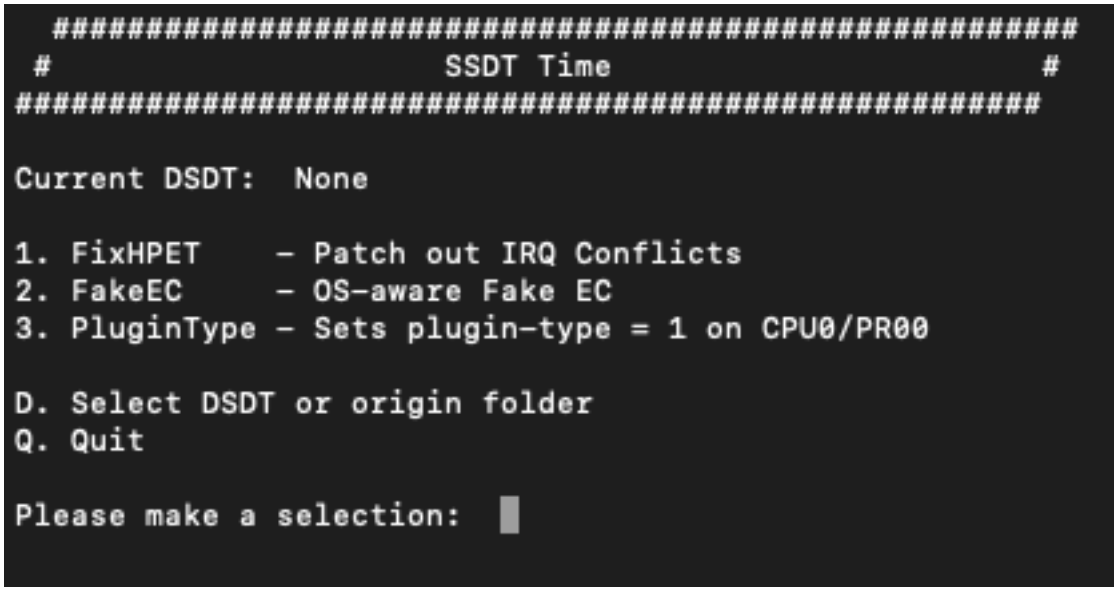
METHODS

SSDTs: Easy Way

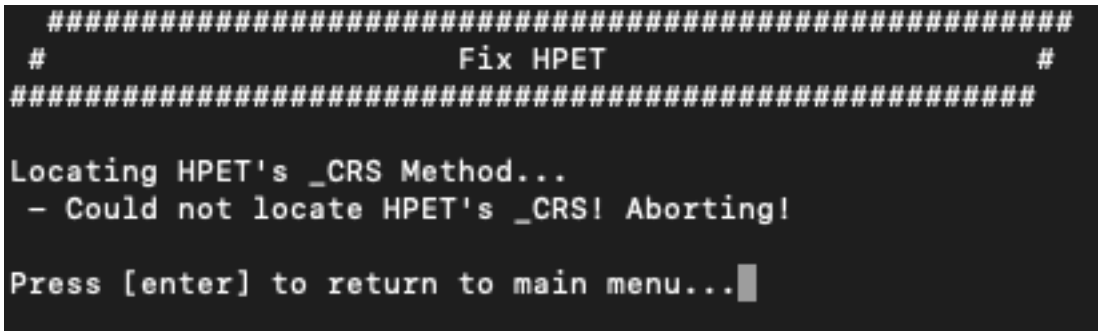
Running SSDTTime

- I used MaciASL. By default, it will open the System DSDT. I exported it as a "System DSDT.dsl" and "System DSDT.aml" using File => Save.
- I downloaded SSDTTime-master.zip.

- I downloaded iasl macOS version, unzipped it, and moved the contents into the Scripts folder of SSDTTime.
    - Right click on iasl, then hold the Option key, then select Open. Close the window. This has to be done ahead of time to give proper permission to the program in macOS; otherwise SSDTTime will prompt you and there is no way to give it proper permission. This is the case with many of the downloaded tools.
- I moved a copy of the "System DSDT.aml" file to the SSDTTime folder and renamed it to "DSDT.aml". It would not work for me with the "System DSDT" name.
- I opened SSDTTime and performed the following:

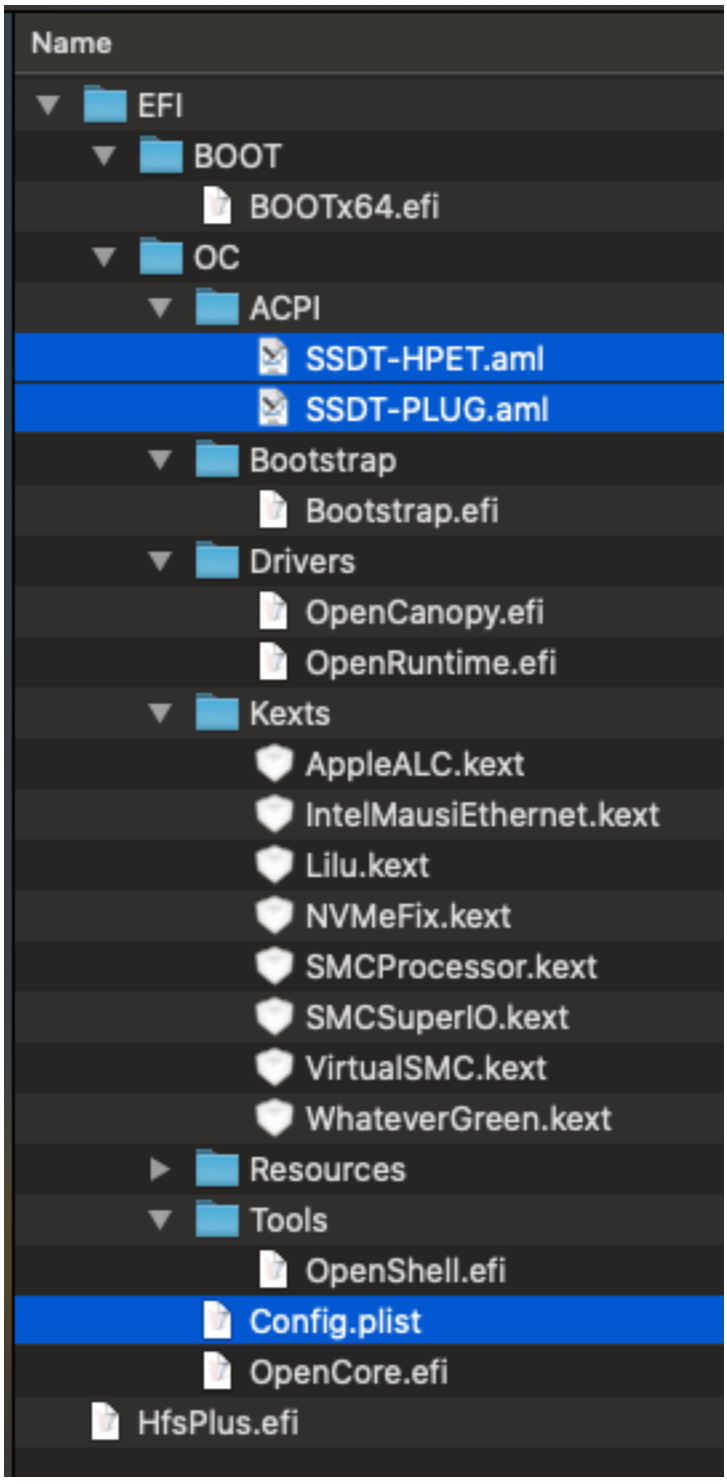


    - I selected "D" and dropped the "DSDT.aml" file into the Terminal window to load it.
    - I selected "1" to create "SSDT-HPET.aml" and "patches_OC.plist". I found out during my first install that this SSDT and the patch are needed to enable my audio controller.
        - I get the following error when trying to do this in macOS. I had to use the SSDT I created in Windows. I am not sure why this would not work. I also noticed that the option to select "C" for leaving out legacy patches is not an option.



- I copied “Sample.plist” from the \OpenCore-0.5.8-RELEASE\Docs folder and placed it into the \EFI\OC folder on the USB.
- I renamed “Sample.plist” to “Config.plist”.
- I used the steps in the OCD Guide to patch my new “Config.plist” using “patches_OC.plist”:
    - Open both files,
    - Delete the ACPI => Patch section from “Config.plist”
    - Copy the ACPI => Patch section from “patches_OC.plist”
    - Paste into where old patches were in “Config.plist”

- Once that was complete, I selected “3” to create “SSDT-PLUG.aml” and moved it into the \EFI\OC\ACPI folder on the USB. My EFI folder now looks like this:



MANUAL

Dumping the DSDT

- I have done this using SSDTTime in Windows and MaciASL in macOS. Both are pretty straightforward.

Decompiling and Compiling

- I have done this using iasl.exe and MaciASL. Again, both are pretty easy to do.

DESKTOP

Disabling desktop dGPUs

- This is not needed for my PC; it only has an iGPU.

LAPTOP

- Obviously, none of this is needed for this desktop PC.

UNIVERSAL

Embedded Controller

Fixing Embedded Controller

- I grabbed the SSDT-EC-USBX.dsl RAW data, copied it to TextEdit, and saved it as a "SSDT-EC-USBX.dsl".

Fixing the Path

- When searching my DSDT for "PNP0C09" I get no results. See the section below for more on that.

Disabling real EC (Desktops only)

- No need for this step and "uncommenting" because of the next section. Read on!

What happens if no PNP0C09 show up

- "This means you only need to fake an Embedded Controller rather so that AppleBusPowerController will load and handle USB power properly and continue booting. To make the actual SSDT, its almost plug and play as no uncommenting needed. The main thing that needs to be changed: LowPinCount path and PCI path."
    - I searched my DSDT using MaciASL for "Name (_ADR, 0x001F0000)".
        - As seen below, my LowPinCount path is "LPCB".

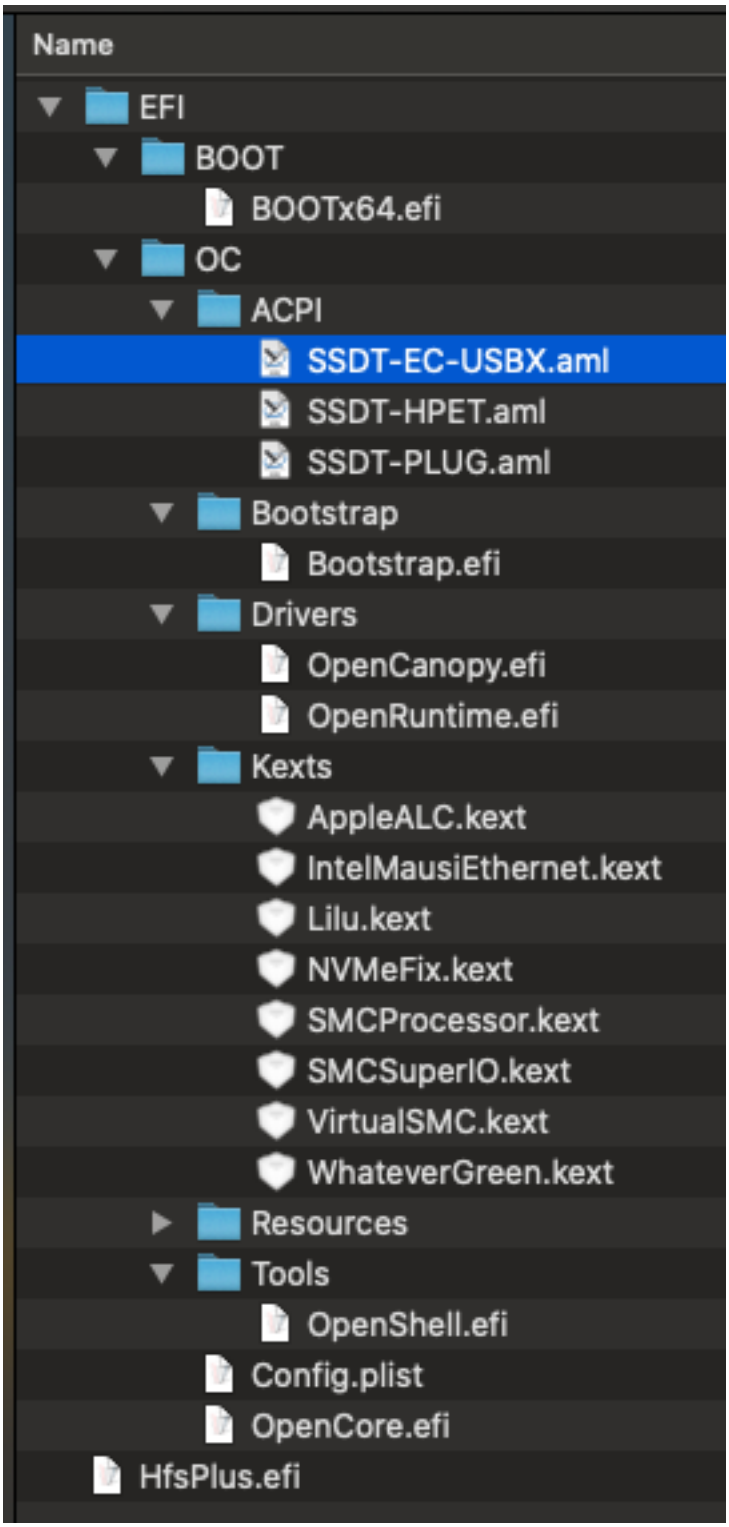
```
Device (LPCB)
{
    Name (_ADR, 0x001F0000)  // _ADR: Address
    Method (SPTS, 1, NotSerialized)
    {
```

- I searched my DSDT using MaciASL for “PNP0A08”.
  - As seen below, my PCI path is “PCI0”.

```
Device (PCI0)
{
    Name (_HID, EisaId ("PNP0A08") /* PCI Express Bus */)  // _H
    Name (_CID, EisaId ("PNP0A03") /* PCI Bus */)  // _CID: Comp
    Name (_ADR, Zero)  // _ADR: Address
    Method (^BN00, 0, NotSerialized)
    {
```

  - Both of these are the default values for the data in “SSDT-EC-USBX.dsl” so I did not need to change it.

- I opened “SSDT-EC-USBX.sdsl” in MaciASL and saved it as “SSDT-EC-USBX.aml”.
- I moved “SSDT-EC-USBX.aml” to the \EFI\OC\ACPI folder on my USB. My EFI folder now looks like this:

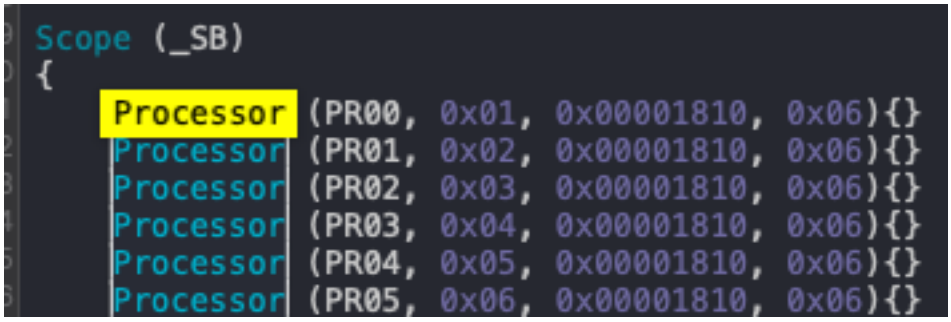


Correcting USB Power

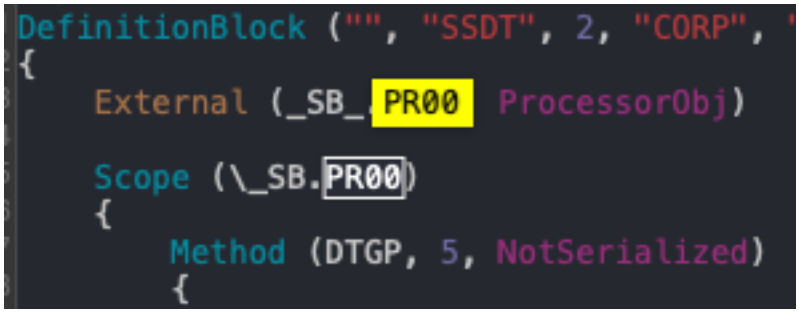
- This was already taken care of with the steps above.

Plugin type

- I used SSDTTime to create “SSDT-PLUG.aml”, but I double-checked the information in this step.
    - First, I searched the DSDT for “Processor” to figure out my processor name. Mine is “PR00”.



    - Then I searched my “SSDT-PLUG.aml” for the correct naming convention listed in the OCD Guide.



    - Everything was as expected.

AWAC vs RTC

Determining which SSDT you need

- I searched my DSDT for “Device (AWAC)” and just “AWAC” to be sure. No results were found so there is no need for either of these SSDTs.

NVRAM PMC

- The description in the SSDT on GitHub said the need for this started with the Z390 chipset. My chipset, the Q370, was released right before the Z390. I’m going to leave this SSDT out for now and then test my NVRAM using the “Post Install => Emulated NVRAM” section of the OCD Guide.
    - I rebooted, cleared the NVRAM in the Boot Picker, and ran these commands in Terminal:

```
[root@Mac-7060 ~ # sudo nvram -c
nvram: Error clearing firmware variables: (iokit/common) not permi
[root@Mac-7060 ~ # sudo nvram myvar=test01
[root@Mac-7060 ~ # exit
```

    - I received an error when trying to clear the NVRAM, but I think that's because I just did it in the Boot Picker. I'll try the rest and see if the custom variable stays.
  - I restarted the PC, logged back in, and ran the next command:

```
[root@Mac-7060 ~ # nvram -p | grep -i myvar
myvar   test01
```

  - Based on my test it seems that I do not need "SSDT-PMC.aml".

IRQ Fix

- I completed this in SSDTTime when I selected "FixHPET" and created "SSDT-HPET.aml" and "patches_OC.plist". This enables use of the audio controller on my PC.

GPU Spoof

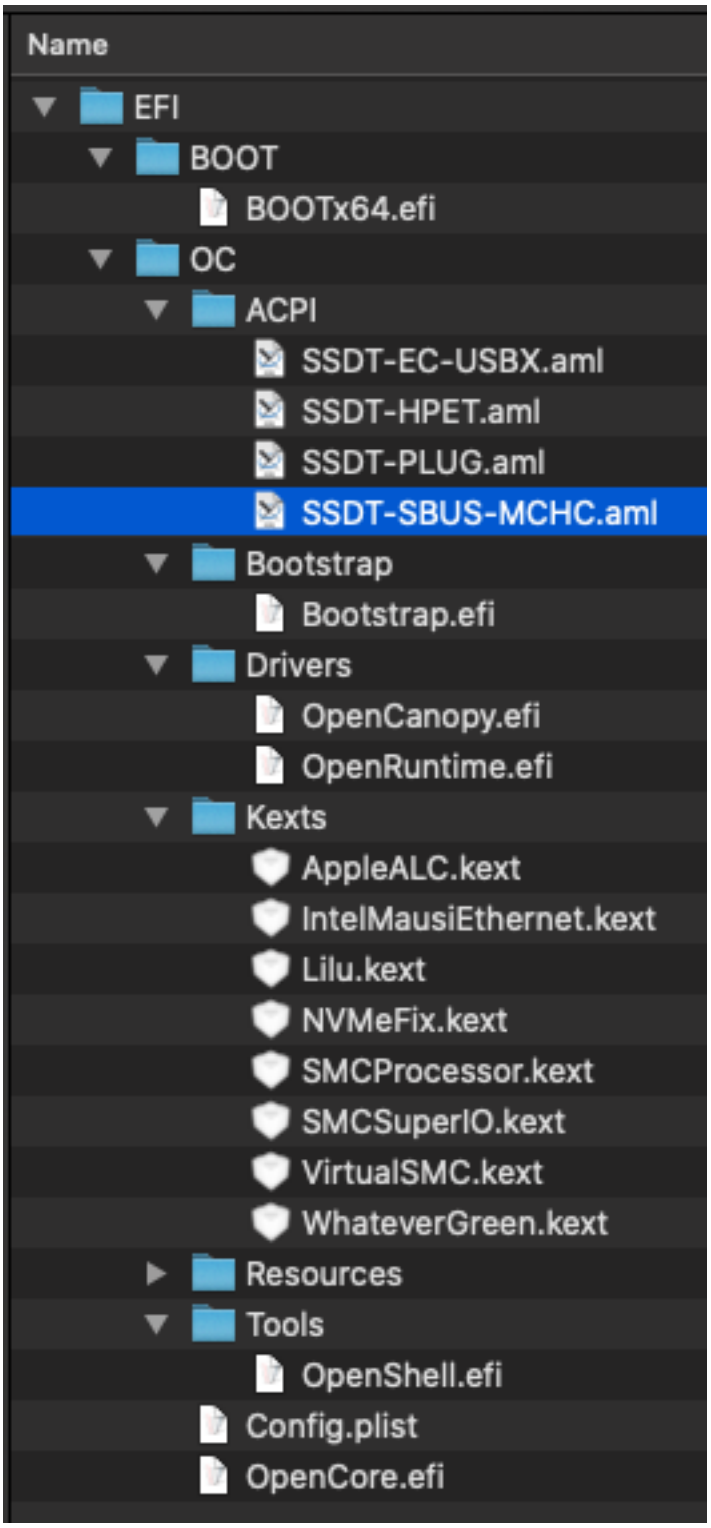
- This is not needed for my PC; it only has an iGPU.

Fixing SMBus Support

- I grabbed the SSDT-SMBUS-MCHC.dsl RAW data, copied it to TextEdit, saved it as a "SSDT-SMBUS-MCHC.dsl", opened it in MaciASL, and saved it as ""SSDT-SMBUS-MCHC.aml".
- I opened Hackintool (v3.4.0) and navigated to the PCIe section to find my PC's SMBus path.

Audio devi...	/PCI0@0/HDEF@1F,3	pci8086,a348	PciRoot(0x0)/Pci(0x1F,0x3
SMBus	/PCI0@0/SBUS@1F,4	pci8086,a323	PciRoot(0x0)/Pci(0x1F,0x4
???	...0/pci8086,a324@1F,5	pci8086,a324	PciRoot(0x0)/Pci(0x1F,0x5

- As recommended in the OCD Guide, I used the beginning four characters of each section to produce "PCI0.SBUS".
- I opened "SSDT-SBUS-MCHC.aml" and see that the default values are the same, so I made no changes.
- I moved "SSDT-SBUS-MCHC.aml" to the \EFI\OC\ACPI folder on my USB. My EFI folder now looks like this:


Name
EFI
BOOT
BOOTx64.efi
OC
ACPI
SSDT-EC-USBX.aml
SSDT-HPET.aml
SSDT-PLUG.aml
SSDT-SBUS-MCHC.aml
Bootstrap
Bootstrap.efi
Drivers
OpenCanopy.efi
OpenRuntime.efi
Kexts
AppleALC.kext
IntelMausiEthernet.kext
Lilu.kext
NVMeFix.kext
SMCProcessor.kext
SMCSuperIO.kext
VirtualSMC.kext
WhateverGreen.kext
Resources
Tools
OpenShell.efi
Config.plist
OpenCore.efi

CLEANUP

Cleanup

- I skipped this as I have been moving the files over as I go. The rest will be taken care of in the next section.

This is the end of the branch away from the OCD Guide. Click the link below to be taken back to the next logical section of the guide.
Link => INTEL CONFIG.PLIST <= Link

Fixing USB

https://dortania.github.io/USB-Map-Guide/

I could not get this method to produce a USB Map that functioned for me. I used the "USB Mapping for OpenCore" article below this one to better understand manually building a USB Map. In the end, I used Hackintool and "The New Beginner's Guide to USB Port Configuration" to create a successful map. I will leave these steps in case I need them later.

System Preparation

- I do not need USBInjectAll because of my chipset.
- Using SMBIOS iMac 19,1 means I do not need an ACPI rename.

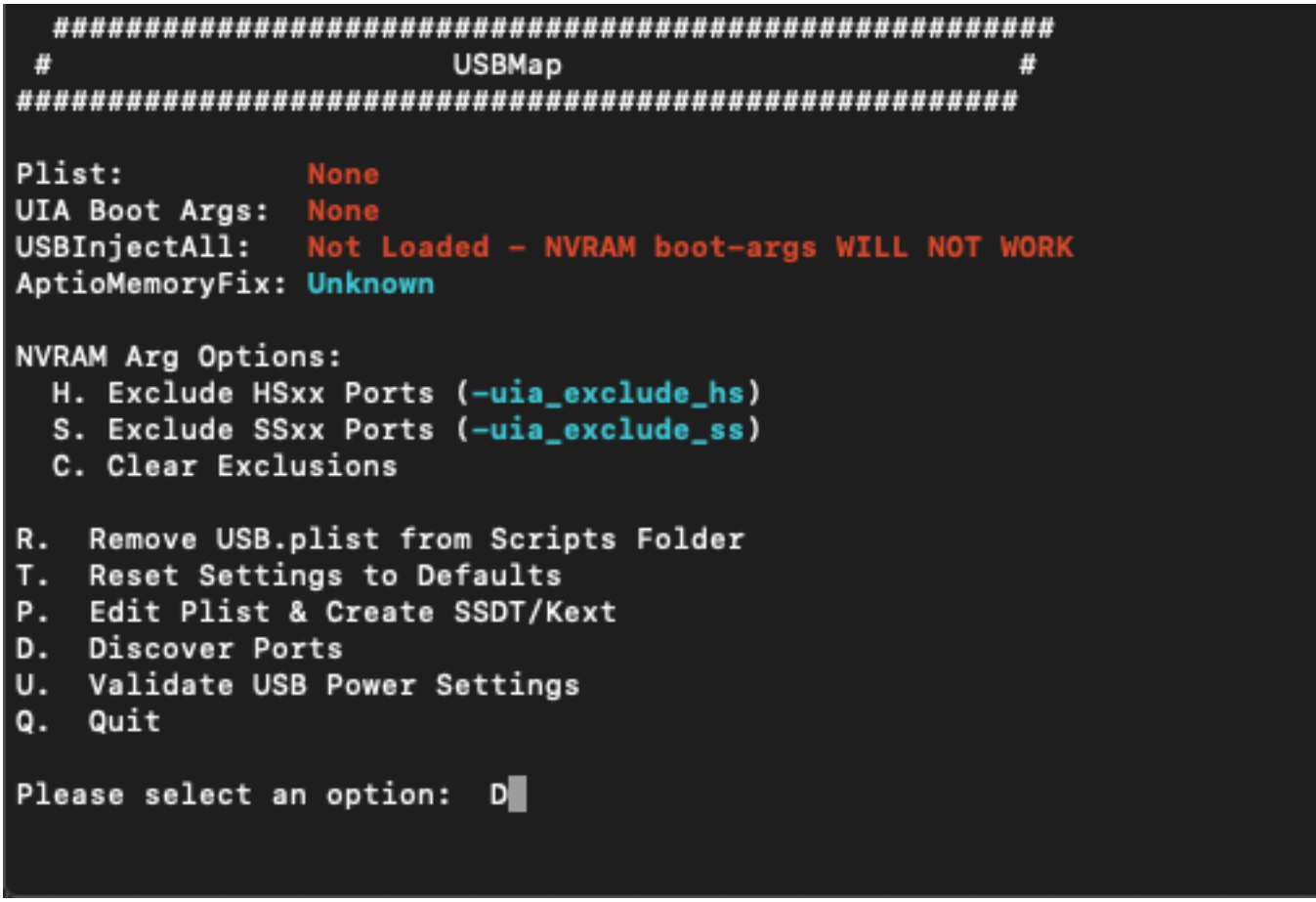
Intel USB Mapping

- I copied this handy table for reference.

Type	Info	Comments
0	USB 2.0 Type-A connector	This is what macOS will default all ports to when no map is present
3	USB 3.0 Type-A connector	3.0, 3.1 and 3.2 ports share the same Type
8	Type C connector - USB 2.0-only	Mainly seen in phones
9	Type C connector - USB 2.0 and USB 3.0 with Switch	Flipping the device **does not** change the ACPI port

10	Type C connector - USB 2.0 and USB 3.0 without Switch	Flipping the device **does** change the ACPI port. generally seen on 3.1/2 motherboard headers
255	Proprietary connector	For Internal USB ports like Bluetooth

- I downloaded the USBMap-master (7eb15fd).
- I read through the "Using The Script" section on GitHub. I did not go through the steps of hiding SS ports and then HS ports. This is because all of my ports show up without the need to do so.
- I opened "USBMap.command" and selected option D:



- I used my Logitech receiver as a USB 2.0 device, a USB 3.0 flash drive, and a USB-C hub in combination with the two others to map out all possible options. The WLAN card I have supports Bluetooth and so that shows up as a USB device as well. Without the port map, Bluetooth does not function on my system.
- I created and filled out the table below:

Logical Name	Port Address	Physical Location	Port Type #	Port Type Style	Port Type Speed
HS01	**One**	**Front**	**0**	**USB-A**	**USB 2.0**
HS02	**0x02**	**Front**	**9**	**USB-C with switch**	**USB 2.0**
HS03	0x03				
HS04	0x04				
HS05	**0x05**	**Back Left Top**	**0**	**USB-A**	**USB 2.0**
HS06	**0x06**	**Back Left Bottom**	**0**	**USB-A**	**USB 2.0**
HS07	**0x07**	**Back Right Top**	**0**	**USB-A**	**USB 2.0**
HS08	**0x08**	**Back Right Bottom**	**0**	**USB-A**	**USB 2.0**
HS09	0x09				
HS10	0x0A				
USR1	0, NotSerialized Zero				
USR2	0, NotSerialized One				
SS01	**0, NotSerialized Zero**	**Front**	**3**	**USB-A**	**USB 3.0**
SS02	**0, NotSerialized One**	**Front**	**9**	**USB-C with switch**	**USB 3.0**
SS03	0, NotSerialized 0x02				
SS04	0, NotSerialized 0x03				
SS05	**0, NotSerialized**	**Back Left Top**	**3**	**USB-A**	**USB 3.0**

	0x04				
SS06	**0, NotSerialized 0x05**	**Back Left Bottom**	**3**	**USB-A**	**USB 3.0**
HS11	0x0B				
HS12	0X0C				
HS13	0X0D				
HS14	**0X0E**	**Internal**	**255**	**Bluetooth**	**USB 2.0**
SS07	**0, NotSerialized 0x06**	**Back Right Top**	**3**	**USB-A**	**USB 3.0**
SS08	**0, NotSerialized 0x07**	**Back Right Bottom**	**3**	**USB-A**	**USB 3.0**
SS09	0, NotSerialized 0x08				
SS10	0, NotSerialized 0x09				

- With everything unplugged except my Logitech receiver and the internal WLAN card, these are my results:

```
 #                  Detecting Ports                    #
##########################################################

1. HS01 - Controller XHC
2. HS02 - Controller XHC
3. HS05 - Controller XHC
    - USB Receiver
4. HS06 - Controller XHC
5. HS07 - Controller XHC
6. HS08 - Controller XHC
7. HS14 - Controller XHC
    - BRCM20702 Hub
        - Bluetooth USB Host Controller
8. SS01 - Controller XHC
9. SS02 - Controller XHC
10. SS05 - Controller XHC
11. SS06 - Controller XHC
12. SS07 - Controller XHC
13. SS08 - Controller XHC

Populated:  XHC:13

Press Q then [enter] to stop
Press N then [enter] to add a custom name to SS08

Waiting 5 seconds:
```

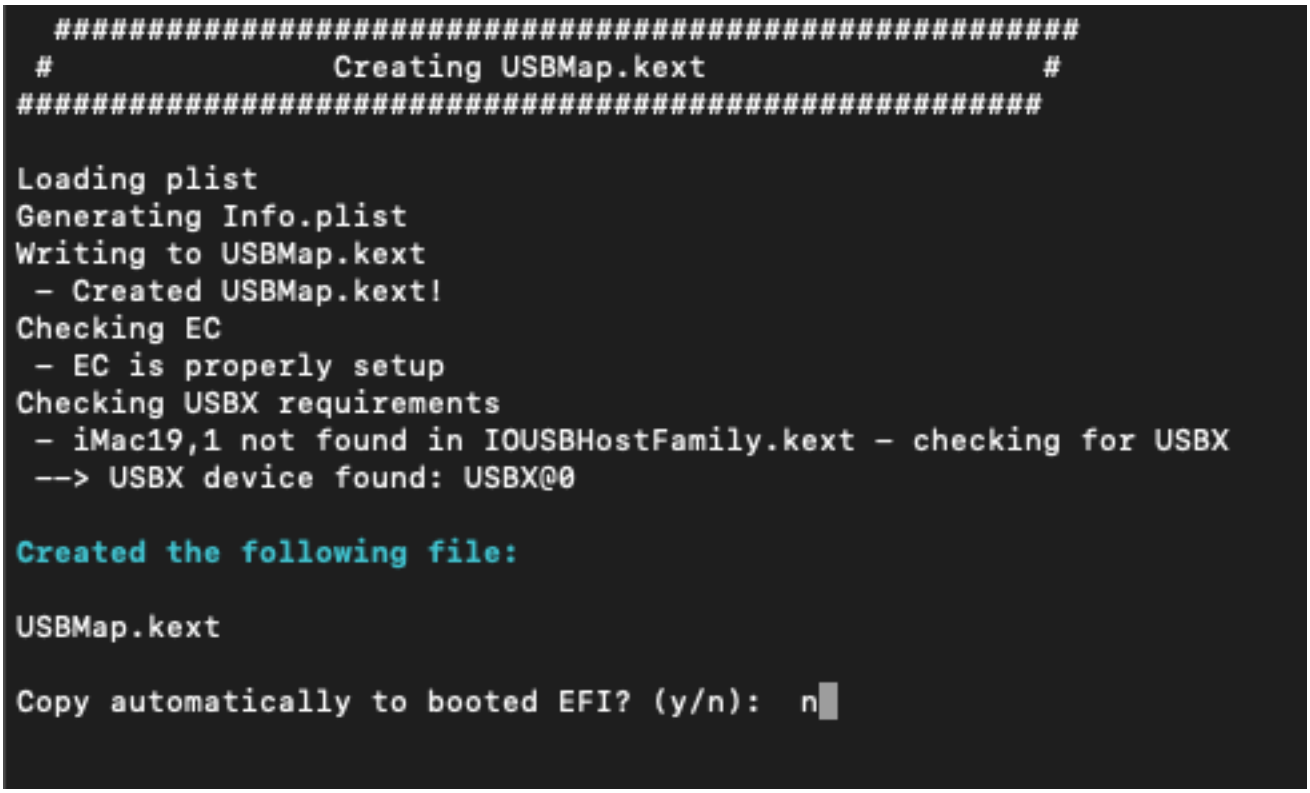
- I typed “Q” and pressed “enter” to end the discovery.
- I selected option “P” to create the files and this is my initial map:

```
[#] 1. HS01 - Type 3 - Controller XHC
    - USB Receiver
[#] 2. HS02 - Type 3 - Controller XHC
    - USB2.0 Hub
        - USB Receiver
[#] 3. HS05 - Type 3 - Controller XHC
    - USB Receiver
[#] 4. HS06 - Type 3 - Controller XHC
    - iPad
    - USB Receiver
[#] 5. HS07 - Type 3 - Controller XHC
    - Cruzer
    - USB Receiver
[#] 6. HS08 - Type 3 - Controller XHC
    - USB Receiver
[#] 7. HS14 - Type 3 - Controller XHC
    - BRCM20702 Hub
        - Bluetooth USB Host Controller
[#] 8. SS01 - Type 3 - Controller XHC
    - Patriot Memory
[#] 9. SS02 - Type 3 - Controller XHC
    - USB3.0 Hub
        - Patriot Memory
[#] 10. SS05 - Type 3 - Controller XHC
    - Patriot Memory
[#] 11. SS06 - Type 3 - Controller XHC
    - Patriot Memory
[#] 12. SS07 - Type 3 - Controller XHC
    - Patriot Memory
[#] 13. SS08 - Type 3 - Controller XHC
    - Patriot Memory
```

- All of my Type-A ports are USB 3.0 (which also supports USB 2.0). So, I'm not sure if I mark the Type for all of them to be "3", or if I have to distinguish Type 0 for the ones labeled HS. I will try marking them all as Type 3 first, since it looks like that is what the OCD Guide is doing.
- All ports are already marked as Type 3, so I left them that way to start.
- I entered `T:2,9:9` to mark HS02 and SS02 as Type C connector - USB 2.0 and USB 3.0 with Switch.
- I entered `T:7:255` to mark HS14 as Proprietary connector.
- My final map looks like this:

```
[#] 1. HS01 - Type 3 - Controller XHC
    - USB Receiver
[#] 2. HS02 - Type 9 - Controller XHC
    - USB2.0 Hub
        - USB Receiver
[#] 3. HS05 - Type 3 - Controller XHC
    - USB Receiver
[#] 4. HS06 - Type 3 - Controller XHC
    - iPad
    - USB Receiver
[#] 5. HS07 - Type 3 - Controller XHC
    - Cruzer
    - USB Receiver
[#] 6. HS08 - Type 3 - Controller XHC
    - USB Receiver
[#] 7. HS14 - Type 255 - Controller XHC
    - BRCM20702 Hub
        - Bluetooth USB Host Controller
[#] 8. SS01 - Type 3 - Controller XHC
    - Patriot Memory
[#] 9. SS02 - Type 9 - Controller XHC
    - USB3.0 Hub
        - Patriot Memory
[#] 10. SS05 - Type 3 - Controller XHC
    - Patriot Memory
[#] 11. SS06 - Type 3 - Controller XHC
    - Patriot Memory
[#] 12. SS07 - Type 3 - Controller XHC
    - Patriot Memory
[#] 13. SS08 - Type 3 - Controller XHC
    - Patriot Memory
```

- I selected option “K” to create the “USBMap.kext”.
- I opted to move the Kext myself.



- I moved "USBMap.kext" from the \USBMap-master\Results folder into my \EFI\OC\Kexts folder.
- I updated my "Config.plist" by doing an OC Snapshot in ProperTree.
- This Kext did not work properly. My Bluetooth was not available, and ports did not recognize USB 3.0 devices.
- I tried recreating the Kext using `T:1,3,4,5,6:0` to make the HS ports Type 0, but this also failed to work.
- In the end I decided to use the guide below called, "USB Mapping for OpenCore". That worked for me.

AMD/3rd Part USB Mapping

- I do not need this section.

Miscellaneous Fixes

Fixing USB Power

- I completed this in the Getting Started with ACPI => UNIVERAL => Embedded Controller section of the OCD Guide.

Fixing Shutdown/Restart

- I do not need this section.

GPRW/UPRW/LANC Instant Wake Patch

- I did not need this section.

Keyboard Wake Issues

- I did not need this section.

USB Mapping for OpenCore

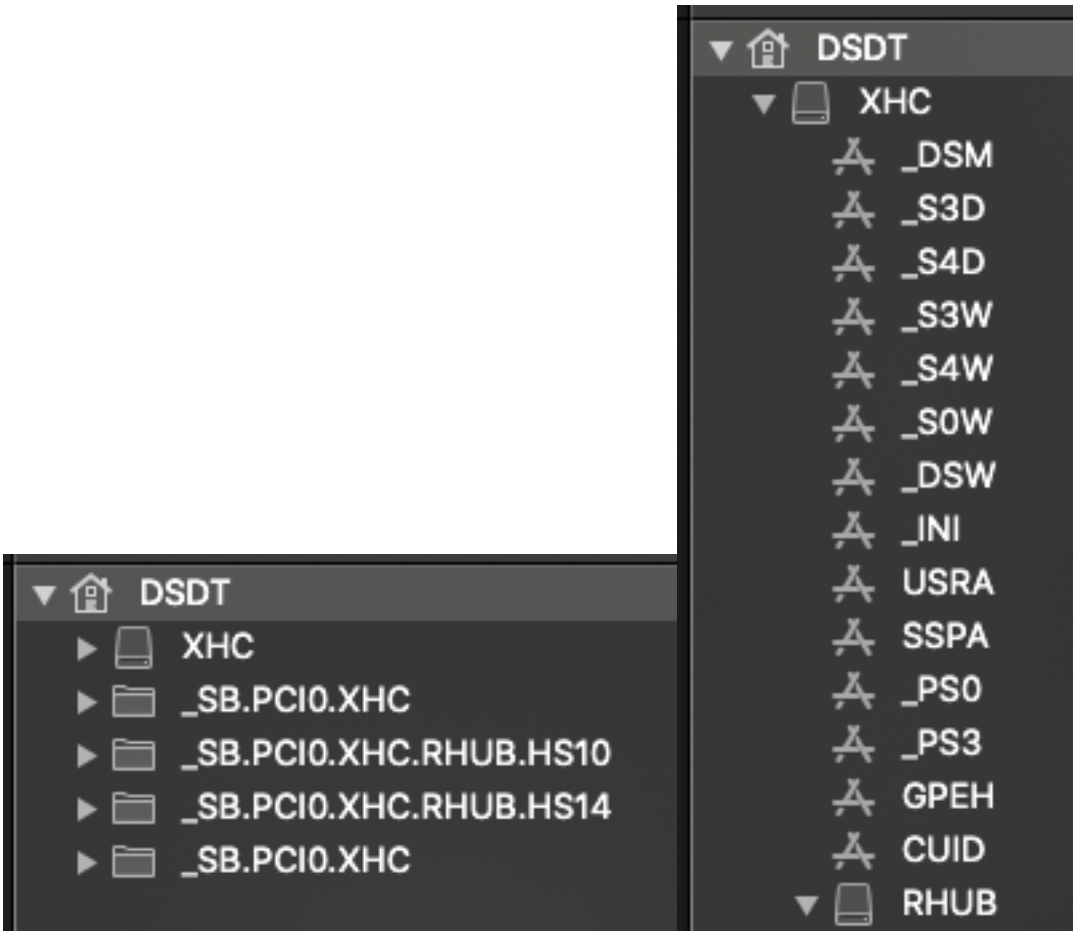
As mentioned above, this helped to understand what's going on but not to produce a Kext file. In the end, I used Hackintool and "The New Beginner's Guide to USB Port Configuration" to create a successful map. I will leave these steps in case I need them later.

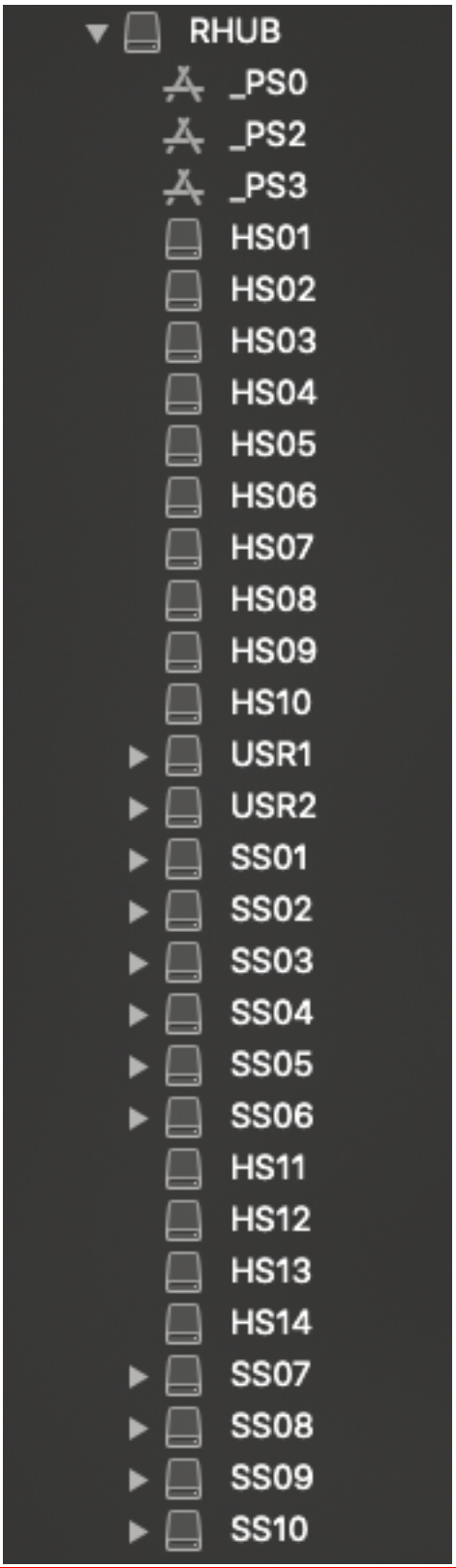
Why you need to care about USB mapping

https://aplus.rs/2020/usb-mapping-why/

Know your hardware

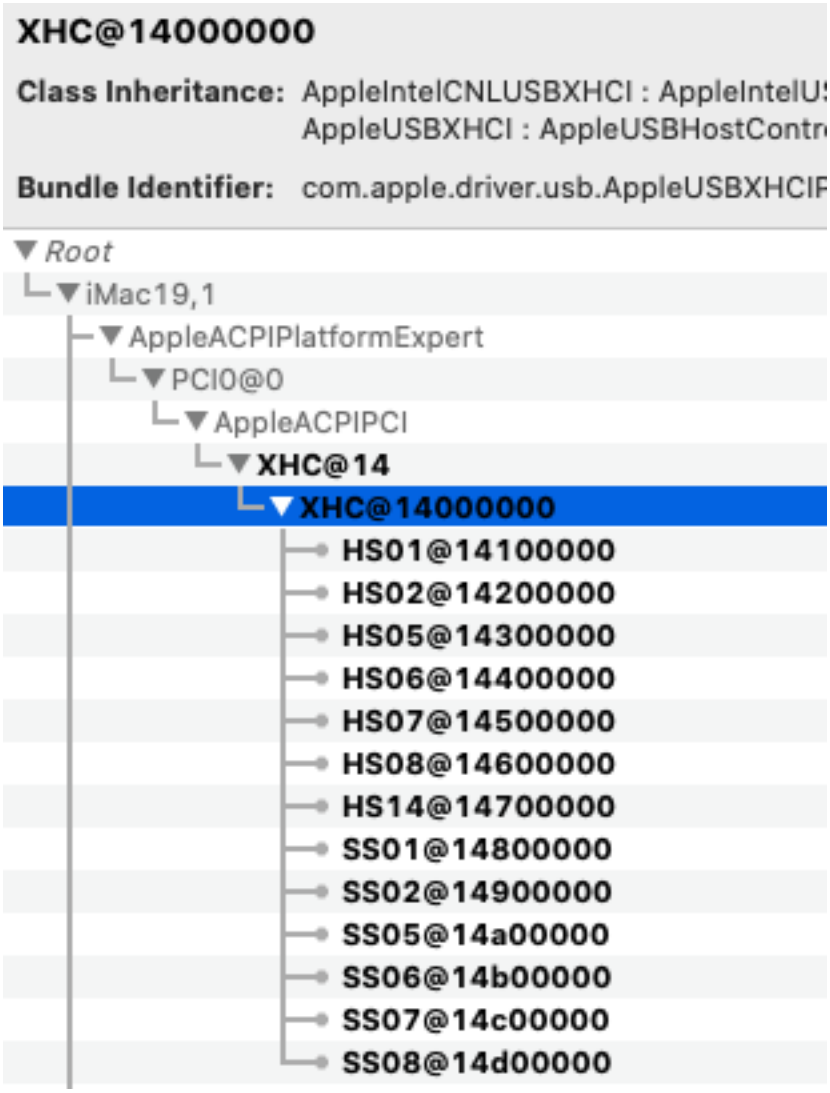
- Based on this section, I used MaciASL to determine that my PC has one USB controller named "XHC".
    - 1 Controller "XHC"
    - 14 HS Logical Ports (USB 2)
    - 10 SS Logical Ports (USB 3)
    - 2 USR Logical Ports (Unknown)


DSDT
XHC
_SB.PCI0.XHC
_SB.PCI0.XHC.RHUB.HS10
_SB.PCI0.XHC.RHUB.HS14
_SB.PCI0.XHC
DSDT
XHC
_DSM
_S3D
_S4D
_S3W
_S4W
_S0W
_DSW
_INI
USRA
SSPA
_PS0
_PS3
GPEH
CUID
RHUB



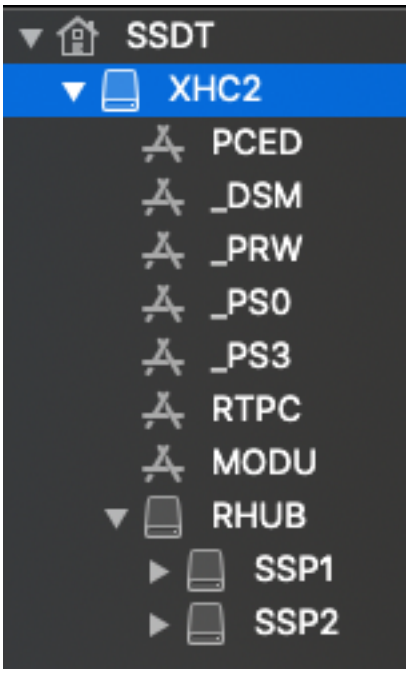
- I already have a problem, because that's 26 logical ports on one controller and macOS caps the amount of each controller at 15.

- To see how my system is mapping the USB ports without “XhciPortLimit” turned on or a USB Mapping kext, I searched IORegistryExplorer:

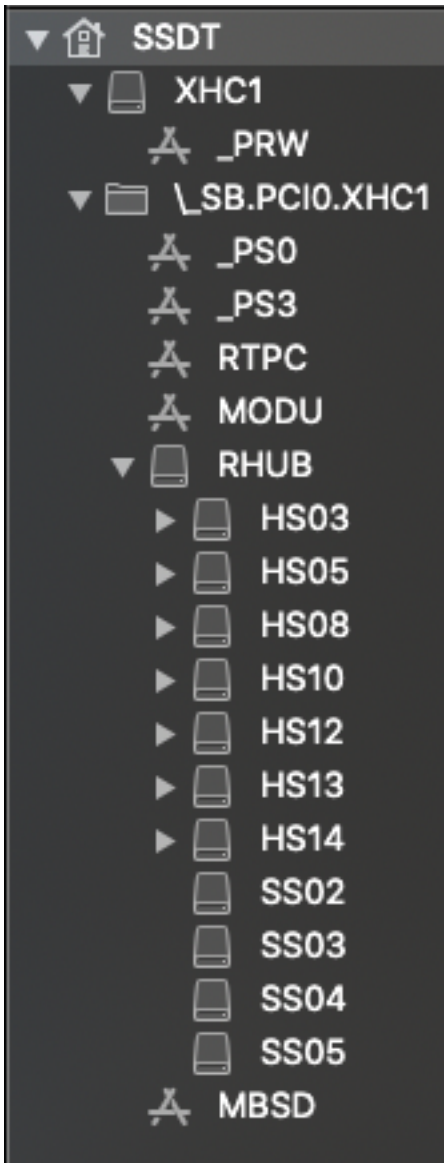


Know what you emulate

- I downloaded the DSDT dump from this source for my iMac 19,1 setup.
- I browsed through the SSDTs in \ACPI Tables\AML
  - “SSDT.aml” and “SSDT-1.aml” through “SSDT-9.aml” contained no references to “XHC”.
  - “SSDT-10.aml” has a controller named “XHC2” with 2 ports.



- “SSDT-11.aml” has a controller named “XHC1” with 11 ports .



How-to: USB mapping for OpenCore

https://aplus.rs/2020/usb-mapping-how/

Building the map

- I used IORegExplorer and multiple USB devices to plug-in and produce a map. Slowly I filled in all of the details I could, using the default mapping, with Kernel => Quirk => XhciPortLimit set to “False” in “Config.plist”.

Logical Name	Port Address	Physical Location	Port Type #	Port Type Style	Port Type Speed
HS01	One	Front	0	USB-A	USB 2.0

HS02	0x02	Front	9	USB-C with switch	USB 2.0
HS03	0x03				
HS04	0x04				
HS05	0x05	Back Left Top	0	USB-A	USB 2.0
HS06	0x06	Back Left Bottom	0	USB-A	USB 2.0
HS07	0x07	Back Right Top	0	USB-A	USB 2.0
HS08	0x08	Back Right Bottom	0	USB-A	USB 2.0
HS09	0x09				
HS10	0x0A				
USR1	0, NotSerialized Zero				
USR2	0, NotSerialized One				
SS01	0, NotSerialized Zero	Front	3	USB-A	USB 3.0
SS02	0, NotSerialized One	Front	9	USB-C with switch	USB 3.0
SS03	0, NotSerialized 0x02				
SS04	0, NotSerialized 0x03				
SS05	0, NotSerialized 0x04	Back Left Top	3	USB-A	USB 3.0
SS06	0, NotSerialized	Back Left Bottom	3	USB-A	USB 3.0

	0x05				
HS11	0x0B				
HS12	0X0C				
HS13	0X0D				
HS14	0X0E	Internal	255	Bluetooth	USB 2.0
SS07	0, NotSerialized 0x06	Back Right Top	3	USB-A	USB 3.0
SS08	0, NotSerialized 0x07	Back Right Bottom	3	USB-A	USB 3.0
SS09	0, NotSerialized 0x08				
SS10	0, NotSerialized 0x09				

Creating the USB-map.kext

- Manually building the Kext seemed too risky without understating what each line in the PLIST does. I used the method below instead.

The New Beginner's Guide to USB Port Configuration

https://www.tonymacx86.com/threads/the-new-beginners-guide-to-usb-port-configuration.286553/#post-2029768

I don't think any of the above matters. According to the OCD Guide, Skylake and newer chipsets have the ports defined in the ACPI. This makes me thing that everything would work without a USB Map, but that doesn't seem true either. All of the appropriate ports do show up without a USB Map, but they don't all work properly; especially the internal Bluetooth port. The method used in this guide is easy to follow and worked for me, resulting in a "USBPorts.kext" file that I added to my "Config.plist".

This is the end of the branch away from the OCD Guide. Click the link below to be taken back to the next logical section of the guide.
=> EXTRAS <=

Reference, Apps Installed

App	Version	Comments
1Password	7.4.3	No issues.
Day One	4.12	No issues.
Dropbox	96.4.172	No issues.
Firefox	75.0	No issues.
Geekbench	5	No issues.
Intel Power Gadget	3.7.0	No issues.
Jettison	1.7.5	Worked sometimes to resolve USB devices being improperly ejected, but overall unreliable as a permeant fix.
Logi Options	8.10.64	No issues.
Logitech Unifying	1.3.375	No issues.
Microsoft Office	16.36	Word will sometimes use more CPU than seems normal. This could possibly be fixes with power management by CPUFriend.
MonitorControl	2.0.0	No issues.
Snagit	2019.1.10	No issues.
Spotify	1.1.31	No issues.
Google Chrome	81.0.4044.138	No issues.

Reference, Audio Codec from Linux

I opened terminal and ran `cat /proc/asound/card0/codec#0`. This is what it produced:

```
Codec: Realtek ALC3234
Address: 0
```

```
AFG Function Id: 0x1 (unsol 1)
Vendor Id: 0x10ec0255
Subsystem Id: 0x1028085a
Revision Id: 0x100002
No Modem Function Group found
Default PCM:
   rates [0x560]: 44100 48000 96000 192000
   bits [0xe]: 16 20 24
   formats [0x1]: PCM
Default Amp-In caps: N/A
Default Amp-Out caps: N/A
State of AFG node 0x01:
  Power states:  D0 D1 D2 D3 D3cold CLKSTOP EPSS
  Power: setting=D0, actual=D0
GPIO: io=3, o=0, i=0, unsolicited=1, wake=0
  IO[0]: enable=0, dir=0, wake=0, sticky=0, data=0, unsol=0
  IO[1]: enable=0, dir=0, wake=0, sticky=0, data=0, unsol=0
  IO[2]: enable=0, dir=0, wake=0, sticky=0, data=0, unsol=0
Node 0x02 [Audio Output] wcaps 0x41d: Stereo Amp-Out
  Control: name="Speaker Playback Volume", index=0, device=0
    ControlAmp: chs=3, dir=Out, idx=0, ofs=0
  Amp-Out caps: ofs=0x57, nsteps=0x57, stepsize=0x02, mute=0
  Amp-Out vals:  [0x3d 0x3d]
  Converter: stream=0, channel=0
  PCM:
    rates [0x60]: 44100 48000
    bits [0xe]: 16 20 24
    formats [0x1]: PCM
  Power states:  D0 D1 D2 D3 EPSS
  Power: setting=D0, actual=D0
Node 0x03 [Audio Output] wcaps 0x41d: Stereo Amp-Out
  Control: name="Headphone+LO Playback Volume", index=0, device=0
    ControlAmp: chs=3, dir=Out, idx=0, ofs=0
  Device: name="ALC3234 Analog", type="Audio", device=0
  Amp-Out caps: ofs=0x57, nsteps=0x57, stepsize=0x02, mute=0
  Amp-Out vals:  [0x00 0x00]
  Converter: stream=0, channel=0
```

PCM: rates [0x60]: 44100 48000 bits [0xe]: 16 20 24 formats [0x1]: PCM Power states: D0 D1 D2 D3 EPSS

Reference, System Power States

S0	Working
S0 low-power idle	Sleep known as Modern Standby. Systems that support Modern Standby do not use S1-S3.
S1, S2, and S3	Sleep
S4	Hibernation
S5	Soft Off. Trickle to system lights and such.
G3	No power at all.

Reference, Full Dell BIOS for OptiPlex 7060

https://www.dell.com/support/manuals/au/en/aubsd1/optiplex-7060-desktop/opti_7060_mff_setup_specs_manual/system-setup-options?guid=guid-c0b6998f-efac-4d0b-93c6-ec55a64e0f81&lang=en-us

General		
	System Information	- Nothing to change
	Boot Sequence	- Checked primary drive - Checked UEFI
	Advanced Boot Options	- Uncheck Enable Legacy Option ROMs - No need for legacy boot options.
	UEFI Boot Path Security	- Selected Never - Not using Admin password
	Date/Time	- Set to current date/time.

		- For some reason, my BIOS date and time are off every time I enter the BIOS. It jumps ahead 5 hours.
System Configuration		
	Integrated NIC	- Unchecked Enable UEFI Network stack. - Not needed if not PXE booting. - Selected Enabled
	SATA Operation	- Selected AHCI - Needed for SATA drives and I am not using RAID.
	Drives	- Checked all drives/SATA ports
	SMART Reporting	- Checked Enable
	USB Configuration	- Checked Enable USB Boot Support - Checked Enable Front and Rear
	Front USB Configuration	- Checked al
	Read USB Configuration	- Checked all
	USB PowerShare	- Unchecked - I will try this later and see if it prevents Sleep from working.
	Audio	- Checked all
	Dust Filter Maintenance	- Selected Disabled
Video	Primary Display	- Selected Intel HD Graphics
Security	Admin Password	- Not set

	System Password	- Not set
	Strong Password	- Unchecked
	Password Configuration	- Default
	Password Bypass	- Selected Disabled
	Password Change	- Checked but not using
	UEFI Capsule Firmware Updates	- Checked
	TPM 2.0 Security	- Unchecked
	Absolute	- Permanently disabled
	Chassis Intrusion	- Disabled
	OROM Keyboard Access	- Enabled
	Admin Setup Lockout	- Unchecked
	Master Password Lockout	- Unchecked
	SMM Security Mitigation	- Unchecked
Secure Boot	Secure Boot Enable	- Unchecked
	Secure Boot Mode	- Deployed Mode
	Expert Key Management	- Unchecked
Intel Software Guard Extensions	Intel SGX Enable	- Disabled
	Enclave Memory Size	- Dithered
Performance	Multi Core Support	- All
	Intel SpeedStep	- Checked

	C-States Control	- Checked
	Intel TurboBoost	- Checked
Power Management	AC Recovery	- Last Power State
	Enable Intel Speed Shift Technology	- Checked
	Auto On Time	- Disabled
	Deep Sleep Control	- Disabled - Enabling this prevents Sleep from working.
	USB Wake Support	- Checked
	Wake on LAN/WLAN	- LAN of WLAN
	Block Sleep	- Unchecked
POST Behavior	Adapter Warnings	- Checked
	Numlock LED	- Checked
	Keyboard Errors	- Checked
	Fastboot	- Thorough. - OpenCore Guide recommends that Fastboot to be set to off.
	Extend BIOS POST Time	- 0 Seconds
	Full Screen Logo	- Unchecked. - I would like to know if I can use my own logo.
	Warning and Errors	- Prompt on Warnings and Errors
Manageability	Intel AMT Capability	- Disabled

	USB Provision	- Dithered
	MEBx Hotkey	- Dithered
Virtualization Support	Virtualization	- Checked
	VT for Direct I/O	- Unchecked
	Trusted Execution	- Dithered
Wireless	Wireless Device Enable	- Checked all
Maintenance	Service Tag	- Informational
	Asset Tag	- Informational
	SERR Messages	- Checked
	BIOS Downgrade	- Checked
	Data Wipe	- Unchecked
	BIOS Recovery	- Checked first option, but not auto-recovery
System Logs	BIOS Events	- Informational
Advanced configurations	ASPM	- Disabled