



# OpenCore AIO Folder Install Guide

Current version: 0.6.4

# CONTENT

- |       |                                |
|-------|--------------------------------|
| one   | what is OpenCore?              |
|       | OpenCore folder                |
|       | AIO OpenCore folder            |
| two   | acpi                           |
| three | kernel extension (kext)        |
| four  | building a complete AIO folder |
| five  | contact                        |



# **OPENCORE, OPENCORE FOLDER AND AIO FOLDER**

# OPEN-CORE.

## WHAT IS OPENCORE?

OpenCore is an open-source unconventional first-in-class piece of software designed to intercept kernel loading to insert a highly advanced rootkit, designed to be an alternative to Clover. OpenCore aims to resolve the constraints and issues imposed by Clover by providing a more versatile and modular system. While OpenCore is primarily designed for Hackintosh systems, it can be used in any scenario where an emulated EFI is needed with many using it on KVMs and real Macs

So, means OpenCore is a “bootloader” – it is a complex piece of software that use to prepare the systems for macOS – specifically by injecting new data for macOS such as SMBIOS, ACPI tables and kexts. How this tool differs from others like Clover is that it has been designed with security and quality in mind, allowing to use many security features found on real Macs, such as SIP and FileVault.



# OpenCore

# OC-FOLDER

## WHAT ARE INSIDE OPENCORE FOLDER?

### 1. BOOT

Duet bootstrap loader, which initialises UEFI environment on legacy BIOS firmware and loads OpenCore.efi similarly to other bootstrap loaders. Modern Duet bootstrap loader will default to OpenCore.efi on the same partition when present.

### 2. ACPI

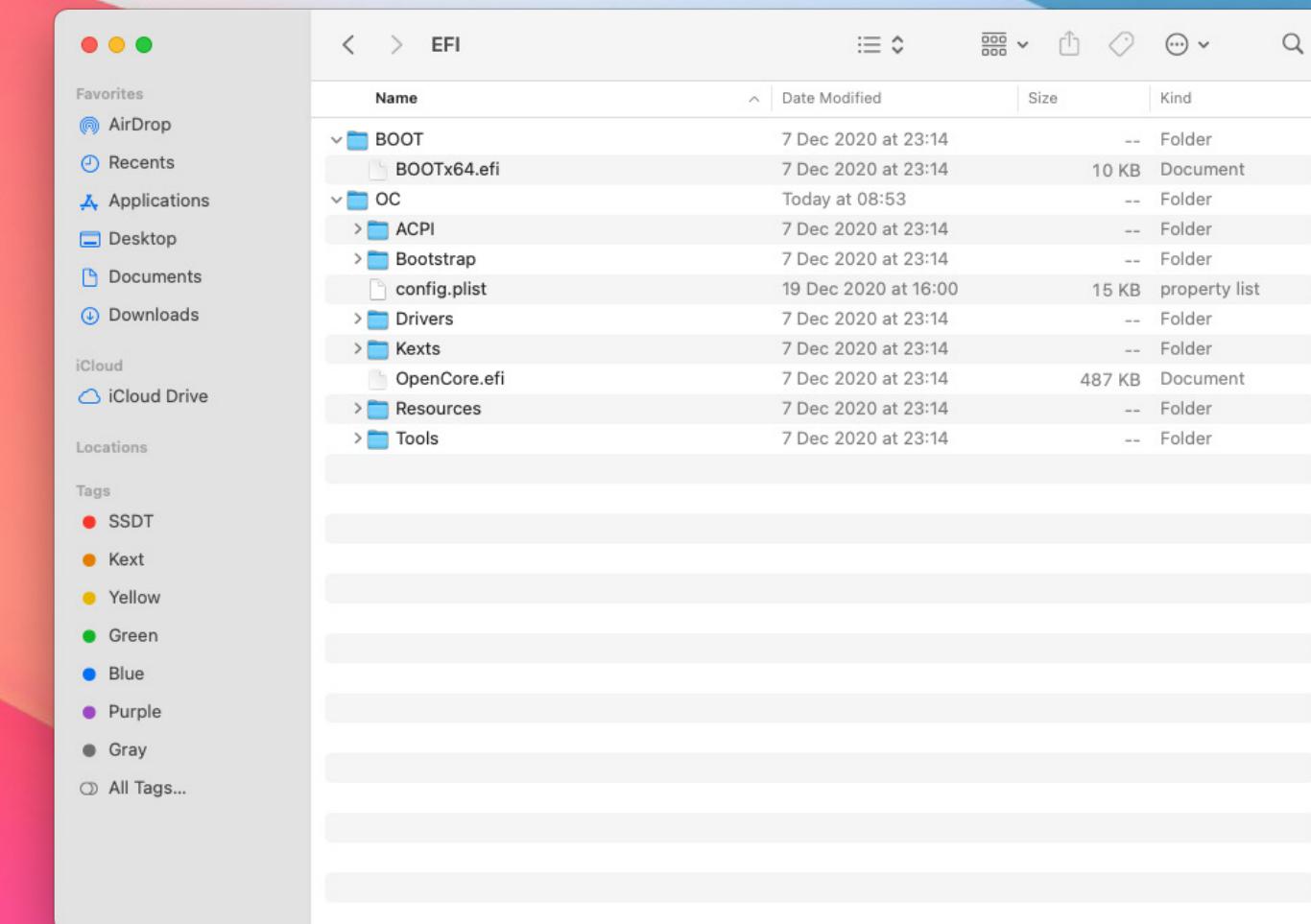
Directory used for storing supplemental ACPI information for ACPI section.

### 3. Drivers

Directory used for storing supplemental UEFI drivers for UEFI section.

### 4. Kexts

Directory used for storing supplemental kernel information for Kernel section.



# OC-FOLDER

## WHAT ARE INSIDE OPENCORE FOLDER?

### 5. Resources

Directory used for storing media resources, such as audio files for screen reader support. This directory also contains image files for graphical user interface.

### 6. Tools

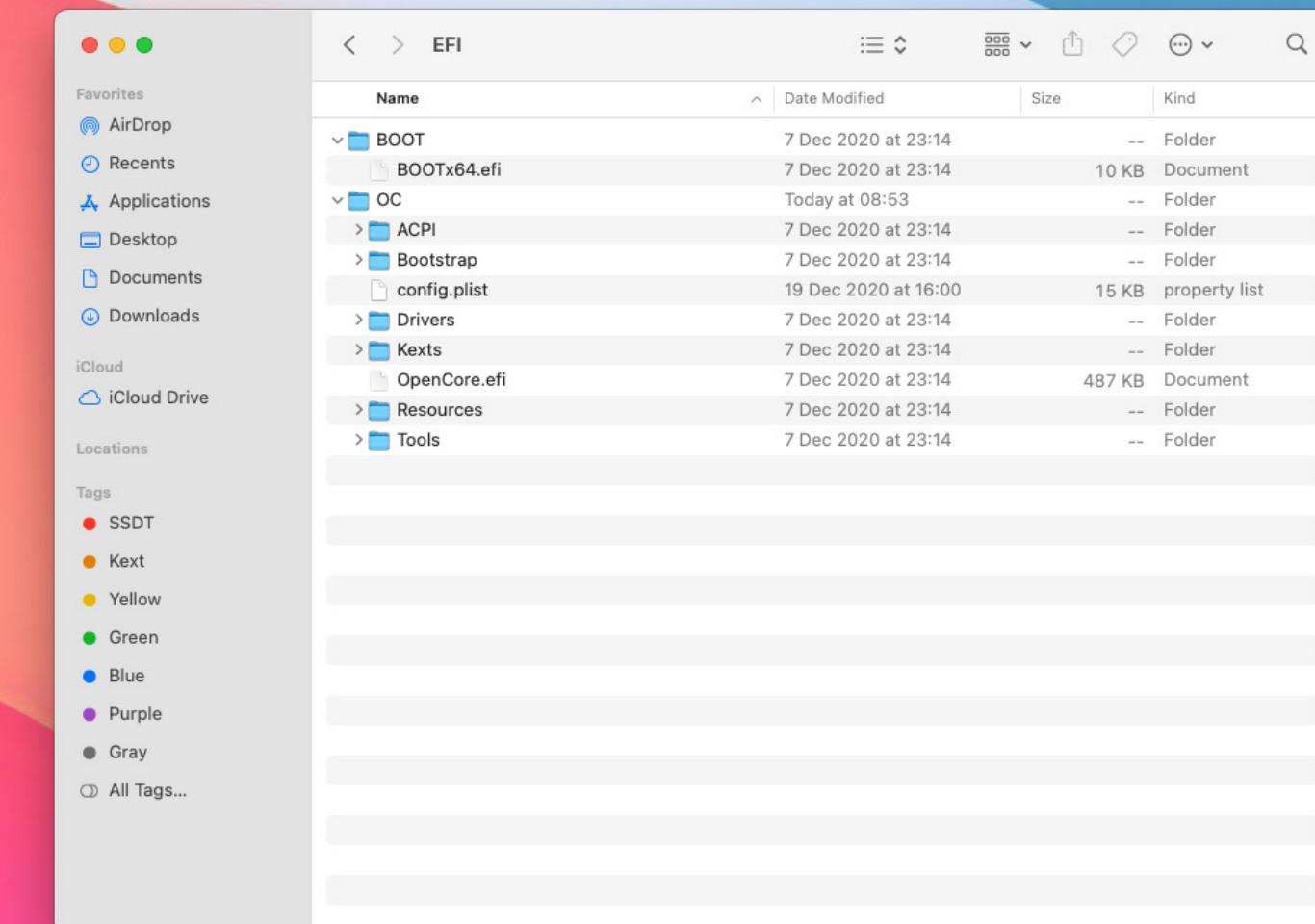
Directory used for storing supplemental tools.

### 7. OpenCore.efi

Main booter driver responsible for operating system loading. The directory OpenCore.efi resides is called the root directory. By default root directory is set to EFI\OC, however, when launching OpenCore.efi directly or through Bootstrap.efi, other directories containing OpenCore.efi can also be supported.

### 8. config.plist

OC Config

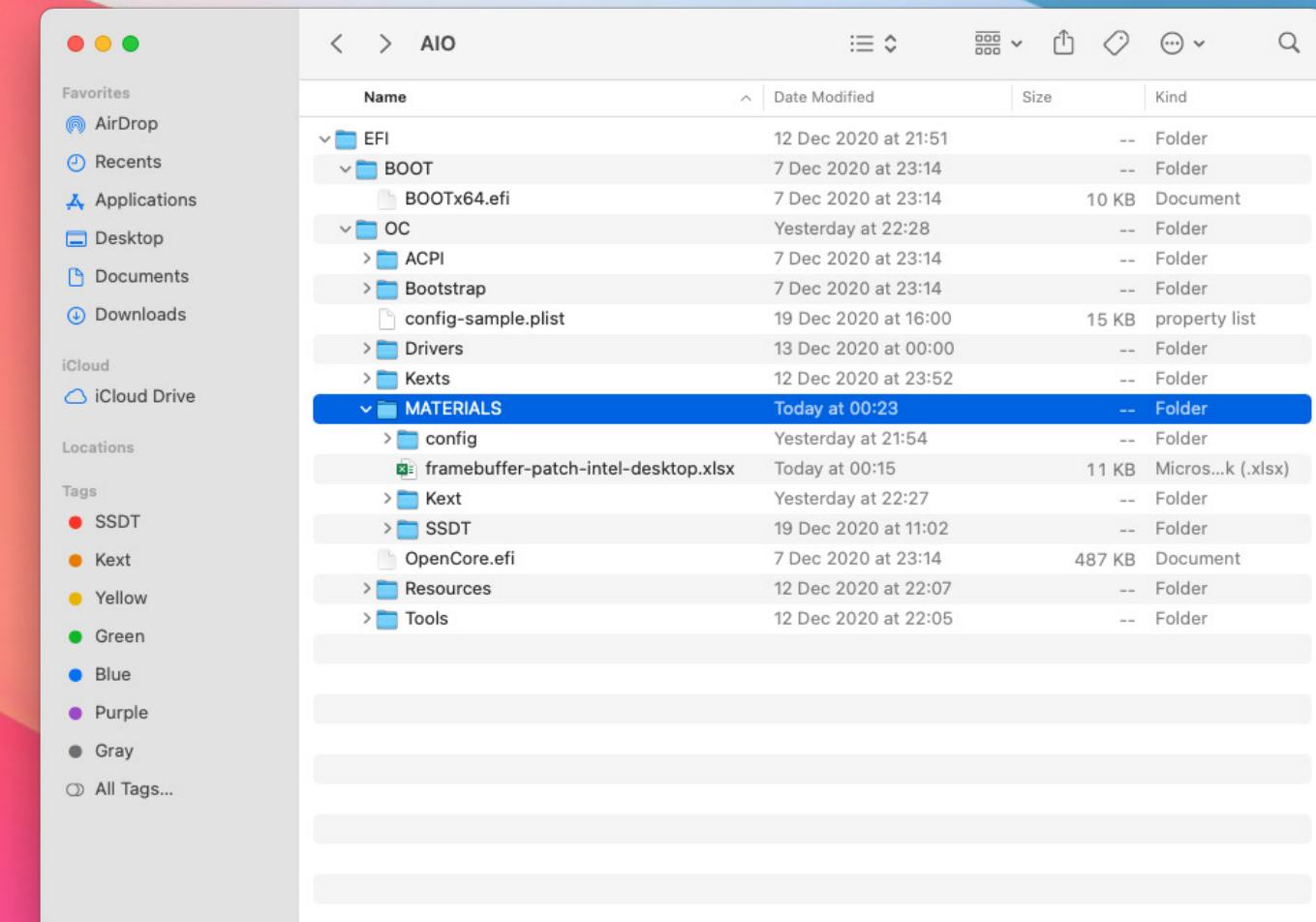


# AIO OC-FOLDER

## WHAT IS AIO OPENCORE FOLDER?

It's OpenCore folder I made is already clean.  
There is MATERIALS folder which has:

1. AIO SSDTs and config.plist files according Intel CPU generation (from Haswell to Comet Lake) and AMD CPU.
2. Necessary kexts for Hackintosh (will update more in the future).
3. iGPU framebuffer patching file for desktop with iGPU.



# **ACPI, DSDT AND SSDT**

# DSDT- SSDT

## WHAT IS ACPI, DSDT, SSDT?

ACPI (Advanced Configuration and Power Interface) is an open industry specification co-developed by Hewlett-Packard, Intel, Microsoft, Phoenix, and Toshiba that defines hardware and software interfaces that enable OS-directed configuration and Power Management (OSPM) to enumerate and configure motherboard devices, and manage their power. The tables allow description of system hardware in a platform-independent manner, and are presented as either fixed-formatted data structures or in AML. The main AML table is the DSDT.

So what are DSDTs and SSDTs? Well, these are tables present in your firmware that outline hardware devices like USB controllers, CPU threads, embedded controllers, system clocks and such. A DSDT(Differentiated System Description Table) can be seen as the body holding most of the info with smaller bits of info being passed by the SSDT(Secondary System Description Table). You can think of the DSDT as the building blueprints with SSDTs being sticky notes outlining extra details to the project.



# DSDT- SSDT

## ACPI LIBRARY

macOS can be very picky about the devices present in the DSDT and so our job is to correct it. The main devices that need to be corrected for macOS to work properly:

### 1. Embedded controllers (EC)

All semi-modern intel machines have an EC (usually called H\_EC, ECDV, ECO, etc...) exposed in their DSDT, with many AMD systems also having it exposed. These controllers are generally not compatible with macOS and can cause panics, so then need to be hidden from macOS. macOS Catalina requires a device named EC to be present though, so a dummy EC is created.  
With laptops, the actual embedded controller still needs to be enabled for battery and hotkeys to work, and renaming the EC can additionally cause issues with windows, so creating a fake EC without disabling the real embedded controller is preferable.

```

1  /*
2  * Intel ACPI Component Architecture
3  * AML/ASL+ Disassembler version 20200925 (64-bit version)
4  * Copyright (c) 2000 - 2020 Intel Corporation
5  *
6  * Disassembling to symbolic ASL+ operators
7  *
8  * Disassembly of iASLOHbj04.aml, Tue Dec 15 16:08:16 2020
9  */
10 /* Original Table Header:
11  *   Signature      "SSDT"
12  *   Length        0x000000BE (190)
13  *   Revision       0x02
14  *   Checksum       0x30
15  *   OEM ID         "DRTNIA"
16  *   OEM Table ID   "SsdtEC"
17  *   OEM Revision    0x00001000 (4096)
18  *   Compiler ID     "INTL"
19  *   Compiler Version 0x20190509 (538510601)
20 */
21 DefinitionBlock ("", "SSDT", 2, "DRTNIA", "SsdtEC", 0x00001000)
22 {
23     External (_SB_.PCI0.LPC_.EC_, DeviceObj)
24     External (_SB_.PCI0.LPCB.EC_, DeviceObj)
25
26     If (!CondRefOf (\_SB.PCI0.LPCB.EC) && !CondRefOf (\_SB.PCI0.LPC.EC))
27     {
28         Scope (\_SB)
29         {
30             Device (EC)
31             {
32                 Name (_HID, "ACID0001") // _HID: Hardware ID
33                 Method (_STA, 0, NotSerialized) // _STA: Status
34                 {
35                     If (_OSI ("Darwin"))
36                     {
37                         Return (0xF)
38                     }
39                     Else
40                     {
41                         Return (Zero)
42                     }
43                 }
44             }
45         }
46     }
47 }
48

```

credit: Dortania's EC Laptop Prebuilt

# DSDT- SSDT

## 2. Plugin Type

This allows the use of XCPM providing native CPU power management on Intel Haswell and newer CPUs, the SSDT will connect to the first thread of the CPU. Not meant for AMD

```

1  * Intel ACPI Component Architecture
2  * AML/ASL+ Disassembler version 20200925 (64-bit version)
3  * Copyright (c) 2000 - 2020 Intel Corporation
4  *
5  * Disassembling to symbolic ASL+ operators
6  *
7  * Disassembly of iASLNaJvDd.aml, Tue Dec 15 16:12:44 2020
8  *
9  * Original Table Header:
10 *   Signature      "SSDT"
11 *   Length        0x000002B5 (693)
12 *   Revision       0x02
13 *   Checksum       0xDD
14 *   OEM ID         "DRTNIA"
15 *   OEM Table ID  "CpuPlug"
16 *   OEM Revision   0x00003000 (12288)
17 *   Compiler ID    "INTEL"
18 *   Compiler Version 0x20190509 (538510601)
19 */
20 /*
21 DefinitionBlock ("", "SSDT", 2, "DRTNIA", "CpuPlug", 0x00003000)
22 {
23     External (_PR_.C000, ProcessorObj)
24     External (_PR_.CPU0, ProcessorObj)
25     External (_PR_.P000, ProcessorObj)
26     External (_PR_.PR00, ProcessorObj)
27     External (_SB_.CPU0, ProcessorObj)
28     External (_SB_.PR00, ProcessorObj)
29     External (_SB_.SCK0.CP00, ProcessorObj)
30     External (_SB_.SCK0.PR00, ProcessorObj)
31
32     Method (PMPM, 4, NotSerialized)
33     {
34         If ((Arg2 == Zero))
35         {
36             Return (Buffer (One)
37             {
38                 0x03
39             })
40         }
41
42         Return (Package (0x02)
43         {
44             "plugin-type",
45             One
46         })
47     }
48
49     If (CondRefOf (_SB.CPU0))
50     {
51         If ((ObjectType (_SB.CPU0) == 0x0C))
52         {
53             Scope (_SB.CPU0)
54             {
55                 Method (_DSM, 4, NotSerialized) // _DSM: Device-Specific Method
56                 {
57                     Return (PMPM (Arg0, Arg1, Arg2, Arg3))
58                 }
59             }
60         }
61     }
62 }

```

credit: Dortania's PLUG Prebuilt

## 3. AWAC system clock

This applies to all 300 series motherboards including many Z370 boards, the specific issue is that newer boards ship with AWAC clock enabled. This is a problem because macOS cannot communicate with AWAC clocks, so this requires us to either force on the legacy RTC clock or if unavailable create a fake one for macOS to play with

```

1  * Intel ACPI Component Architecture
2  * AML/ASL+ Disassembler version 20200925 (64-bit version)
3  * Copyright (c) 2000 - 2020 Intel Corporation
4  *
5  * Disassembling to symbolic ASL+ operators
6  *
7  * Disassembly of iASLh0l48B.aml, Tue Dec 15 16:22:38 2020
8  *
9  * Original Table Header:
10 *   Signature      "SSDT"
11 *   Length        0x00000049 (73)
12 *   Revision       0x02
13 *   Checksum       0x2D
14 *   OEM ID         "DRTNIA"
15 *   OEM Table ID  "AWAC"
16 *   OEM Revision   0x00000000 (0)
17 *   Compiler ID    "INTEL"
18 *   Compiler Version 0x20190509 (538510601)
19 */
20 /*
21 DefinitionBlock ("", "SSDT", 2, "DRTNIA", "AWAC", 0x00000000)
22 {
23     External (STAS, IntObj)
24
25     Scope (_SB)
26     {
27         If (_OSI ("Darwin"))
28         {
29             STAS = One
30         }
31     }
32 }
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62

```

credit: Dortania's AWAC Prebuilt

# DSDT- SSDT

## 4. NVRAM

True 300 series motherboards(non-Z370) don't declare the FW chip as MMIO in ACPI and so the kernel ignores the MMIO region declared by the UEFI memory map. This SSDT brings back NVRAM support

## 4. Backlight

Used for fixing backlight control support on laptops

SSDT-PMC.aml

Aa  
Fonts  
Compile  
Patch  
Summary  
Log

```
2 * Intel ACPI Component Architecture
3 * AML/ASL+ Disassembler version 20200925 (64-bit version)
4 * Copyright (c) 2000 - 2020 Intel Corporation
5 *
6 * Disassembling to symbolic ASL+ operators
7 *
8 * Disassembly of iASL0oaVw6.aml, Tue Dec 15 16:28:46 2020
9 *
10 * Original Table Header:
11 *   Signature      "SSDT"
12 *   Length         0x0000019E (414)
13 *   Revision       0x02
14 *   Checksum       0x24
15 *   OEM ID         "DRTNIA"
16 *   OEM Table ID  "PMCR"
17 *   OEM Revision   0x00001000 (4096)
18 *   Compiler ID    "INTL"
19 *   Compiler Version 0x20190509 (538510601)
20 */
21 DefinitionBlock ("", "SSDT", 2, "DRTNIA", "PMCR", 0x00001000)
22 {
23     External (_SB_.PCI0.LPC0, DeviceObj)
24     External (_SB_.PCI0.LPC, DeviceObj)
25     External (_SB_.PCI0.LPCB, DeviceObj)
26
27     If (CondRefOf (\_SB.PCI0.LPCB))
28     {
29         Scope (_SB.PCI0.LPCB)
30         {
31             Device (PMCR)
32             {
33                 Name (_HID, EisaId ("APP9876")) // _HID: Hardware ID
34                 Method (_STA, 0, NotSerialized) // _STA: Status
35                 {
36                     If (_OSI ("Darwin"))
37                     {
38                         Return (0x0B)
39                     }
40                     Else
41                     {
42                         Return (Zero)
43                     }
44                 }
45
46                 Name (_CRS, ResourceTemplate ()) // _CRS: Current Resource Settings
47                 {
48                     Memory32Fixed (ReadWrite,
49                         0xFE000000, // Address Base
50                         0x00010000, // Address Length
51                     )
52                 }
53             }
54         }
55
56         If (CondRefOf (\_SB.PCI0.LPC0))
57         {
58             Scope (_SB.PCI0.LPC0)
59             {
60                 Device (PMCR)
61             }
62         }
63     }
64 }
```

The screenshot shows the Intel ASL+ Disassembler application window. The title bar reads "SSDT-PNLF-CFL.aml". The left sidebar shows a tree view with a selected node: "SSDT" > "\_SB.PCI0.GFX0.PNLF". The main pane displays the assembly code for the SSDT table. The code includes comments about the Intel ACPI Component Architecture, the disassembly version, copyright information, and the date of compilation. It defines the SSDT table header with fields like Signature, Length, Revision, Checksum, OEM ID, OEM Table ID, OEM Revision, Compiler ID, and Compiler Version. The code then defines a DefinitionBlock with a name of "", a type of "SSDT", a class of 2, a class identifier of "ACDT", and a GUID of "PNLFCFL". This block contains an External definition for the device object and a Device definition for the "\_SB.PCI0.GFX0.PNLF" device. The Device definition includes names for Address, Hardware ID, Compatible ID, Unique ID, and Status.

```
/*
 * Intel ACPI Component Architecture
 * AML/ASL+ Disassembler version 20200925 (64-bit version)
 * Copyright (c) 2000 - 2020 Intel Corporation
 *
 * Disassembling to symbolic ASL+ operators
 *
 * Disassembly of iASLRo2GgB.aml, Tue Dec 15 16:31:28 2020
 *
 * Original Table Header:
 */
Signature      "SSDT"
Length         0x0000007D (125)
Revision       0x02
Checksum       0x84
OEM ID         "ACDT"
OEM Table ID   "PNLFCFL"
OEM Revision   0x00000000 (0)
Compiler ID    "INTEL"
Compiler Version 0x20190509 (538510601)
*/
DefinitionBlock ("", "SSDT", 2, "ACDT", "PNLFCFL", 0x00000000)
{
    External (_SB_.PCI0.GFX0, DeviceObj)
    Device (_SB_.PCI0.GFX0.PNLF)
    {
        Name (_ADR, Zero) // _ADR: Address
        Name (_HID, EisId ("APP0002")) // _HID: Hardware ID
        Name (_CID, "backlight") // _CID: Compatible ID
        Name (_UID, 0x13) // _UID: Unique ID
        Name (_STA, 0x0B) // _STA: Status
    }
}
```

*credit: Dortania's PMC Prebuilt*

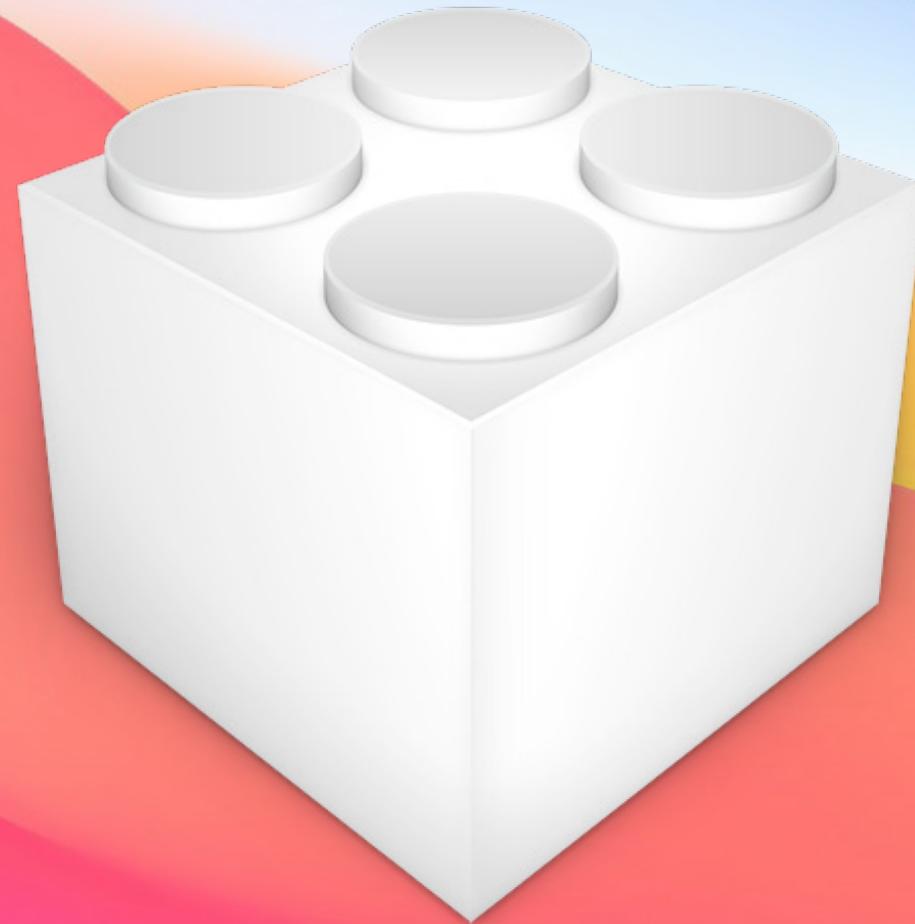
*credit: Dortania's PNFL Prebuilt*

# **KERNEL EXTENSION (KEXT)**

# KEXT

## WHAT IS KEXT?

Kext files are essentially drivers for Mac OS. "Kext" stands for Kernel Extension; kext files "extend" Mac OS X's kernel, the core part of the operating system, by providing additional code to be loaded when your computer boots. "Hackintoshes" often require special kexts to enable sound, ethernet, and more. Some Hackintosh-specific kexts are modified versions of existing Mac OS X kexts. Other kexts are extra additions to the normal list of kexts that OS X runs on startup. In the end, all of these Hackintosh kexts serve the same purpose: to add support to hardware that isn't officially supported by Apple.



# KEXT

## KEXT LIBRARY

There are some common kexts, and should need for a Hackintosh machine.

### 1. Lilu (Essential kext)

A kext to patch many processes, required for AppleALC, WhateverGreen, VirtualSMC and many other kexts. Without Lilu, they will not work.

### 2. VirtualSMC (Essential kext)

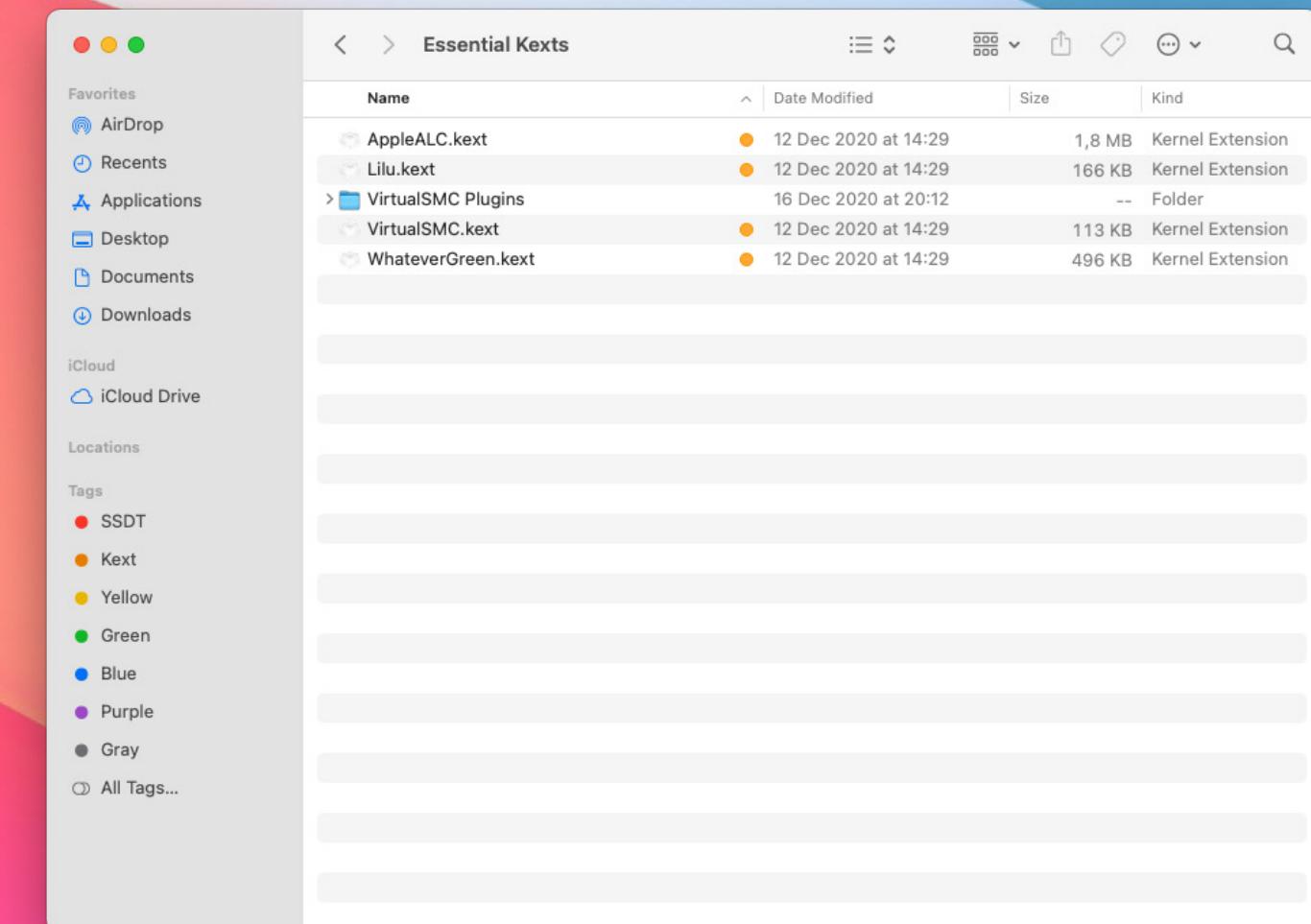
Emulates the SMC chip found on real macs, without this macOS will not boot.  
Alternative is FakeSMC which can have better or worse support, most commonly used on legacy hardware.

### 3. WhateverGreen (Graphics)

Used for graphics patching DRM, boardID, framebuffer fixes, etc, all GPUs benefit from this kext.

### 4. AppleALC (Audio)

Used for AppleHDA patching, allowing support for the majority of on-board sound controllers



# KEXT

## KEXT LIBRARY

### Ethernet:

#### 1. IntelMausi

- Required for the majority of Intel NICs, chipsets that are based off of I211 will need the SmallTreeIntel82576 kext
- Intel's 82578, 82579, i217, i218 and i219 NICs are officially supported

#### 2. SmallTreeIntel82576

- Required for I211 NICs, based off of the SmallTree kext but patched to support I211
- Required for most AMD boards running Intel NICs

#### 3. AtherosE2200Ethernet

- Required for Atheros and Killer NICs

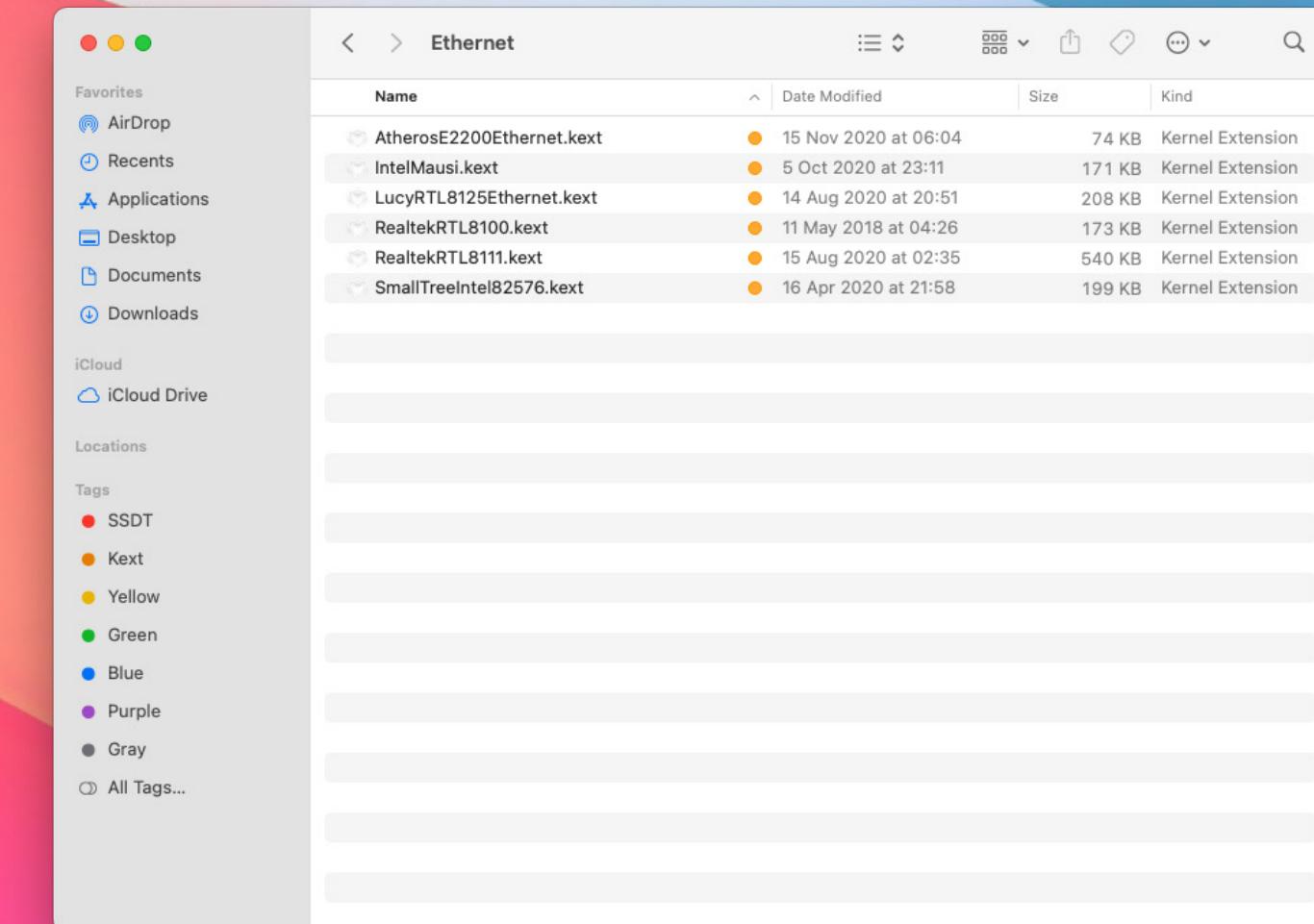
Note: Atheros Killer E2500 models are actually Realtek based, for these systems please use RealtekRTL8111 instead

#### 4. RealtekRTL8111

- For Realtek's Gigabit Ethernet

#### 5. LucyRTL8125Ethernet

- For Realtek's 2.5Gb Ethernet



# KEXT

## KEXT LIBRARY

### USB:

#### 1. USBInjectAll

- Used for injecting Intel USB controllers on systems without defined USB ports in ACPI
- AsRock motherboard should need this
- Coffee Lake and older laptops are however recommended to use this kext
- Does not work on AMD CPUs at all

#### 2. XHCI-unsupported

- Needed for non-native USB controllers
- AMD CPU based systems don't need this
- Common chipsets needing this:

H370

B360

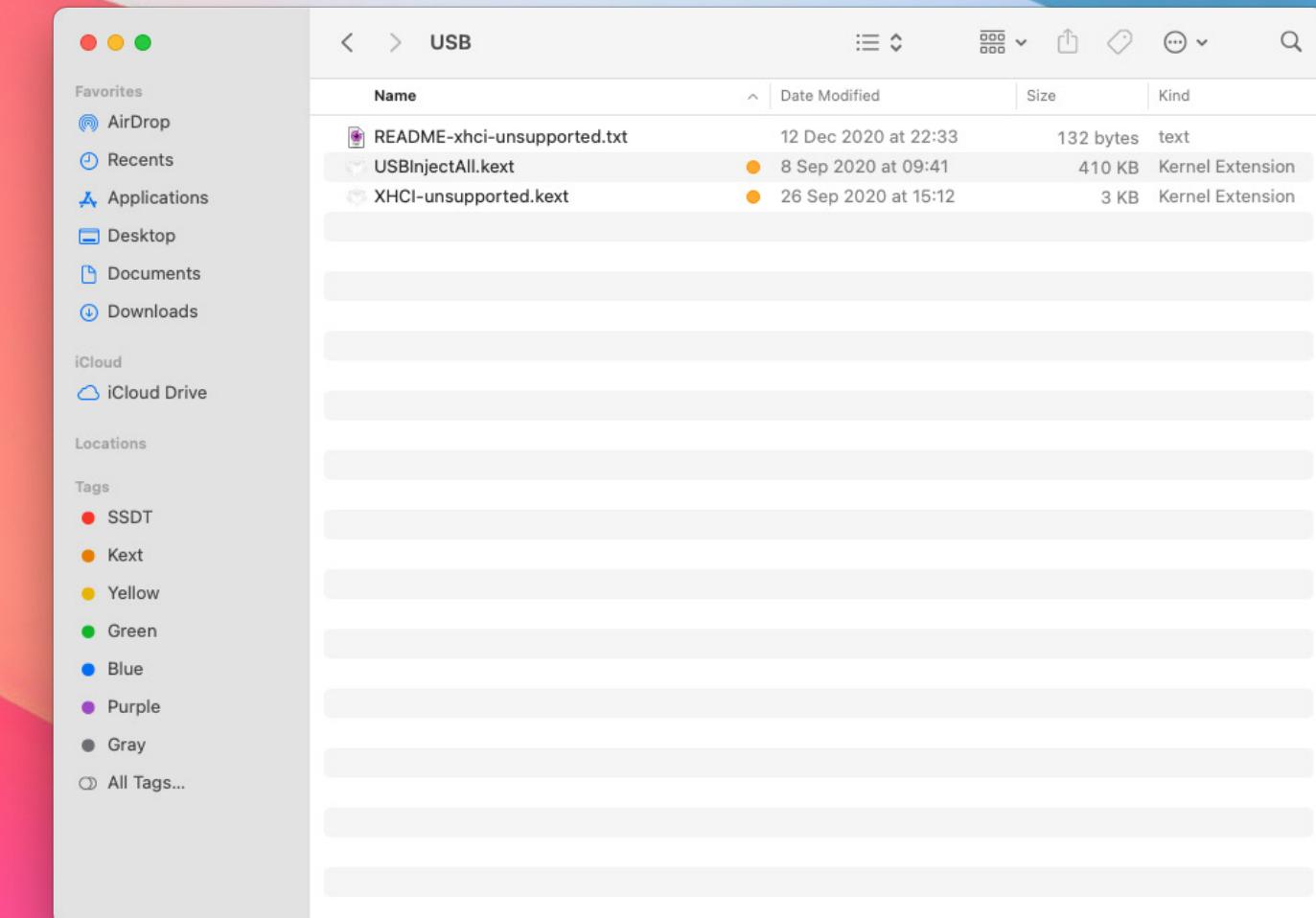
H310

Z390(Not needed on Mojave and newer)

X79

X99

AsRock boards(On Intel motherboards specifically, B460/Z490+ boards do not need it however)



## KEXT LIBRARY

### Wifi/Bluetooth

#### 1. Intel

##### Wifi: AirportItlwm

- Adds support for a large variety of Intel wireless cards and works natively in recovery thanks to IO80211Family integration.
- Requires Apple's Secure Boot to function correctly.

##### Bluetooth: IntelBluetoothFirmware

- Adds Bluetooth support to macOS when paired with an Intel wireless card.

#### 2. Broadcom

##### Wifi: AirportBrcmFixup

- Used for patching non-Apple/non-Fenvi Broadcom cards, will not work on Intel, Killer, Realtek, etc...

##### Bluetooth: BrcomPatchRAM

- Used for uploading firmware on Broadcom Bluetooth chipset, required for all non-Apple/non-Fenvi Airport cards.
- To be paired with BrcmFirmwareData.kext:
  - a. BrcomPatchRAM3 for 10.15+ (must be paired with BrcomBluetoothInjector)
  - b. BrcomPatchRAM2 for 10.11-10.14
  - c. BrcomPatchRAM for 10.8-10.10

Name	Date Modified	Size	Kind
↳ Broadcom (DW1560/1820a)	Today at 09:15	--	Folder
↳ AirportBrcmFixup.kext	7 Dec 2020 at 22:49	62 KB	Kernel Extension
↳ BrcmBluetoothInjector.kext	5 Oct 2020 at 23:12	75 KB	Kernel Extension
↳ BrcmFirmwareData.kext	5 Oct 2020 at 23:12	2,8 MB	Kernel Extension
↳ BrcmFirmwareRepo.kext	5 Oct 2020 at 23:12	2,8 MB	Kernel Extension
↳ BrcmNonPatchRAM.kext	5 Oct 2020 at 23:12	35 KB	Kernel Extension
↳ BrcmNonPatchRAM2.kext	5 Oct 2020 at 23:12	38 KB	Kernel Extension
↳ BrcmPatchRAM.kext	5 Oct 2020 at 23:12	144 KB	Kernel Extension
↳ BrcmPatchRAM2.kext	5 Oct 2020 at 23:12	153 KB	Kernel Extension
↳ BrcmPatchRAM3.kext	5 Oct 2020 at 23:12	138 KB	Kernel Extension
↳ Intel	12 Dec 2020 at 22:48	--	Folder
↳ AirportItlwm_BigSur.kext	3 Dec 2020 at 21:59	13,9 MB	Kernel Extension
↳ AirportItlwm_Catalina.kext	3 Dec 2020 at 21:59	13,9 MB	Kernel Extension
↳ IntelBluetoothFirmware.kext	3 Aug 2020 at 22:27	15,5 MB	Kernel Extension
↳ IntelBluetoothInjector.kext	3 Aug 2020 at 22:27	15 KB	Kernel Extension

# KEXT

## KEXT LIBRARY

### Laptop Specifics

#### 1. VoodooPS2

- For systems with PS2 keyboards, mice, and trackpads

#### 2. RehabMan's VoodooPS2

- For older systems with PS2 keyboards, mice, and trackpads, or when you don't want to use VoodooInput.

#### 3. VoodooRMI

- For systems with Synaptics SMBus-based devices, mainly for trackpads and trackpoints.

#### 4. VoodooSMBus

- For systems with ELAN SMBus-based devices, mainly for trackpads and trackpoints.

#### 5. Voodool2C

- Used for fixing I2C devices, found with some fancier touchpads and touchscreen machines.

#### 6. BrightnessKeys

- Mapping Fn brightness keys

Laptop Specifics

Name	Date Modified	Size	Kind
BrightnessKeys.kext	3 Nov 2020 at 02:18	39 KB	Kernel Extension
RehabMansVoodooPS2Controller.kext	8 Oct 2018 at 21:45	298 KB	Kernel Extension
Voodool2C.kext	9 Oct 2020 at 06:08	551 KB	Kernel Extension
Voodool2CAtmelMXT.kext	9 Oct 2020 at 06:08	50 KB	Kernel Extension
Voodool2CELAN.kext	9 Oct 2020 at 06:08	49 KB	Kernel Extension
Voodool2CFTE.kext	9 Oct 2020 at 06:08	49 KB	Kernel Extension
Voodool2CHID.kext	9 Oct 2020 at 06:08	222 KB	Kernel Extension
Voodool2CSynaptics.kext	9 Oct 2020 at 06:08	54 KB	Kernel Extension
VoodooPS2Controller.kext	7 Dec 2020 at 22:59	343 KB	Kernel Extension
VoodooRMI.kext	6 Oct 2020 at 02:14	723 KB	Kernel Extension
VoodooSMBus.kext	6 Oct 2020 at 02:05	104 KB	Kernel Extension
README.txt	12 Dec 2020 at 23:12	114 bytes	text



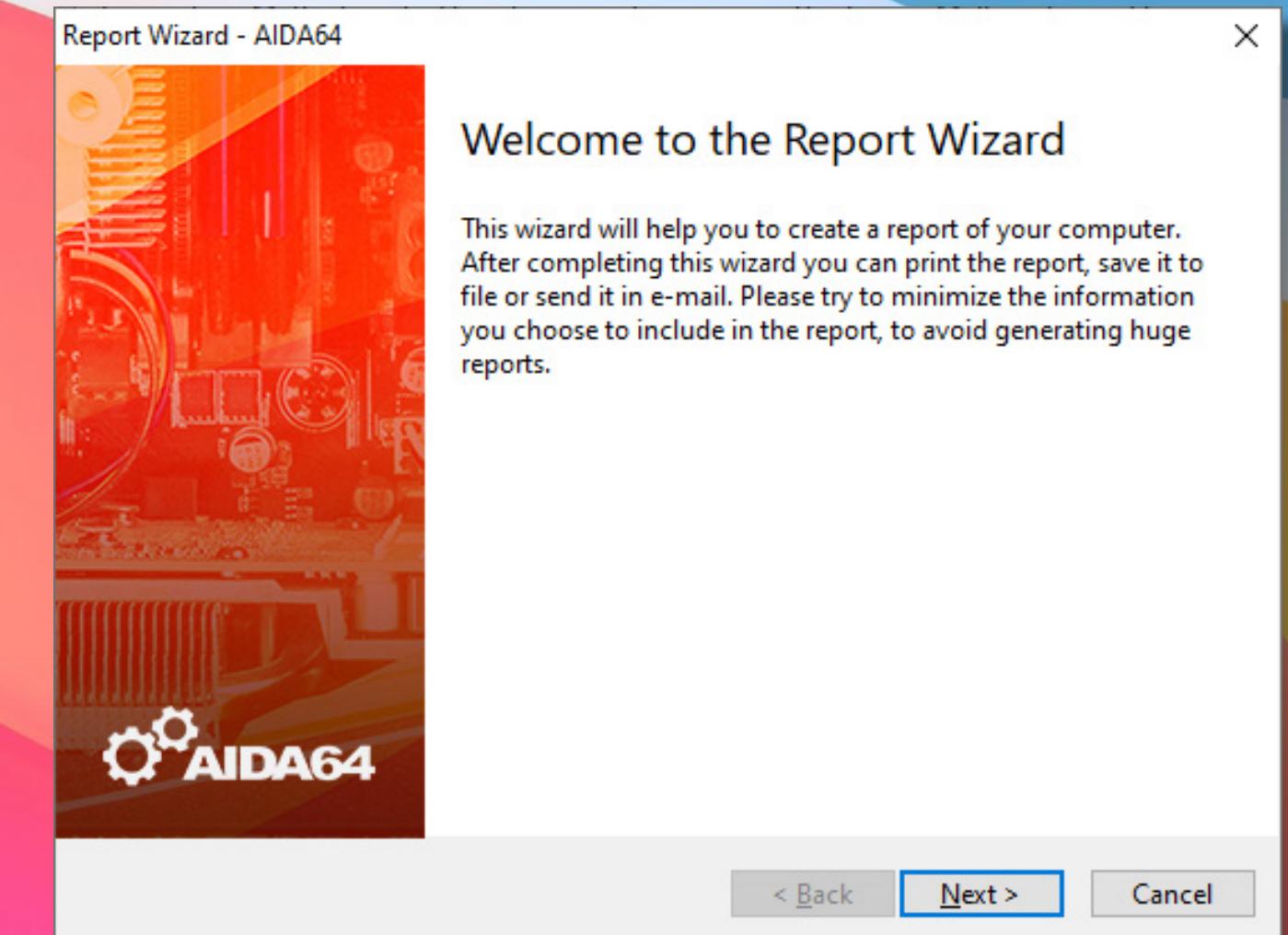
# **BUILDING A COMPLETE AIO FOLDER**

# HARDWARE-CHECKING

## PREPARATION

Use the software like AIDA64 for checking your hardware.

- Choose **Report**  to open Report Wizard.
- Click Next > Hardware-related pages > HTML
- Then check the following things:
  - 1. Audio**
  - 2. Ethernet**
  - 3. Wireless** (on some desktop machine if you have Wi-Fi card, you can check it)
  - 4. Keyboard/Mouse** (Only laptop need this, desktop can skip)



# HARDWARE-CHECKING

## PREPARATION

- For example, on my Dell G7 7888 (Laptop model) I have:

1. **Audio** Realtek ALC256
2. **Ethernet** Killer E2400
3. **Wireless** BCM4352 (this wifi card is DW1560, replaced the original is Intel AC 9560)
4. **Keyboard** HID, Standard PS/2; **Mouse** HID

Realtek ALC256 @ Intel Cannon Point PCH - cAVS (Audio, Voice, Speech) [B0]

 Broadcom BCM4352 802.11ac Wireless Network Adapter  
 Killer e2400 PCI-E Gigabit Ethernet Controller

### Input:

Keyboard	HID Keyboard Device
Keyboard	HID Keyboard Device
Keyboard	Standard PS/2 Keyboard
Mouse	HID-compliant mouse
Mouse	HID-compliant mouse

# KEXT-GATHERING

➡ About Audio, let check the Audio codec from Acidanthera on this wiki website:  
<https://github.com/acidanthera/AppleALC/wiki/Supported-codecs>

For example, the ALC on my Dell is 256, there are a lot of layout id in the picture. But for me, currently I use id 21.

Trick: You can check the closest ID with your machine by using this link:

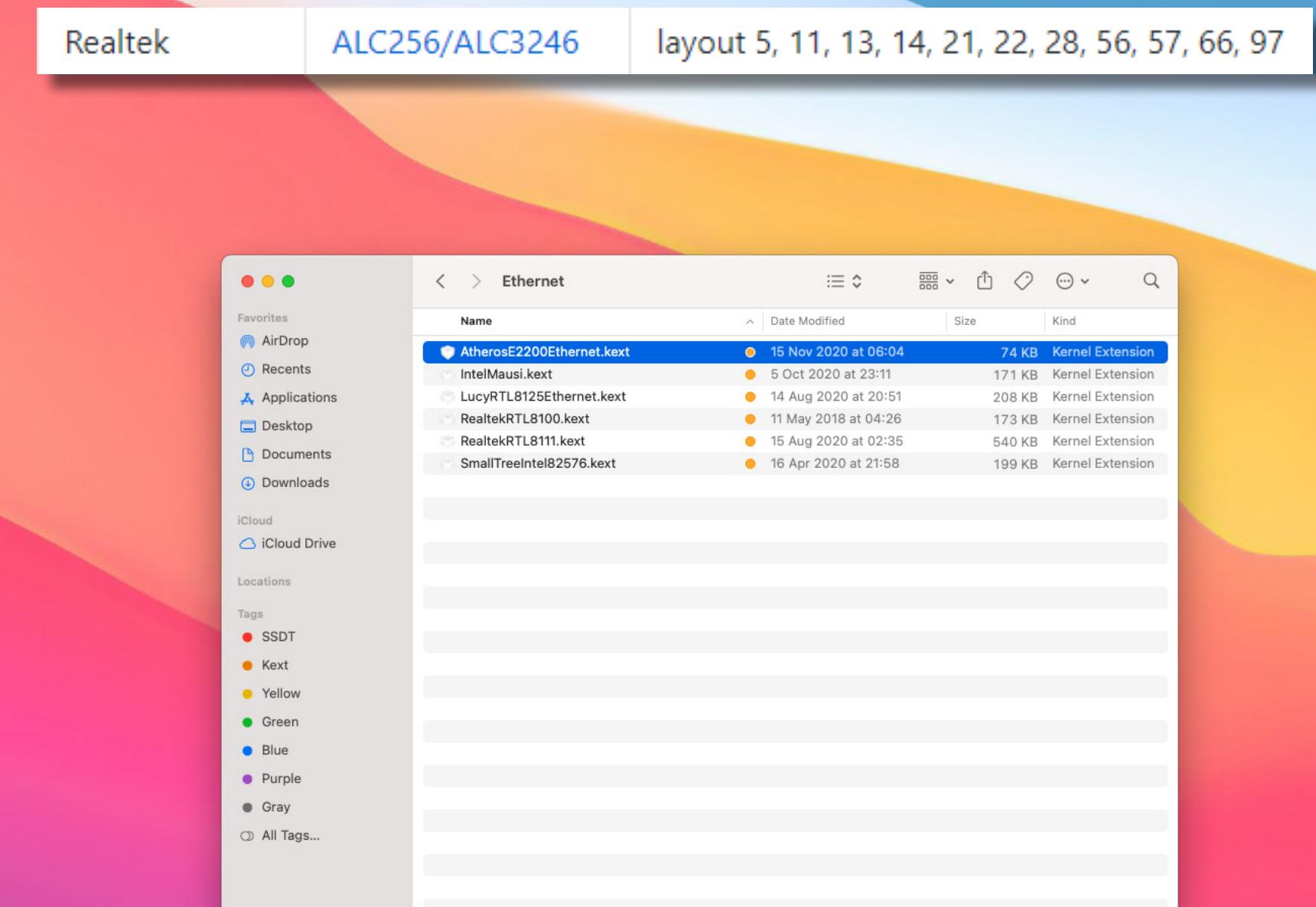
<https://github.com/acidanthera/AppleALC/blob/master/Resources/xxxxxx/Info.plist>

Replace xxxxxx with your codec, like ALC255, ALC256 or CX8200 (mainly see on some Asus laptop machines)

➡ About Ethernet, grab the correct kext from **MATERIALS > Ethernet folder**

For example, with Killer E2400, I use **AtherosE2200Ethernet.kext** because it supports Atheros and Killer NICs.

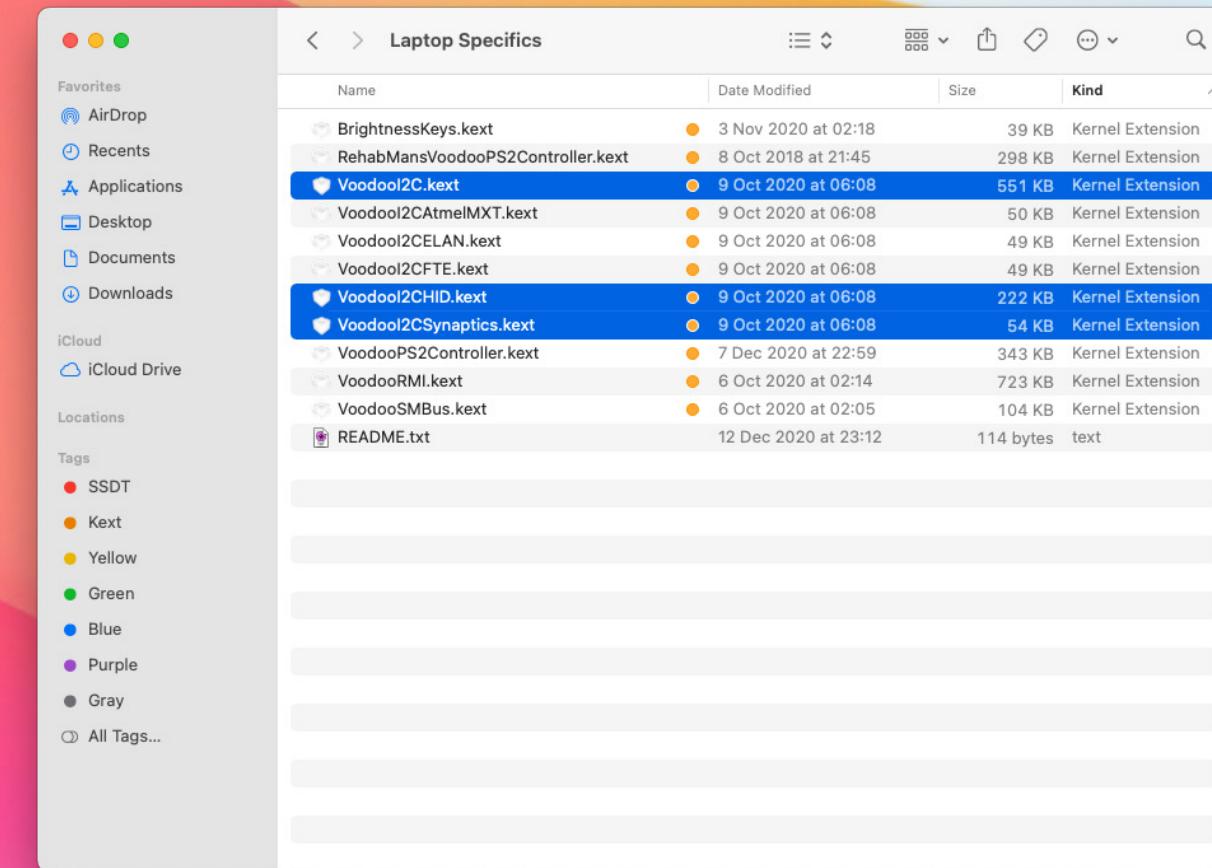
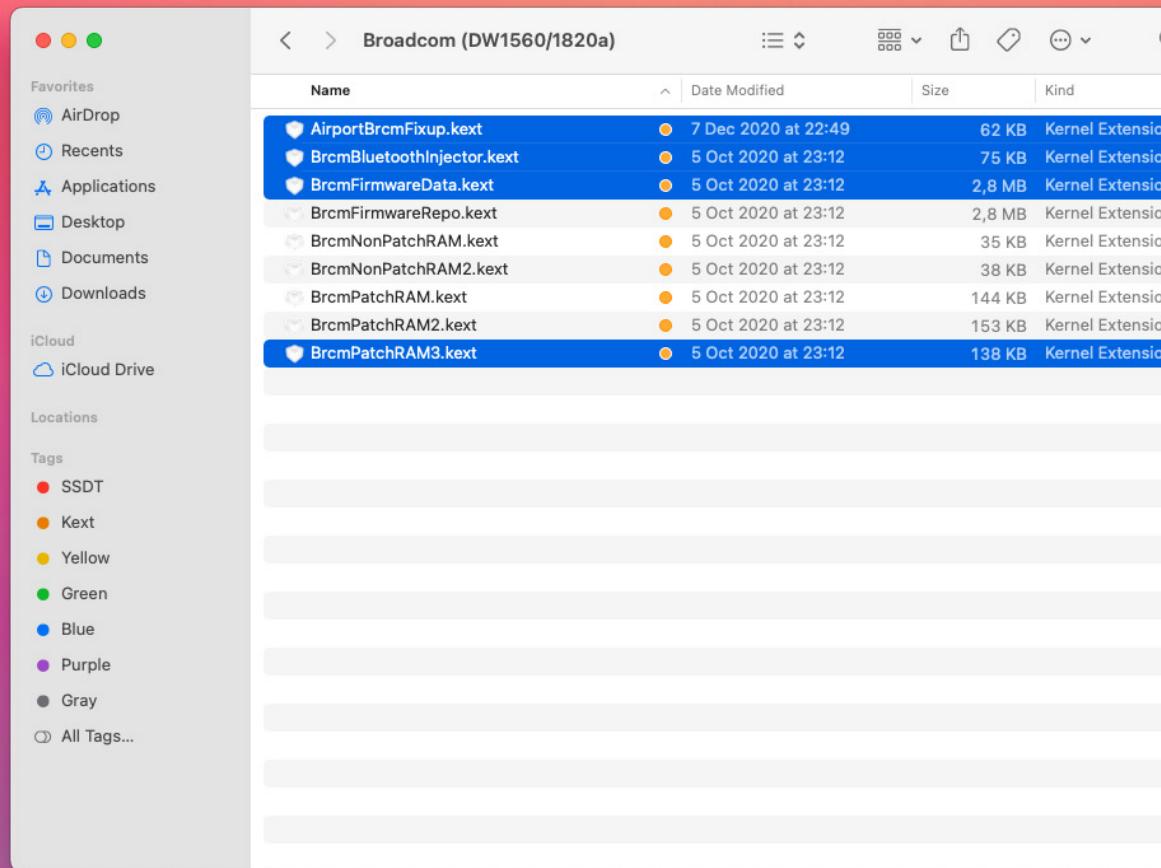
**Note:** Killer E2500 models are actually Realtek based, for these systems you should use RealtekRTL8111.



# KEXT-GATHERING

➡️ About Wireless, this DW1560 wifi card (based BCM4352 chipset) is supported on macOS Big Sur latest, using **AirportBrcmFixup.kext** (with bluetooth, using **BrcmPatchRAM.kext**)  
 For more information, check here: <https://dortania.github.io/Wireless-Buyers-Guide/>  
 Note: Currently most Intel Wi-Fi cards are supported. They are not completely perfect but can be a replacement of USB Wi-Fi Dongles  
 For more information, check here: <https://github.com/OpenIntelWireless>

➡️ About Keyboard and Mouse, on my Dell, it shows Keyboard is HID, PS/2; Mouse is HID.  
 That means with Keyboard, I will grab **VoodooPS2Controller.kext** (on most laptop models, the keyboard is always PS/2). With Mouse, it actually is the Touchpad, I will grab **VoodooI2C.kext** and its plugin **VoodooI2CHID.kext** to get touchpad works.



# KEXT-GATHERING

After finish, the complete Kexts folder should be like that.

In the picture is example about my Dell G7 7588.

I will explain a bit about those kexts:

## 1. AirportBrcmFixup

- For Wi-Fi with BCM card.

## 2. AppleALC

- For audio

## 3. AtherosE2200Ethernet

- For ethernet.

## 4. Combo 3 Brcm kexts

- For bluetooth

## 5. Lilu

- Essential kext (must have).

- For Desktop, it's easier. You only need 4 essential kext Lilu, VirtualSMC, AppleALC and WhateverGreen. Beside, get **USBInjectAll** for USB ports (plus **XHCI-unsupported** if need), **SMCProcessor** monitoring CPU temperature if you want.
- For AMD system, I created a folder name **AMD AREA**. It's for AMD only. Only one thing you do left is find the info about Ethernet and grab the correct kext.

## 6. NoTouchID

- Disable finger sensor, not worked on Hackintosh.

## 7. VirtualSMC (must have) + SMC plugins

- Emulates the SMC chip, hardware monitoring.

## 8. USBInjectAll

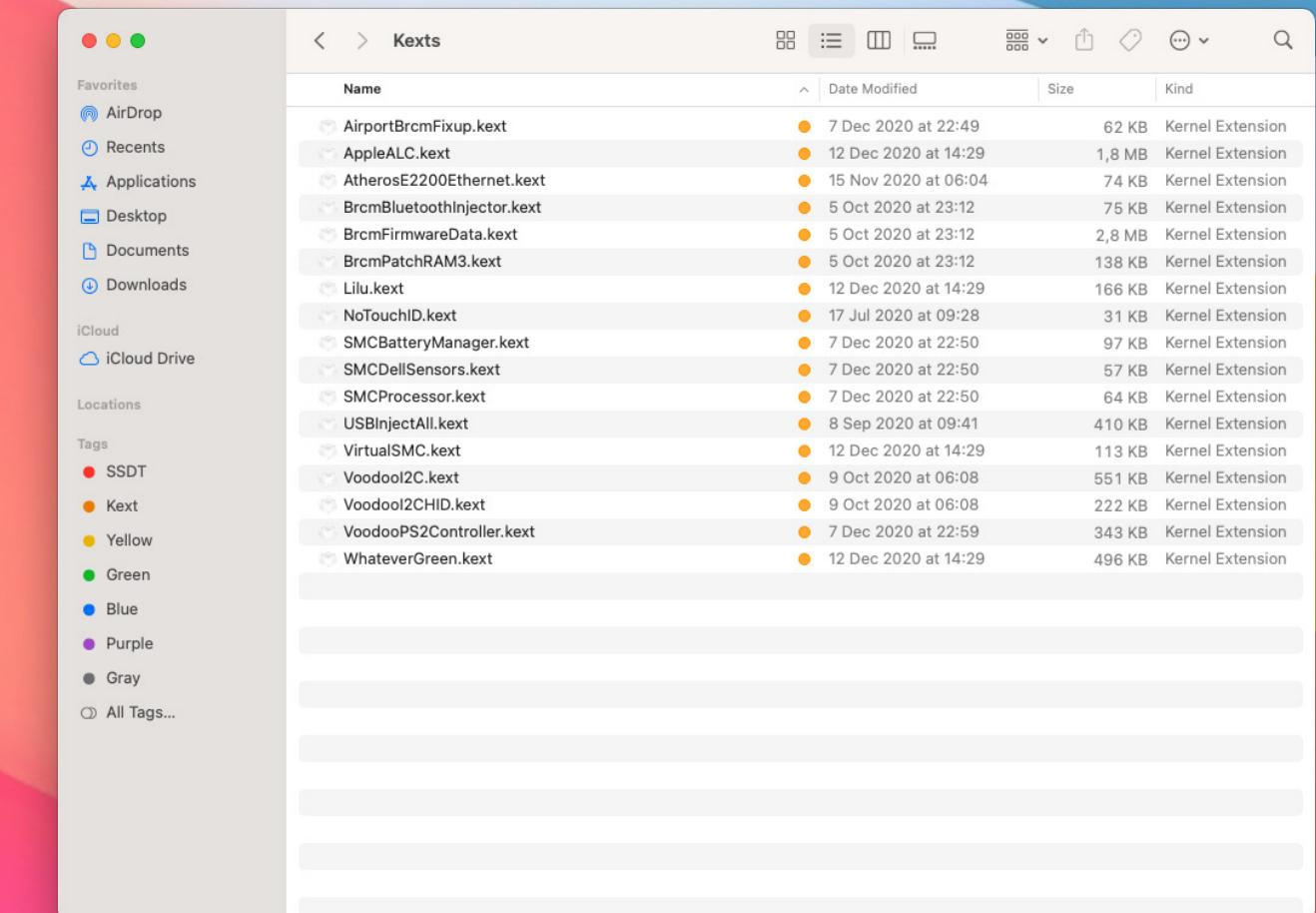
- Injecting Intel USB controllers on systems without defined USB ports in ACPI.

## 8. VoodooPS2Controller

- For keyboard.

## 9. VoodooI2C + I2CHID

- For touchpad.



NOW KEXT IS DONE, HEAD TO GRAB SSDT!

# SSDT-GATHERING

## PREPARATION

- This step is very simple. Just grab the correct SSDT which is corresponding to the CPU.

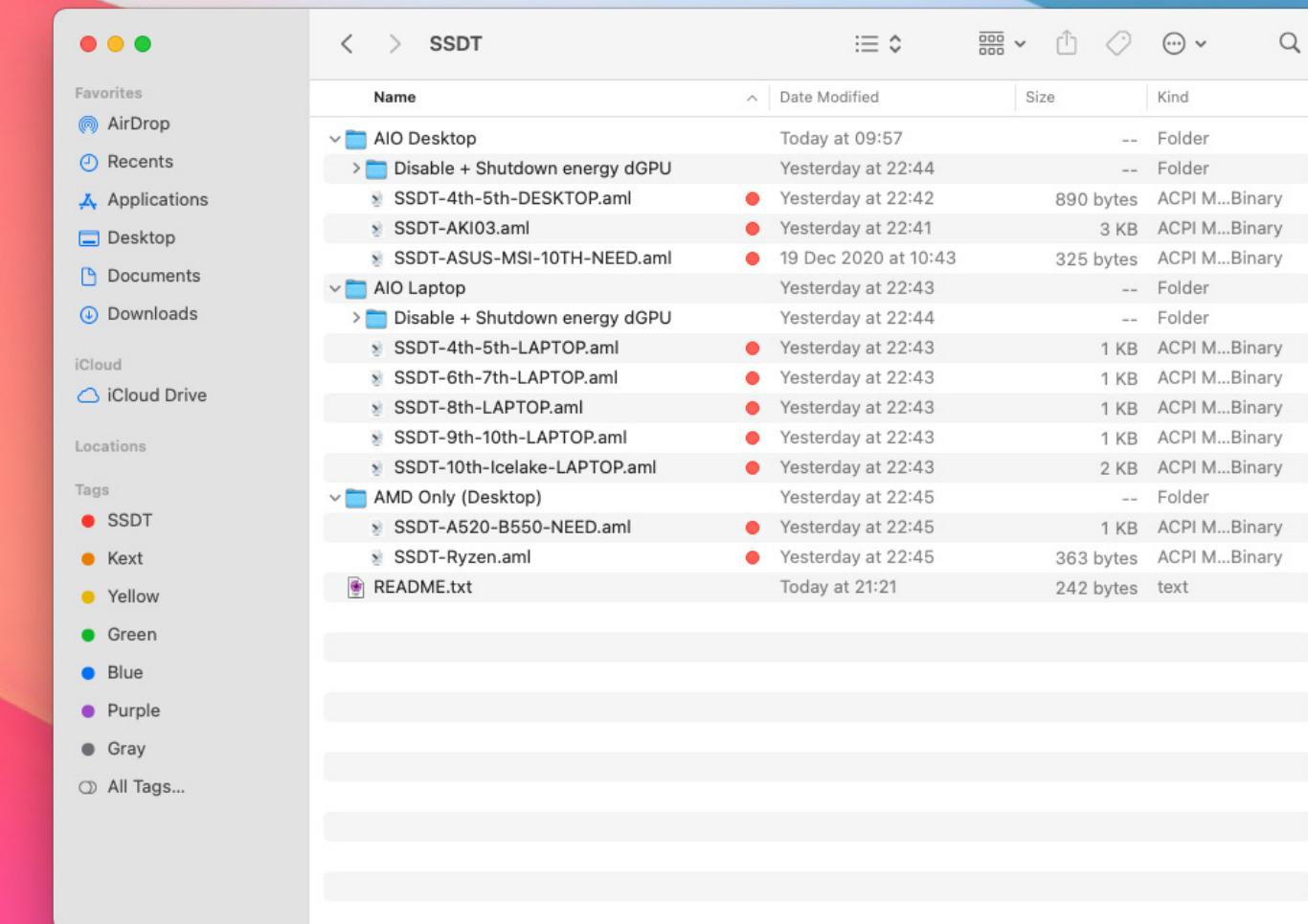
- Desktop users:

1. Haswell and Broadwell: Use **SSDT-4th-5th-DESKTOP.aml**.
2. Skylake to Comet Lake: Use **SSDT-AKI03.aml**.

Note: For somebody who is using Intel Gen 10th ASUS and MSI motherboard (chipset LGA1200), you should need **SSDT-ASUS-MSI-10TH-NEED.aml**. too. It is RHUB SSDT.

- Laptop users: Take correct SSDT.

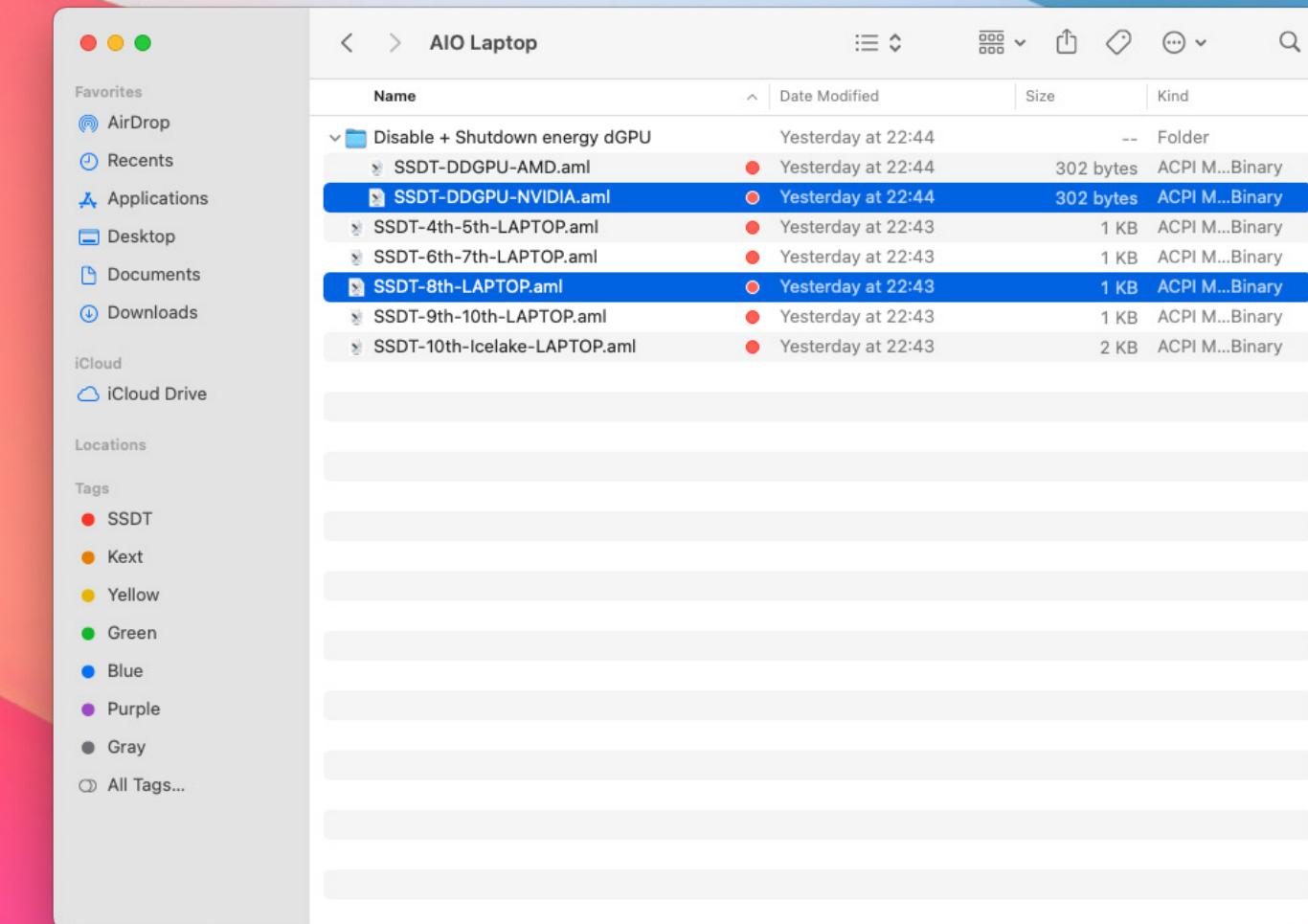
Then put it in **OC/ACPI** folder.



# SSDT-GATHERING

For example, with the G7, I will take the SSDT for 8th Gen laptop. Also SSDT for disabling dGPU NVIDIA 1050Ti (not worked)

Note: Now on most laptop models always have dGPU (AMD or NVIDIA), and almost them are not worked on Hackintosh (except some special machines like Dell Presision M4800, Dell Alienware,...). So beside the SSDT for CPU, you should need the SSDT for disabling the dGPU.



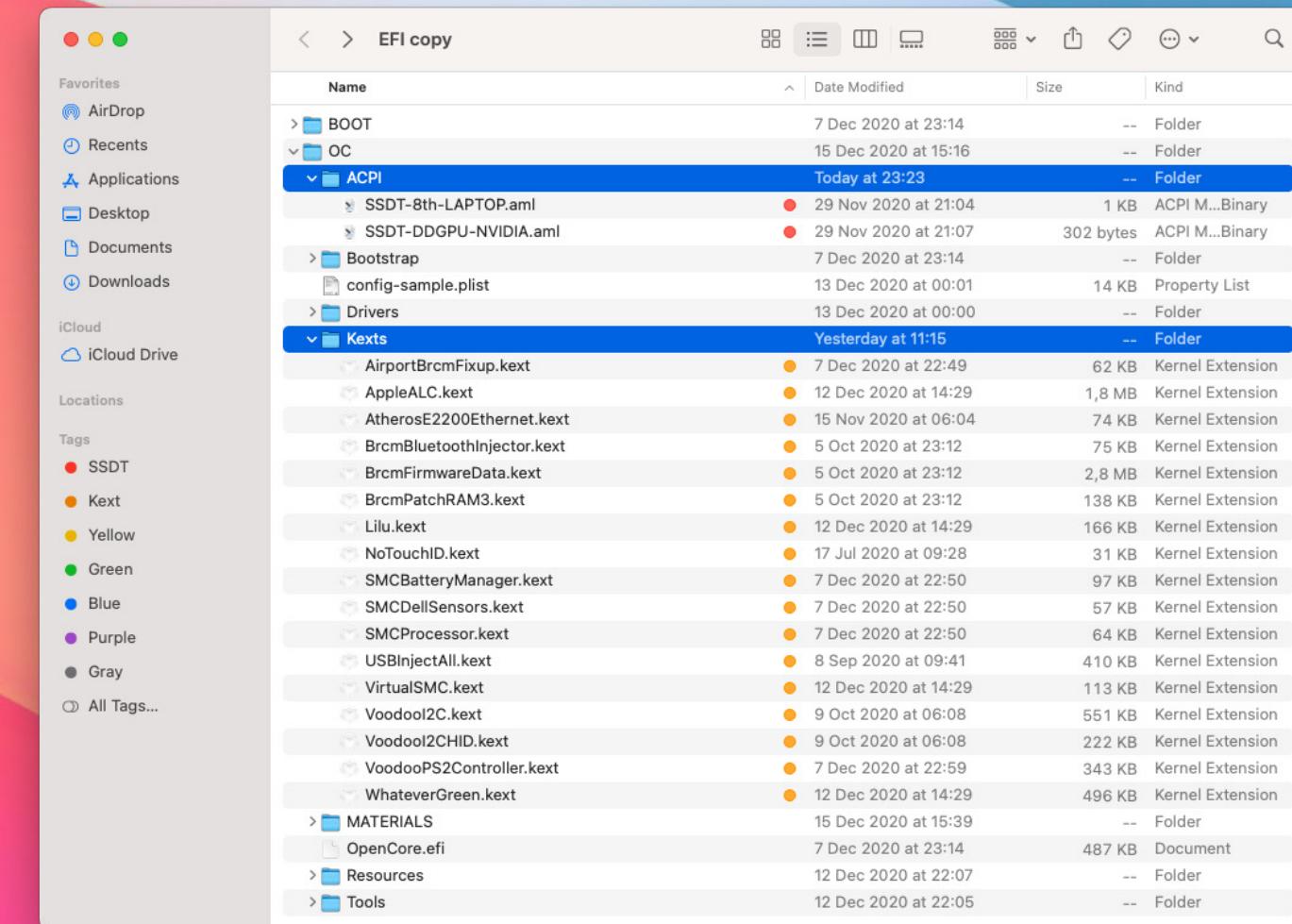
# SSDT-GATHERING

When done, the complete folder should be like that.

The picture is an example OC folder for my Dell G7 7588:

- CPU: Intel Core i7-8750H
- dGPU: NVIDIA 1050Ti (disabled using SSDT)

**NOW THE PREPARATION IS DONE, HEAD TO CONFIGURE CONFIG.PLIST FILE!**



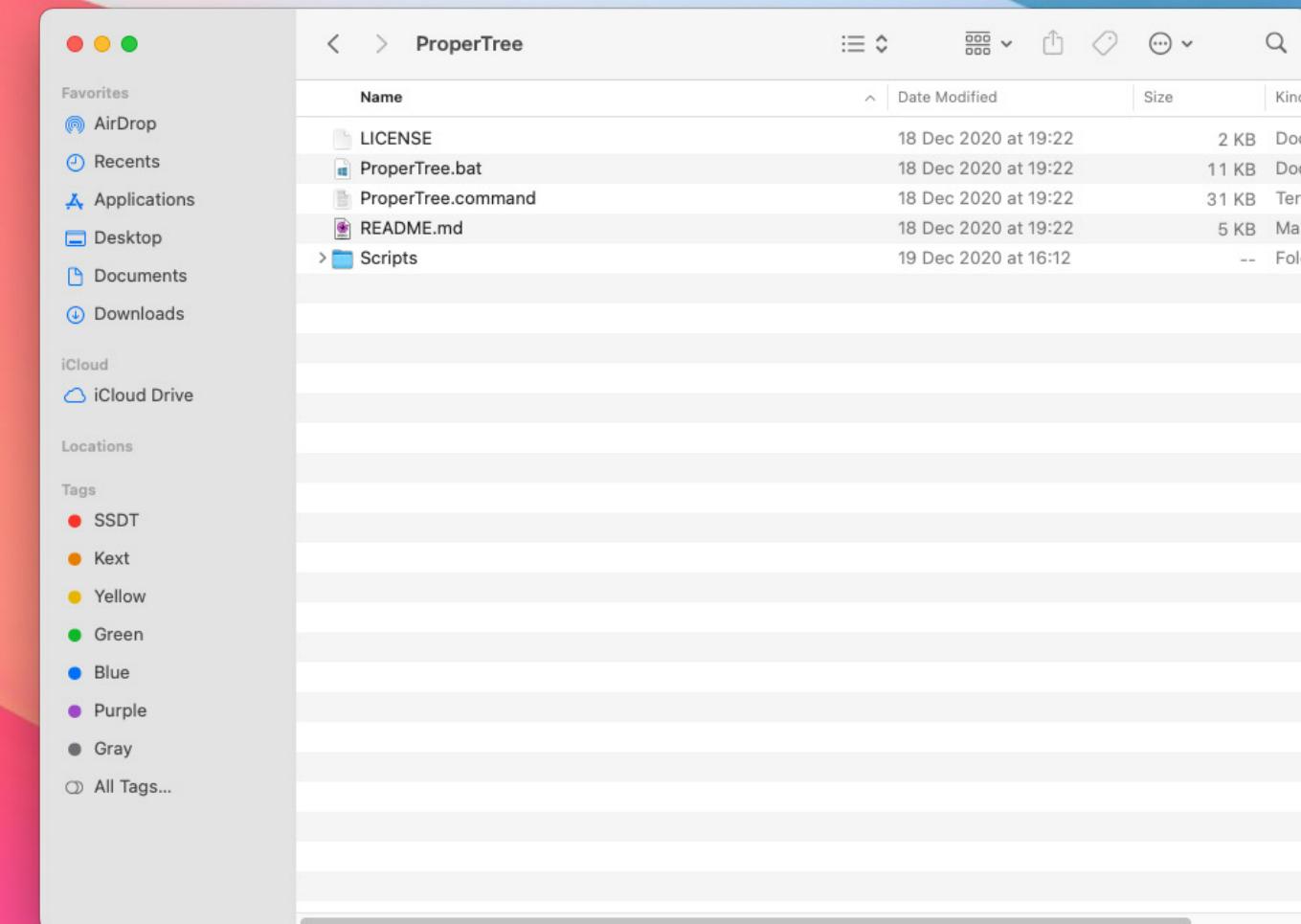
# CONFIGURE- CONFIG.PLIST

On Windows, the best GUI plist editor I used to use is **ProperTree** (**Author**: CorpNewt).

Download: <https://github.com/corpnewt/ProperTree>

- Run **ProperTree.bat**. It will require to install Python if Py hasn't been installed on your machine yet. Install Python and done.

Note: For Mac user, run **ProperTree.command** instead.



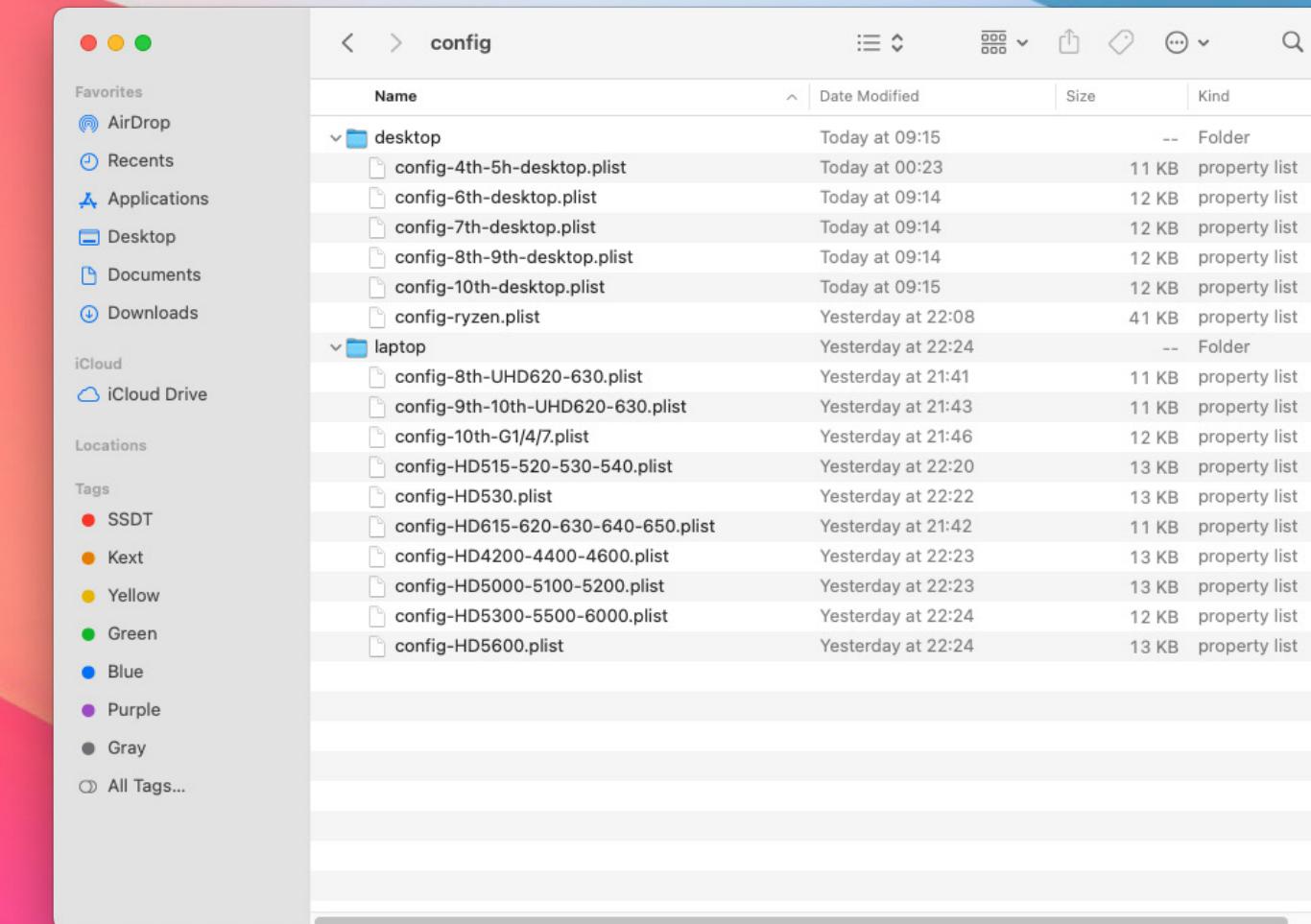
# CONFIGURE- CONFIG.PLIST

- First, go to **OC/MATERIALS/config** folder, find the correct file and copy it, then paste to **OC** folder.

- Rename file to **config.plist**.  
ex: **config-ryzen.plist** -> **config.plist**.

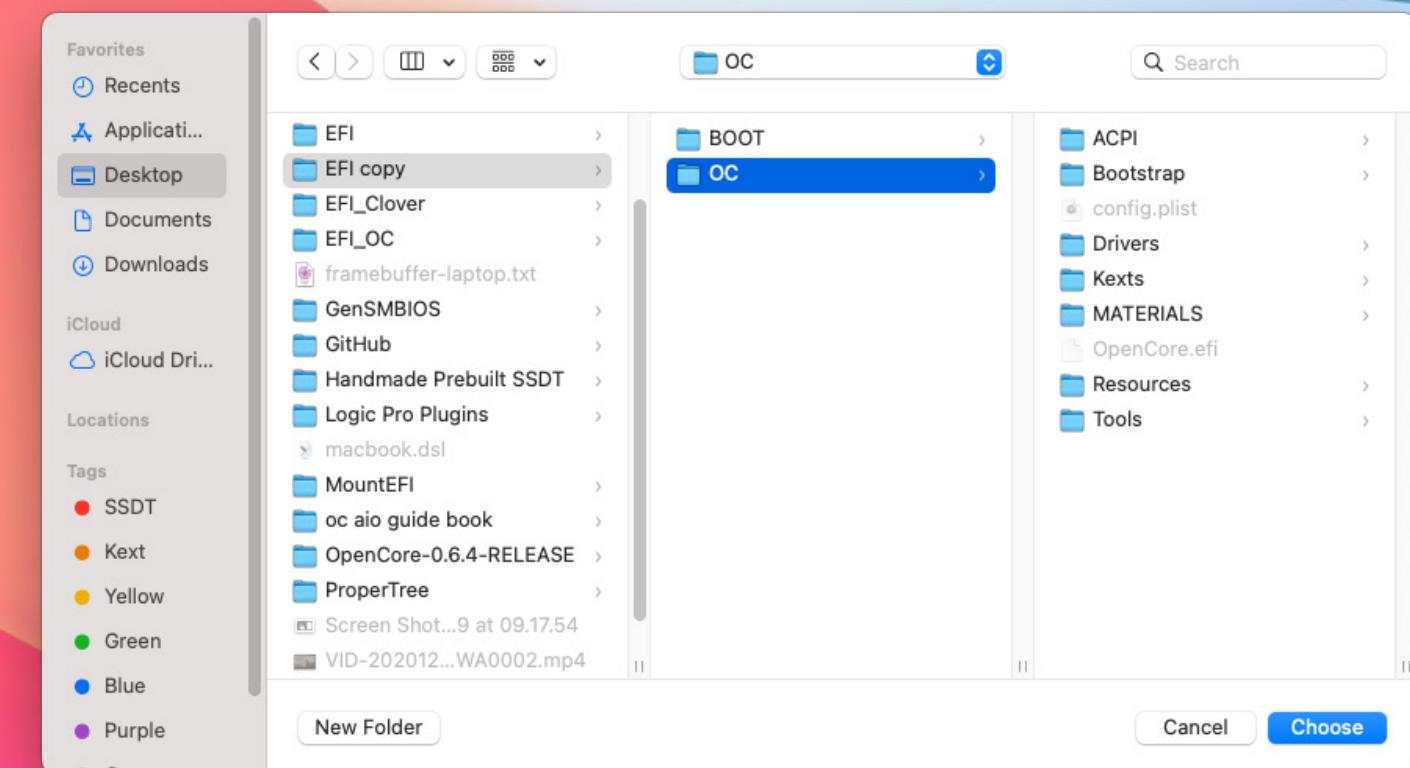
- Back to ProperTree, press **Ctrl + O** (Mac user is **Cmd + O**) to open Open dialog.

- Find the **config.plist** file which is renamed before and choose it.



# CONFIGURE- CONFIG.PLIST

- Press **Ctrl + Shift + R** (Mac user is **Cmd + Shift + R**)
- Another dialog will show up, choose the **OC folder which has the selected config.plist file** before.
- Press Choose



# CONFIGURE- CONFIG.PLIST

- After snapshoting, **Ctrl + S** (Mac) is **Cmd + S**) to save the file.

- This file I have already changed some necessary keys. If you have any issues when booting, you should go to the Dortania website, choose your platform and check again. Mostly the following things:

1. Booter -> Quirks
2. Kernel -> Quirks
3. UEFI -> Quirks

[https://dortania.github.io/  
OpenCore-Install-Guide/config.  
plist/#selecting-your-platform](https://dortania.github.io/OpenCore-Install-Guide/config.plist/#selecting-your-platform)

- Troubleshooting: [https://  
dortania.github.io/OpenCore-  
Install-Guide/troubleshooting/  
troubleshooting.html](https://dortania.github.io/OpenCore-Install-Guide/troubleshooting/troubleshooting.html)

- The **CustomSMBIOSGuid** and **UpdateSMBIOSMode** should be left default like I set. These will let OpenCore doesn't inject SMBIOS info to other OSes.

/Users/mizuhara/Desktop/EFI copy/OC/config.plist - Edited

Key	Type	Value
Root	: Dictionary	8 key/value pairs
ACPI	: Dictionary	4 key/value pairs
Add	: Array	2 children
0	: Dictionary	3 key/value pairs
Comment	: String	SSDT-8th-LAPTOP.aml
Path	: String	SSDT-8th-LAPTOP.aml
Enabled	: Boolean	True
1	: Dictionary	3 key/value pairs
Comment	: String	SSDT-DDGPU-NVIDIA.aml
Path	: String	SSDT-DDGPU-NVIDIA.aml
Enabled	: Boolean	True
Delete	: Array	0 children
Patch	: Array	1 child
0	: Dictionary	12 key/value pairs
Comment	: String	Fix RTC _STA bug
Count	: Number	0
Enabled	: Boolean	True
Find	: Data	<A00A9353 54415301>
Limit	: Number	0
Mask	: Data	<>
OemTableId	: Data	<>
Replace	: Data	<A00A910A FF0BFFFF>
ReplaceMask	: Data	<>
Skip	: Number	0
TableLength	: Number	0
TableSignature	: Data	<44534454>
Quirks	: Dictionary	5 key/value pairs
FadtEnableReset	: Boolean	False
NormalizeHeaders	: Boolean	False
RebaseRegions	: Boolean	False
ResetHwSig	: Boolean	False
ResetLogoStatus	: Boolean	False
Booter	: Dictionary	3 key/value pairs
MmioWhitelist	: Array	0 children
Patch	: Array	0 children
Quirks	: Dictionary	18 key/value pairs
AllowRelocationBlock	: Boolean	False
AvoidRuntimeDefrag	: Boolean	True
DevirtualiseMmio	: Boolean	False
DisableSingleUser	: Boolean	False
DisableVariableWrite	: Boolean	False
DiscardHibernateMap	: Boolean	False
EnableSafeModeSlide	: Boolean	True

# • CONFIG.PLIST- CONFIGURE

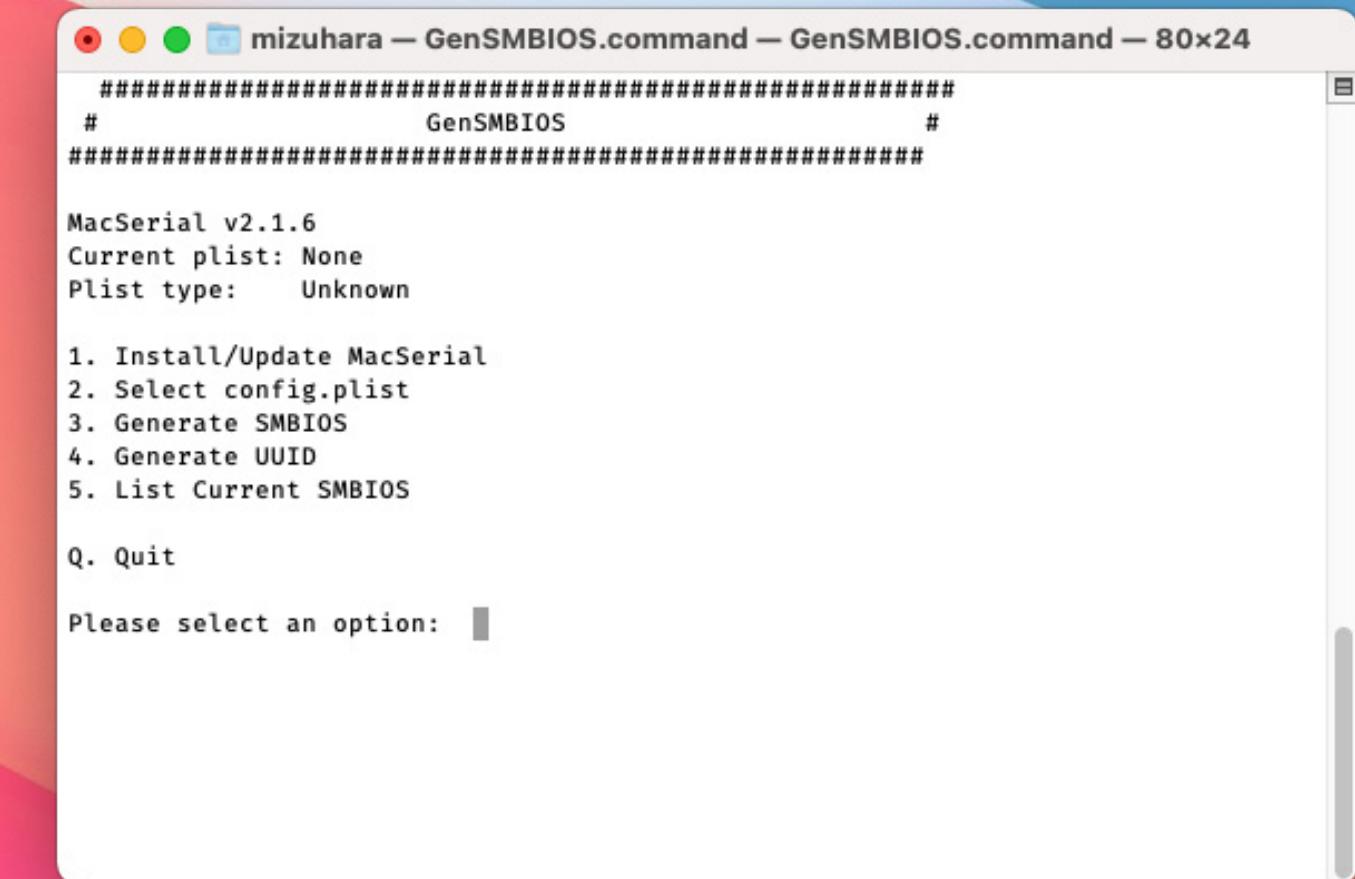
- Next, in this step we will get the SMBIOS infomation (can help to get FaceTime/iMessenger worked). In this guide I use **GenSMBIOS** (**Author:** CorpNewt)

Download: <https://github.com/corpnewt/GenSMBIOS>

- Extract zip file, then open **GenSMBIOS.bat** (Windows user) or **GenSMBIOS.command** (Mac user)

- Type 1 and hit Enter to install MacSerial library.

- After installed, it will show the version of MacSerial. Currently is 2.1.6.



```
mizuhara — GenSMBIOS.command — GenSMBIOS.command — 80x24
#####
#          GenSMBIOS      #
#####

MacSerial v2.1.6
Current plist: None
Plist type: Unknown

1. Install/Update MacSerial
2. Select config.plist
3. Generate SMBIOS
4. Generate UUID
5. List Current SMBIOS

Q. Quit

Please select an option: █
```

# CONFIGURE- CONFIG.PLIST

- Type the SMBIOS which you want to generate. For example in the picture, I want to generate iMacPro1,1 SMBIOS.

```
mizuhara — GenSMBIOS.command — GenSMBIOS.command — 80x24
#####
#          Generate SMBIOS      #
#####

M. Main Menu
Q. Quit

Please type the SMBIOS to gen and the number
of times to generate [max 20] (i.e. iMac18,3 5): iMacPro1,1
```

- Then I got its info.

```
mizuhara — GenSMBIOS.command — GenSMBIOS.command — 80x24
#####
#          iMacPro1,1 SMBIOS Info      #
#####

Type:      iMacPro1,1
Serial:    C02ZMCZZHX87
Board Serial: C02944405QXJG361F
SmUUID:   9BFFE32B-7C2C-451C-A4DA-63EFC5545EA6

Press [enter] to return ...
```

# CONFIG- CONFIG.PLIST

- Now you should fill the SMBIOS which got from GenSMBIOS to config.plist file.

- Open the config.plist file with ProperTree.

- Go **PlatformInfo -> Generic**:

1. Type = SystemProductName: iMacPro1,1
2. Serial = SystemSerialNumber: C02ZMCZZHX87
3. Board Serial = MLB: C 02944405QXJG361F
4. SmUUID = SystemUUID: 9BFFE32B-7C2C-451C-A4DA-63EFC5545EA6

- Save config.plist file. We're done!

```
mizuhara — GenSMBIOS.command — GenSMBIOS.command — 80x24
#####
#          iMacPro1,1 SMBIOS Info      #
#####

Type:      iMacPro1,1
Serial:    C02ZMCZZHX87
Board Serial: C02944405QXJG361F
SmUUID:   9BFFE32B-7C2C-451C-A4DA-63EFC5545EA6

Press [enter] to return ...
```

/Users/mizuhara/Desktop/OpenCore-0.6.4-RELEASE/Docs/Sample.plist - Edited

Key	Type	Value
Root	Dictionary	8 key/value pairs
> ACPI	Dictionary	4 key/value pairs
> Booter	Dictionary	3 key/value pairs
> DeviceProperties	Dictionary	2 key/value pairs
> Kernel	Dictionary	7 key/value pairs
> Misc	Dictionary	6 key/value pairs
> NVRAM	Dictionary	6 key/value pairs
> PlatformInfo	Dictionary	7 key/value pairs
Automatic	Boolean	True
CustomMemory	Boolean	False
> Generic	Dictionary	9 key/value pairs
AdvertiseWindows	Boolean	False
SystemMemoryStatus	String	Auto
MLB	String	C02944405QXJG361F
ProcessorType	Number	0
ROM	Data	<11223344 5566>
SpoofVendor	Boolean	True
SystemProductName	String	iMacPro1,1
SystemSerialNumber	String	C02ZMCZZHX87
SystemUUID	String	9BFFE32B-7C2C-451C-A4DA-63EFC5545EA6
UpdateDataHub	Boolean	True
UpdateNVRAM	Boolean	True
UpdateSMBIOS	Boolean	True
UpdateSMBIOSMode	String	Create
> UEFI	Dictionary	9 key/value pairs

## CONTACT

twitter @kumakumalina

discord mizuhara#0782

## CREDIT

Apple for macOS

Acidanthera team

Dortania guide

CorpNewt for ProperTree and GenSMBIOS

AMD-OSX team for AMD\_Vanilla guide

Olarila's administration MaLdOn for SSDT-OLARILA.aml