Why OpenCore over Clover and others

USB Creation
Creating the USB
Making the installer in macOS
Making the installer in Windows
Making the installer in Linux
Adding The Base OpenCore Files
Gathering files
Getting started with ACPI
config.plist Setup

Configs
Intel Desktop config.plist
Penryn
Clarkdale
Sandy Bridge
Ivy Bridge
Haswell
Skylake
Kaby Lake
Coffee Lake
Comet Lake
Starting Point
ACPI
Booter
DeviceProperties
Kernel
Misc
NVRAM
PlatformInfo
UEFI
Cleaning up
Intel BIOS settings
Intel Laptop config.plist
Intel HEDT config.plist
AMD Desktop config.plist
Apple Secure Boot

Installation
Installation Process

Troubleshooting
General Troubleshooting
OpenCore Boot Issues
Kernel Issues
Userspace Issues
Post-Install Issues
Miscellaneous Issues
OpenCore Debugging
Understanding the macOS Boot Process
System Debugging: In-depth

Post Install
Post-Install
Universal
Laptop Specifics
Cosmetics
Multiboot
Miscellaneous

Extras
Fixing KASLR slide values
Disabling GPU
macOS 13: Ventura
Clover Conversion
Choosing the right SMBIOS

Misc
Supporting the guides
Credits

Desktop Comet Lake

Support	Version
Initial macOS Support	macOS 10.15, Catalina

Starting Point

So making a config.plist may seem hard, it's not. It just takes some time but this guide will tell you how to configure everything, you won't be left in the cold. This also means if you have issues, review your config settings to make sure they're correct. Main things to note with OpenCore:

- **All properties must be defined**, there are no default OpenCore will fall back on so **do not delete sections unless told explicitly so**. If the guide doesn't mention the option, leave it at default.
- **The Sample.plist cannot be used As-Is**, you must configure it to your system
- **DO NOT USE CONFIGURATORS**, these rarely respect OpenCore's configuration and even some like Mackie's will add Clover properties and corrupt plists!

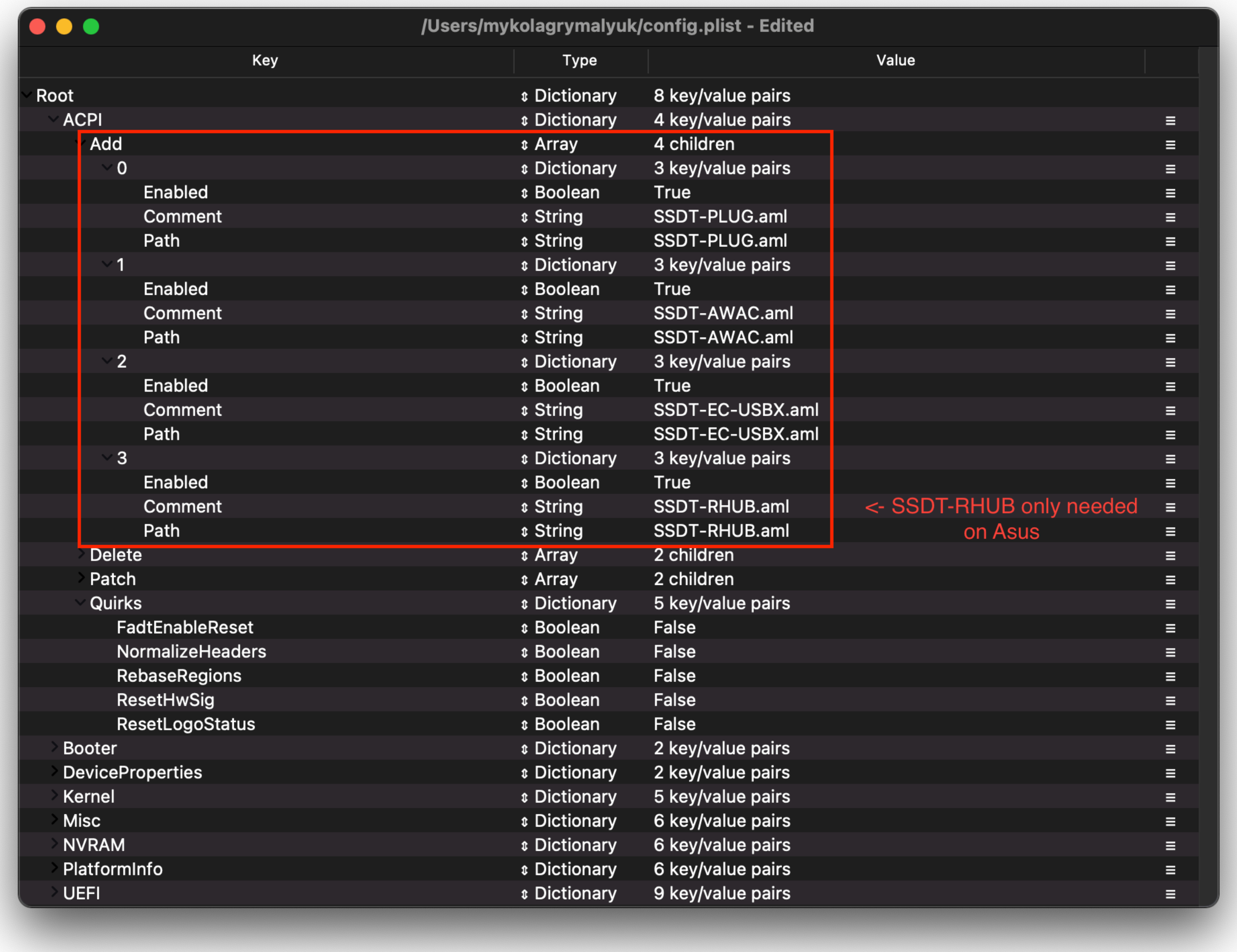
Now with all that, a quick reminder of the tools we need

- ProperTree
  - Universal plist editor
- GenSMBIOS
  - For generating our SMBIOS data
- Sample/config.plist
  - See previous section on how to obtain: config.plist Setup

WARNING

Read this guide more than once before setting up OpenCore and make sure you have it set up correctly. Do note that images will not always be the most up-to-date so please read the text below them, if nothing's mentioned then leave as default.

ACPI



/Users/mykolagrymalyuk/config.plist - Edited

Key	Type	Value
Root	Dictionary	8 key/value pairs
ACPI	Dictionary	4 key/value pairs
Add	Array	4 children
0	Dictionary	3 key/value pairs
Enabled	Boolean	True
Comment	String	SSDT-PLUG.aml
Path	String	SSDT-PLUG.aml
1	Dictionary	3 key/value pairs
Enabled	Boolean	True
Comment	String	SSDT-AWAC.aml
Path	String	SSDT-AWAC.aml
2	Dictionary	3 key/value pairs
Enabled	Boolean	True
Comment	String	SSDT-EC-USBX.aml
Path	String	SSDT-EC-USBX.aml
3	Dictionary	3 key/value pairs
Enabled	Boolean	True
Comment	String	SSDT-RHUB.aml
Path	String	SSDT-RHUB.aml
Delete	Array	2 children
Patch	Array	2 children
Quirks	Dictionary	5 key/value pairs
FadtEnableReset	Boolean	False
NormalizeHeaders	Boolean	False
RebaseRegions	Boolean	False
ResetHwSig	Boolean	False
ResetLogoStatus	Boolean	False
Booter	Dictionary	2 key/value pairs
DeviceProperties	Dictionary	2 key/value pairs
Kernel	Dictionary	5 key/value pairs
Misc	Dictionary	6 key/value pairs
NVRAM	Dictionary	6 key/value pairs
PlatformInfo	Dictionary	6 key/value pairs
UEFI	Dictionary	9 key/value pairs

<- SSDT-RHUB only needed on Asus

Add

Info

This is where you'll add SSDTs for your system, these are very important to **booting macOS** and have many uses like USB maps , disabling unsupported GPUs and such. And with our system, **it's even required to boot**. Guide on making them found here: **Getting started with ACPI**

For us we'll need a couple of SSDTs to bring back functionality that Clover provided:

Required SSDTs	Description
SSDT-PLUG	Allows for native CPU power management on Haswell and newer, see Getting Started With ACPI Guide for more details.
SSDT-EC-	Fixes both the embedded controller and USB power, see Getting Started With ACPI Guide

USBX	for more details.
SSDT-AWAC	This is the 300 series RTC patch, required for all B460 and Z490 boards which prevent systems from booting macOS. The alternative is SSDT-RTC0 for when AWAC SSDT is incompatible due to missing the Legacy RTC clock, to check whether you need it and which to use please see Getting started with ACPI page.
SSDT-RHUB	Needed to fix Root-device errors on Asus and potentially MSI boards. Gigabyte and AsRock motherboards **do not** need this SSDT

Note that you **should not** add your generated `DSDT.aml` here, it is already in your firmware. So if present, remove the entry for it in your `config.plist` and under EFI/OC/ACPI.

For those wanting a deeper dive into dumping your DSDT, how to make these SSDTs, and compiling them, please see the Getting started with ACPI **page.** Compiled SSDTs have a **.aml** extension(Assembled) and will go into the `EFI/OC/ACPI` folder and **must** be specified in your config under `ACPI -> Add` as well.

Delete

This blocks certain ACPI tables from loading, for us we can ignore this.

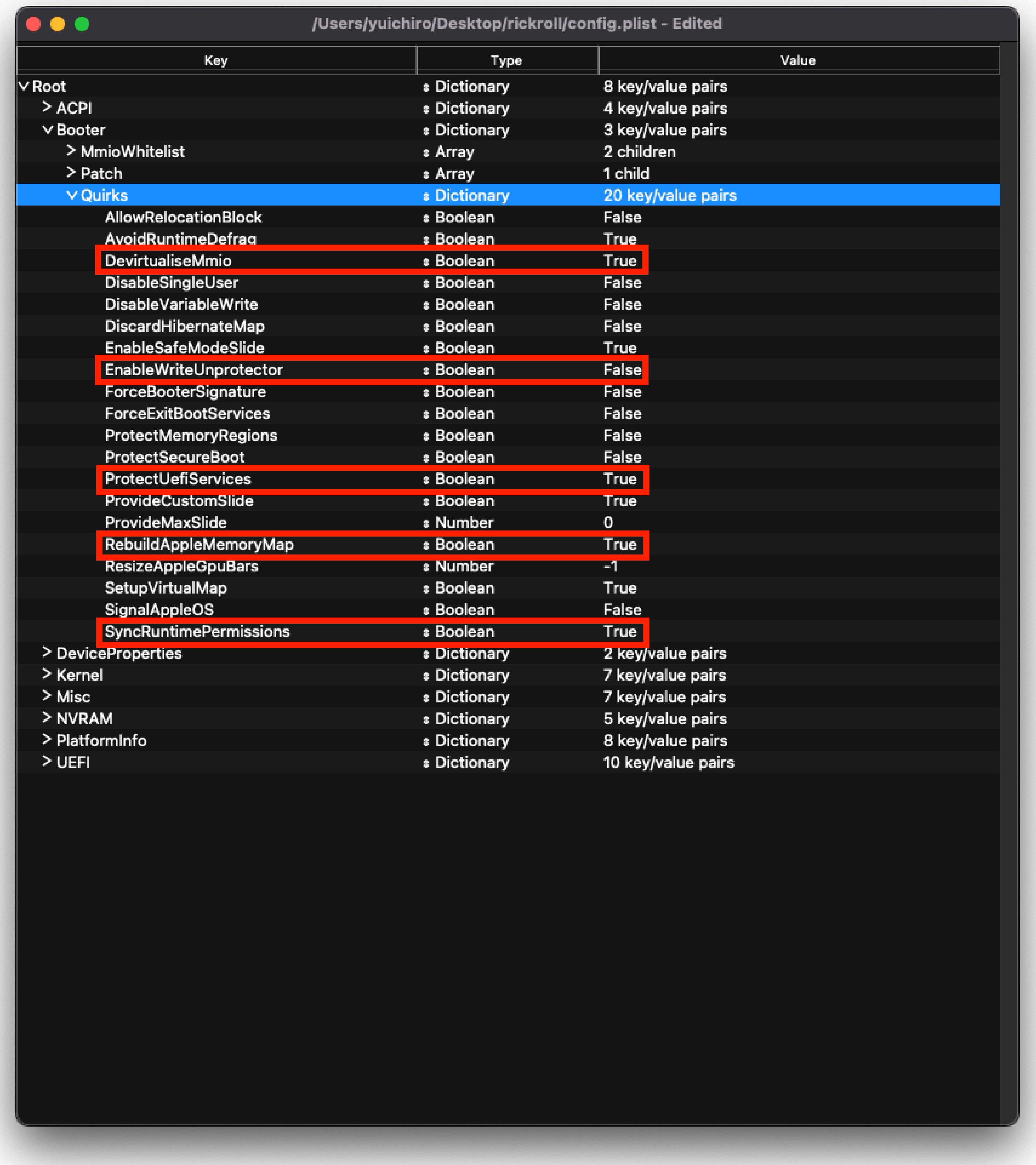
Patch

This section allows us to dynamically modify parts of the ACPI (DSDT, SSDT, etc.) via OpenCore. For us, our patches are handled by our SSDTs. This is a much cleaner solution as this will allow us to boot Windows and other OSes with OpenCore

Quirks

Settings relating to ACPI, leave everything here as default as we have no use for these quirks.

Booter



/Users/yuichiro/Desktop/rickroll/config.plist - Edited

Key	Type	Value
∨ Root	Dictionary	8 key/value pairs
> ACPI	Dictionary	4 key/value pairs
∨ Booter	Dictionary	3 key/value pairs
> MmioWhitelist	Array	2 children
> Patch	Array	1 child
∨ Quirks	Dictionary	20 key/value pairs
AllowRelocationBlock	Boolean	False
AvoidRuntimeDefrag	Boolean	True
DevirtualiseMmio	Boolean	True
DisableSingleUser	Boolean	False
DisableVariableWrite	Boolean	False
DiscardHibernateMap	Boolean	False
EnableSafeModeSlide	Boolean	True
EnableWriteUnprotector	Boolean	False
ForceBooterSignature	Boolean	False
ForceExitBootServices	Boolean	False
ProtectMemoryRegions	Boolean	False
ProtectSecureBoot	Boolean	False
ProtectUefiServices	Boolean	True
ProvideCustomSlide	Boolean	True
ProvideMaxSlide	Number	0
RebuildAppleMemoryMap	Boolean	True
ResizeAppleGpuBars	Number	-1
SetupVirtualMap	Boolean	True
SignalAppleOS	Boolean	False
SyncRuntimePermissions	Boolean	True
> DeviceProperties	Dictionary	2 key/value pairs
> Kernel	Dictionary	7 key/value pairs
> Misc	Dictionary	7 key/value pairs
> NVRAM	Dictionary	5 key/value pairs
> PlatformInfo	Dictionary	8 key/value pairs
> UEFI	Dictionary	10 key/value pairs

This section is dedicated to quirks relating to boot.efi patching with OpenRuntime, the replacement for AptioMemoryFix.efi

MmioWhitelist

This section is allowing devices to be passthrough to macOS that are generally ignored, for us we can ignore this section.

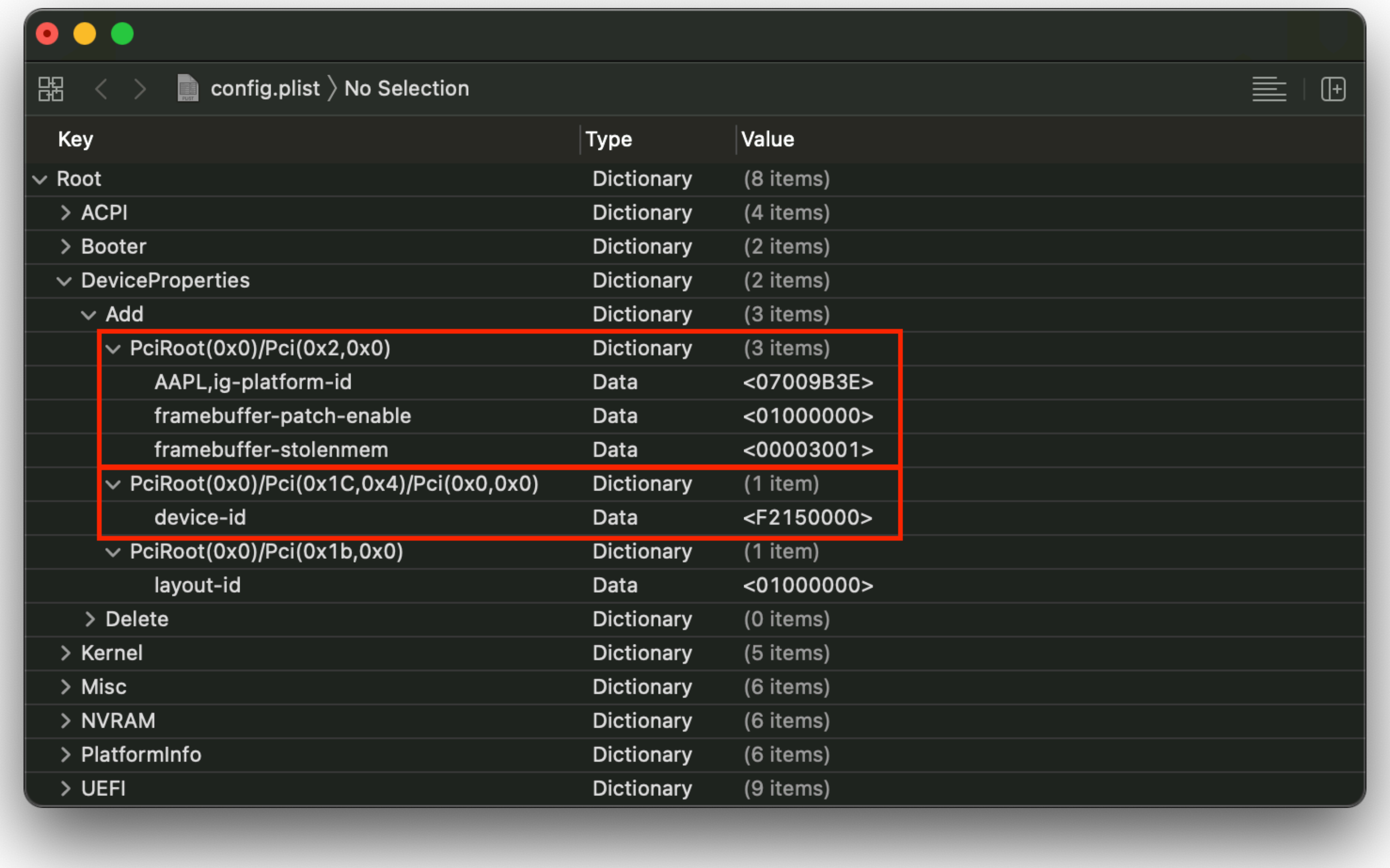
Quirks

Settings relating to boot.efi patching and firmware fixes, for us, we need to change the following:

Quirk	Enabled	Comment
DevirtualiseMmio	YES	
EnableWriteUnprotector	NO	
ProtectUefiServices	YES	
RebuildAppleMemoryMap	YES	
ResizeAppleGpuBars	-1	If your firmware supports increasing GPU Bar sizes (ie Resizable BAR Support), set this to `0`
SetupVirtualMap	NO	
SyncRuntimePermissions	YES	

▶ More in-depth Info

DeviceProperties



```
Key                                          Type        Value
Root                                         Dictionary  (8 items)
  ACPI                                       Dictionary  (4 items)
  Booter                                     Dictionary  (2 items)
  DeviceProperties                           Dictionary  (2 items)
    Add                                      Dictionary  (3 items)
      PciRoot(0x0)/Pci(0x2,0x0)              Dictionary  (3 items)
        AAPL,ig-platform-id                  Data        <07009B3E>
        framebuffer-patch-enable             Data        <01000000>
        framebuffer-stolenmem                Data        <00003001>
      PciRoot(0x0)/Pci(0x1C,0x4)/Pci(0x0,0x0) Dictionary (1 item)
        device-id                            Data        <F2150000>
      PciRoot(0x0)/Pci(0x1b,0x0)             Dictionary  (1 item)
        layout-id                            Data        <01000000>
    Delete                                   Dictionary  (0 items)
  Kernel                                     Dictionary  (5 items)
  Misc                                       Dictionary  (6 items)
  NVRAM                                      Dictionary  (6 items)
  PlatformInfo                               Dictionary  (6 items)
  UEFI                                       Dictionary  (9 items)
```

Add

Sets device properties from a map.

PciRoot(0x0)/Pci(0x2,0x0)

This section is set up via WhateverGreen's Framebuffer Patching Guide and is used for setting important iGPU properties. **If you have a `-F` series CPU, you can ignore this section as you do not have an iGPU.**

The config.plist doesn't already have a section for this so you will have to create it manually.

`AAPL,ig-platform-id` is what macOS uses to determine how the iGPU drivers interact with our system, and two values choose between are as follows:

AAPL,ig-platform-id	Comment
`07009B3E`	Used when the Desktop iGPU is used to drive a display
`00009B3E`	Alternative to `07009B3E` if it doesn't work
`0300C89B`	Used when the Desktop iGPU is only used for computing tasks and doesn't drive a display

- **Note**: With macOS 10.15.5 and newer, there seems to be a lot of issues with black screen using `07009B3E`, if you get similar issues try swapping to `00009B3E`

We also add 2 more properties, `framebuffer-patch-enable` and `framebuffer-stolenmem`. The first enables patching via WhateverGreen.kext, and the second sets the min stolen memory to 19MB. This is usually unnecessary, as this can be configured in BIOS(64MB recommended) but required when not available.

- **Note**: Headless framebuffers(where the dGPU is the display out) do not need `framebuffer-patch-enable` and `framebuffer-stolenmem`

Key	Type	Value
AAPL,ig-platform-id	Data	`07009B3E`

framebuffer-patch-enable	Data	`01000000`
framebuffer-stolenmem	Data	`00003001`

(This is an example for a desktop UHD 630 without a dGPU and no BIOS options for iGPU memory)

PciRoot(0x0)/Pci(0x1C,0x1)/Pci(0x0,0x0)

This entry relates to Intel's I225-V 2.5GBe controller found on higher end Comet Lake boards, what we'll be doing here is tricking Apple's I225LM driver into supporting our I225-V network controller:

Key	Type	Value
device-id	Data	`F2150000`

- **Note**: If your board didn't ship with the Intel I225 NIC, there's no reason to add this entry.
- **Note 2**: If you get a kernel panic on the AppleIntel210Ethernet kext, your Ethernet's path is likely `PciRoot(0x0)/Pci(0x1C,0x4)/Pci(0x0,0x0)`

PciRoot(0x0)/Pci(0x1b,0x0)

`layout-id`

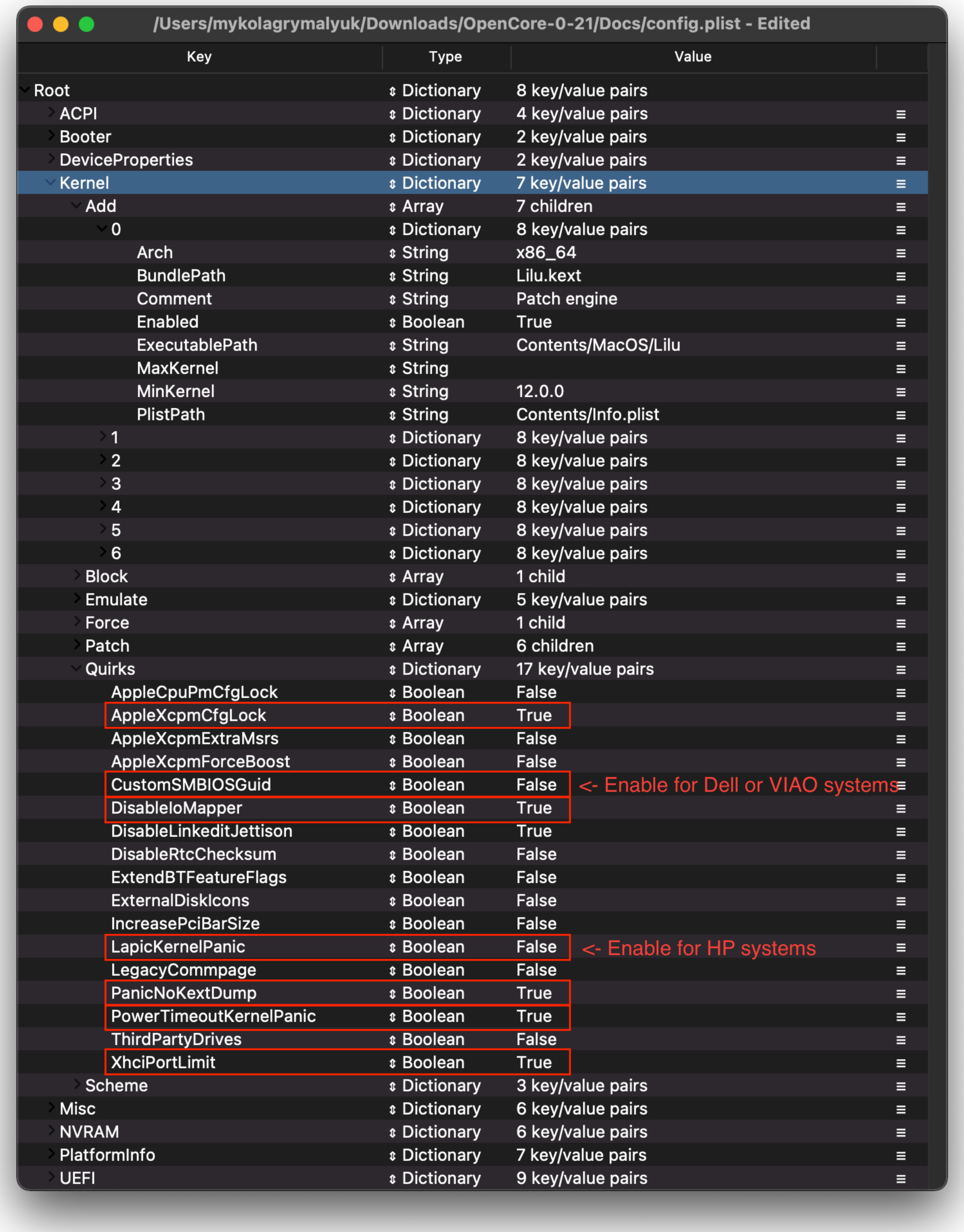
- Applies AppleALC audio injection, you'll need to do your own research on which codec your motherboard has and match it with AppleALC's layout. AppleALC Supported Codecs .
- You can delete this property outright as it's unused for us at this time

For us, we'll be using the boot-arg `alcid=xxx` instead to accomplish this. `alcid` will override all other layout-IDs present. More info on this is covered in the Post-Install Page

Delete

Removes device properties from the map, for us we can ignore this

Kernel



/Users/mykolagrymalyuk/Downloads/OpenCore-0-21/Docs/config.plist - Edited

Key	Type	Value
Root	Dictionary	8 key/value pairs
ACPI	Dictionary	4 key/value pairs
Booter	Dictionary	2 key/value pairs
DeviceProperties	Dictionary	2 key/value pairs
Kernel	Dictionary	7 key/value pairs
Add	Array	7 children
0	Dictionary	8 key/value pairs
Arch	String	x86_64
BundlePath	String	Lilu.kext
Comment	String	Patch engine
Enabled	Boolean	True
ExecutablePath	String	Contents/MacOS/Lilu
MaxKernel	String	
MinKernel	String	12.0.0
PlistPath	String	Contents/Info.plist
1	Dictionary	8 key/value pairs
2	Dictionary	8 key/value pairs
3	Dictionary	8 key/value pairs
4	Dictionary	8 key/value pairs
5	Dictionary	8 key/value pairs
6	Dictionary	8 key/value pairs
Block	Array	1 child
Emulate	Dictionary	5 key/value pairs
Force	Array	1 child
Patch	Array	6 children
Quirks	Dictionary	17 key/value pairs
AppleCpuPmCfgLock	Boolean	False
AppleXcpmCfgLock	Boolean	True
AppleXcpmExtraMsrs	Boolean	False
AppleXcpmForceBoost	Boolean	False
CustomSMBIOSGuid	Boolean	False
DisableIoMapper	Boolean	True
DisableLinkeditJettison	Boolean	True
DisableRtcChecksum	Boolean	False
ExtendBTFeatureFlags	Boolean	False
ExternalDiskIcons	Boolean	False
IncreasePciBarSize	Boolean	False
LapicKernelPanic	Boolean	False
LegacyCommpage	Boolean	False
PanicNoKextDump	Boolean	True
PowerTimeoutKernelPanic	Boolean	True
ThirdPartyDrives	Boolean	False
XhciPortLimit	Boolean	True
Scheme	Dictionary	3 key/value pairs
Misc	Dictionary	6 key/value pairs
NVRAM	Dictionary	6 key/value pairs
PlatformInfo	Dictionary	7 key/value pairs
UEFI	Dictionary	9 key/value pairs

<- Enable for Dell or VIAO systems

<- Enable for HP systems

Add

Why OpenCore over Clover and others

USB Creation
- Creating the USB
  - Making the installer in macOS
  - Making the installer in Windows
  - Making the installer in Linux
- Adding The Base OpenCore Files
- Gathering files
- Getting started with ACPI
- config.plist Setup

Configs
- Intel Desktop config.plist
  - Penryn
  - Clarkdale
  - Sandy Bridge
  - Ivy Bridge
  - Haswell
  - Skylake
  - Kaby Lake
  - Coffee Lake
  - Comet Lake
    - Starting Point
    - ACPI
    - Booter
    - DeviceProperties
    - Kernel
    - Misc
    - NVRAM
    - PlatformInfo
    - UEFI
    - Cleaning up
    - Intel BIOS settings
- Intel Laptop config.plist
- Intel HEDT config.plist
- AMD Desktop config.plist
- Apple Secure Boot

Installation
- Installation Process

Troubleshooting
- General Troubleshooting
  - OpenCore Boot Issues
  - Kernel Issues
  - Userspace Issues
  - Post-Install Issues
  - Miscellaneous Issues
- OpenCore Debugging
- Understanding the macOS Boot Process
- System Debugging: In-depth

Post Install
- Post-Install
- Universal
- Laptop Specifics
- Cosmetics
- Multiboot
- Miscellaneous

Extras
- Fixing KASLR slide values
- Disabling GPU
- macOS 13: Ventura
- Clover Conversion
- Choosing the right SMBIOS

Misc
- Supporting the guides
- Credits

▶ More in-depth Info

Emulate

Needed for spoofing unsupported CPUs, thankfully in 10.15.5+ Comet Lake S support was added so no need to spoof here. For those running High Sierra or Mojave, you will need the below to spoof to a supported CPU model(due to stability issues, this guide will not go over such CPUID spoofs)

Force

Used for loading kexts off system volume, only relevant for older operating systems where certain kexts are not present in the cache(ie. IONetworkingFamily in 10.6).

For us, we can ignore.

Block

Blocks certain kexts from loading. Not relevant for us.

Patch

Patches both the kernel and kexts.

Fixing I225-V controllers

This entry relates to Intel's I225-V 2.5GBe controller found on higher end Comet Lake boards, what we'll be doing here is tricking Apple's I225LM driver into supporting our I225-V network controller. However, this is only needed on Catalina and Big Sur, up to 11.3.

Key	Type	Value
Base	String	__Z18e1000_set_mac_typeP8e1000_hw
Comment	String	I225-V patch
Count	Number	1
Enabled	Boolean	True
Find	Data	`F2150000`
Identifier	String	com.apple.driver.AppleIntelI210Ethernet
MinKernel	String	19.0.0
MaxKernel	String	20.4.0
Replace	Data	`F3150000`

- **Note 1**: If your board didn't ship with the Intel I225 NIC, there's no reason to add this entry.
- **Note 2**: Leave all other keys at their default values

Quirks

Info

Settings relating to the kernel, for us we'll be enabling the following:

Quirk	Enabled	Comment
AppleXcpmCfgLock	YES	Not needed if `CFG-Lock` is disabled in the BIOS
DisableIoMapper	YES	Not needed if `VT-D` is disabled in the BIOS
LapicKernelPanic	NO	HP Machines will require this quirk
PanicNoKextDump	YES	
PowerTimeoutKernelPanic	YES	
XhciPortLimit	YES	Disable if running macOS 11.3+

▶ More in-depth Info

Scheme

Settings related to legacy booting(ie. 10.4-10.6), for majority you can skip however for those planning to boot legacy OSes you can see below:

▶ More in-depth Info

Why OpenCore over Clover and others

USB Creation

Creating the USB
Making the installer in macOS
Making the installer in Windows
Making the installer in Linux
Adding The Base OpenCore Files
Gathering files
Getting started with ACPI
config.plist Setup

Configs

Intel Desktop config.plist
Penryn
Clarkdale
Sandy Bridge
Ivy Bridge
Haswell
Skylake
Kaby Lake
Coffee Lake
Comet Lake
Starting Point
ACPI
Booter
DeviceProperties
Kernel
Misc
NVRAM
PlatformInfo
UEFI
Cleaning up
Intel BIOS settings
Intel Laptop config.plist
Intel HEDT config.plist
AMD Desktop config.plist
Apple Secure Boot

Installation

Installation Process

Troubleshooting

General Troubleshooting
OpenCore Boot Issues
Kernel Issues
Userspace Issues
Post-Install Issues
Miscellaneous Issues
OpenCore Debugging
Understanding the macOS Boot Process
System Debugging: In-depth

Post Install

Post-Install
Universal
Laptop Specifics
Cosmetics
Multiboot
Miscellaneous

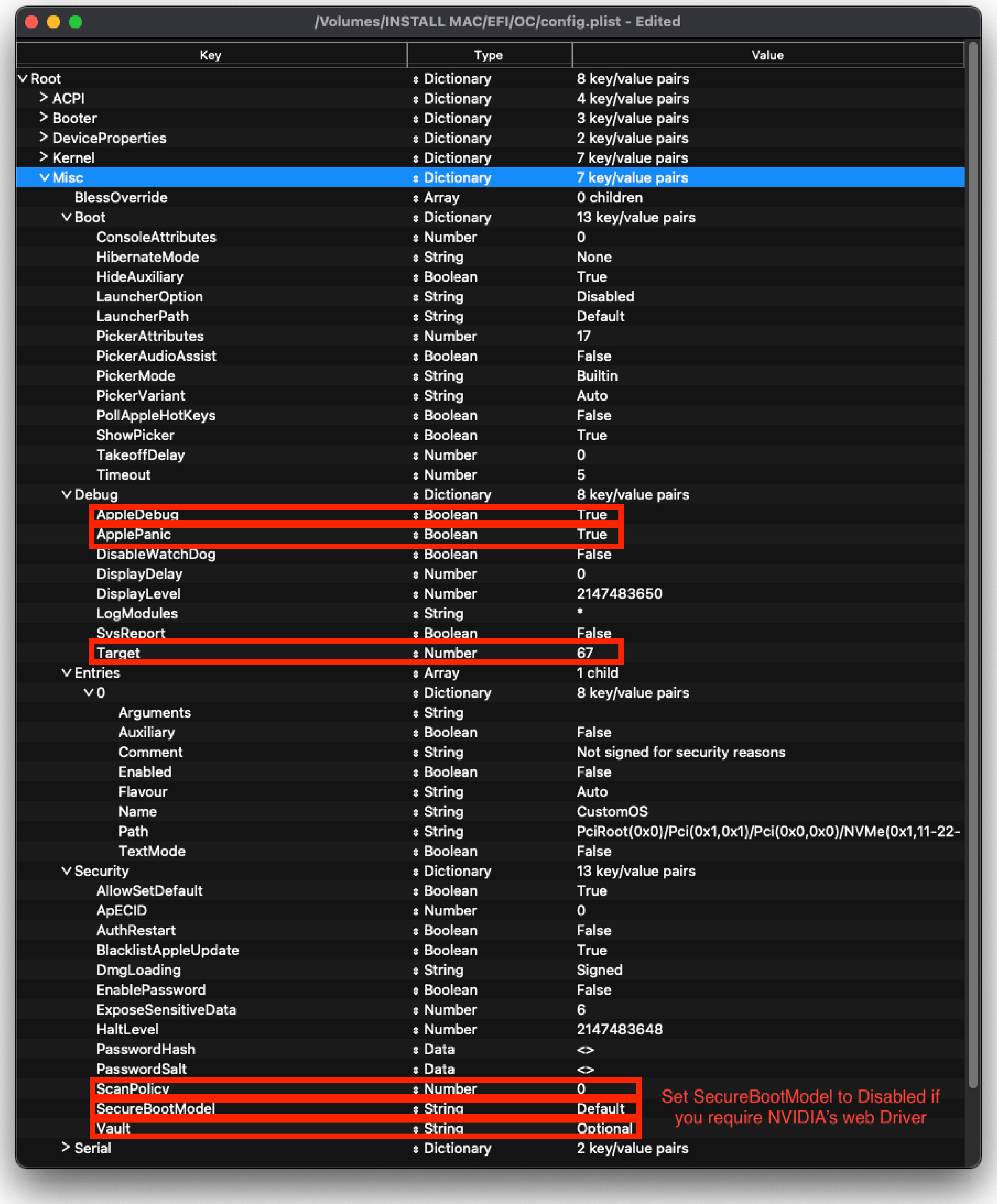
Extras

Fixing KASLR slide values
Disabling GPU
macOS 13: Ventura
Clover Conversion
Choosing the right SMBIOS

Misc

Supporting the guides
Credits

Misc



/Volumes/INSTALL MAC/EFI/OC/config.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	7 key/value pairs
BlessOverride	Array	0 children
Boot	Dictionary	13 key/value pairs
ConsoleAttributes	Number	0
HibernateMode	String	None
HideAuxiliary	Boolean	True
LauncherOption	String	Disabled
LauncherPath	String	Default
PickerAttributes	Number	17
PickerAudioAssist	Boolean	False
PickerMode	String	Builtin
PickerVariant	String	Auto
PollAppleHotKeys	Boolean	False
ShowPicker	Boolean	True
TakeoffDelay	Number	0
Timeout	Number	5
Debug	Dictionary	8 key/value pairs
AppleDebug	Boolean	True
ApplePanic	Boolean	True
DisableWatchDog	Boolean	False
DisplayDelay	Number	0
DisplayLevel	Number	2147483650
LogModules	String	*
SysReport	Boolean	False
Target	Number	67
Entries	Array	1 child
0	Dictionary	8 key/value pairs
Arguments	String	
Auxiliary	Boolean	False
Comment	String	Not signed for security reasons
Enabled	Boolean	False
Flavour	String	Auto
Name	String	CustomOS
Path	String	PciRoot(0x0)/Pci(0x1,0x1)/Pci(0x0,0x0)/NVMe(0x1,11-22-
TextMode	Boolean	False
Security	Dictionary	13 key/value pairs
AllowSetDefault	Boolean	True
ApECID	Number	0
AuthRestart	Boolean	False
BlacklistAppleUpdate	Boolean	True
DmgLoading	String	Signed
EnablePassword	Boolean	False
ExposeSensitiveData	Number	6
HaltLevel	Number	2147483648
PasswordHash	Data	<>
PasswordSalt	Data	<>
ScanPolicy	Number	0
SecureBootModel	String	Default
Vault	String	Optional
Serial	Dictionary	2 key/value pairs

Set SecureBootModel to Disabled if you require NVIDIA's web Driver

Boot

Info

Quirk	Enabled	Comment
HideAuxiliary	YES	Press space to show macOS recovery and other auxiliary entries

▶ More in-depth Info

Debug

Info

Helpful for debugging OpenCore boot issues(We'll be changing everything *but* `DisplayDelay`):

Quirk	Enabled
AppleDebug	YES
ApplePanic	YES
DisableWatchDog	YES
Target	67

▶ More in-depth Info

Security

Info

Security is pretty self-explanatory, **do not skip**. We'll be changing the following:

AllowSetDefault	YES	
BlacklistAppleUpdate	YES	
ScanPolicy	0	
SecureBootModel	Default	Leave this as `Default` for OpenCore to automatically set the correct value corresponding to your SMBIOS. The next page goes into more detail about this setting.
Vault	Optional	This is a word, it is not optional to omit this setting. You will regret it if you don't set it to Optional, note that it is case-sensitive

▶ More in-depth Info

Serial

Used for serial debugging (Leave everything as default).

Tools

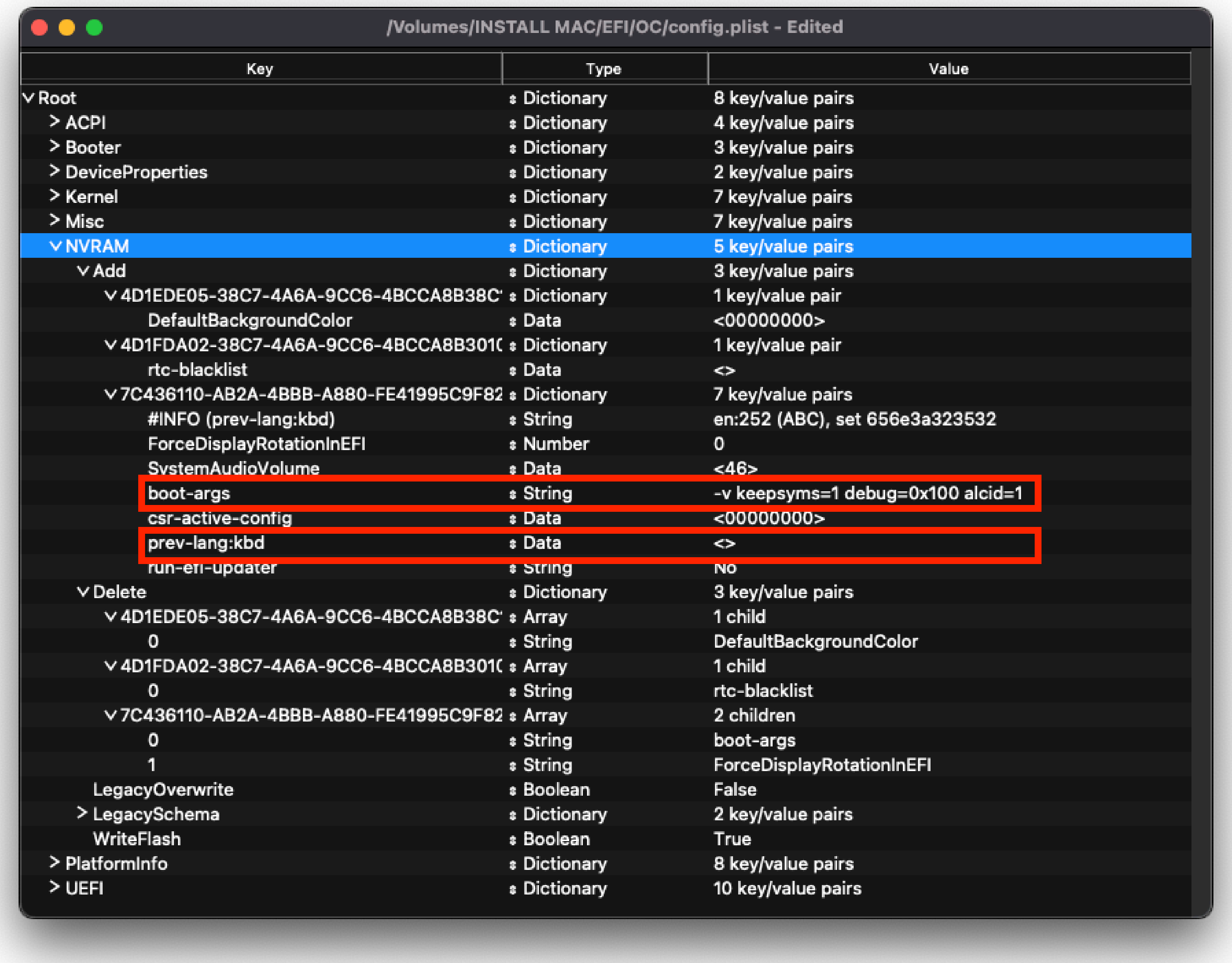
Used for running OC debugging tools like the shell, ProperTree's snapshot function will add these for you.

Entries

Used for specifying irregular boot paths that can't be found naturally with OpenCore.

Won't be covered here, see 8.6 of Configuration.pdf for more info

NVRAM



/Volumes/INSTALL MAC/EFI/OC/config.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    7 key/value pairs
  ∨ NVRAM                                    Dictionary    5 key/value pairs
    ∨ Add                                    Dictionary    3 key/value pairs
      ∨ 4D1EDE05-38C7-4A6A-9CC6-4BCCA8B38C'  Dictionary    1 key/value pair
          DefaultBackgroundColor             Data          <00000000>
      ∨ 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B301(  Dictionary    1 key/value pair
          rtc-blacklist                      Data          <>
      ∨ 7C436110-AB2A-4BBB-A880-FE41995C9F82 Dictionary    7 key/value pairs
          #INFO (prev-lang:kbd)              String        en:252 (ABC), set 656e3a323532
          ForceDisplayRotationInEFI          Number        0
          SystemAudioVolume                  Data          <46>
          boot-args                          String        -v keepsyms=1 debug=0x100 alcid=1
          csr-active-config                  Data          <00000000>
          prev-lang:kbd                      Data          <>
          run-efi-updater                    String        No
    ∨ Delete                                 Dictionary    3 key/value pairs
      ∨ 4D1EDE05-38C7-4A6A-9CC6-4BCCA8B38C'  Array         1 child
          0                                  String        DefaultBackgroundColor
      ∨ 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B301(  Array         1 child
          0                                  String        rtc-blacklist
      ∨ 7C436110-AB2A-4BBB-A880-FE41995C9F82 Array         2 children
          0                                  String        boot-args
          1                                  String        ForceDisplayRotationInEFI
      LegacyOverwrite                        Boolean       False
    > LegacySchema                           Dictionary    2 key/value pairs
      WriteFlash                             Boolean       True
  > PlatformInfo                             Dictionary    8 key/value pairs
  > UEFI                                     Dictionary    10 key/value pairs
```

Add

4D1EDE05-38C7-4A6A-9CC6-4BCCA8B38C14

Used for OpenCore's UI scaling, default will work for us. See in-depth section for more info

▶ More in-depth Info

4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102

OpenCore's NVRAM GUID, mainly relevant for RTCMemoryFixup users

▶ More in-depth Info

7C436110-AB2A-4BBB-A880-FE41995C9F82

System Integrity Protection bitmask

boot-args	Description
-v	This enables verbose mode, which shows all the behind-the-scenes text that scrolls by as you're booting instead of the Apple logo and progress bar. It's invaluable to any Hackintosher, as it gives you an inside look at the boot process, and can help you identify issues, problem kexts, etc.
debug=0x100	This disables macOS's watchdog which helps prevents a reboot on a kernel panic. That way you can *hopefully* glean some useful info and follow the breadcrumbs to get past the issues.
keepsyms=1	This is a companion setting to debug=0x100 that tells the OS to also print the symbols on a kernel panic. That can give some more helpful insight as to what's causing the panic itself.
alcid=1	Used for setting layout-id for AppleALC, see supported codecs to figure out which layout to use for your specific system. More info on this is covered in the Post-Install Page

- **Networking-Specific boot-args**:

boot-args	Description
e1000=0	Disables `com.apple.DriverKit-AppleEthernetE1000` (Apple's DEXT driver) from matching to the Intel I225-V Ethernet controller found on higher end Comet Lake boards, causing Apple's I225 kext driver to load instead. This boot argument is optional on most boards as they are compatible with the DEXT driver. However, it is required on Gigabyte and several other boards, which can only use the kext driver, as the DEXT driver causes hangs. You don't need this if your board didn't ship with the I225-V NIC. On macOS 12.2.1 and below, use `dk.e1000=0` instead

- **GPU-Specific boot-args**:

boot-args	Description
agdpmod=pikera	Used for disabling board ID checks on some Navi GPUs (RX 5000 & 6000 series), without this you'll get a black screen. **Don't use if you don't have Navi** (ie. Polaris and Vega cards shouldn't use this)
-radcodec	Used for allowing officially unsupported AMD GPUs (spoofed) to use the Hardware Video Encoder
radpg=15	Used for disabling some power-gating modes, helpful for properly initializing AMD Cape Verde based GPUs
unfairgva=1	Used for fixing hardware DRM support on supported AMD GPUs
nvda_drv_vrl=1	Used for enabling NVIDIA's Web Drivers on Maxwell and Pascal cards in macOS Sierra and High Sierra
-wegnoegpu	Used for disabling all other GPUs than the integrated Intel iGPU, useful for those wanting to run newer versions of macOS where their dGPU isn't supported

- **csr-active-config**: `00000000`
  - Settings for 'System Integrity Protection' (SIP). It is generally recommended to change this with `csrutil` via the recovery partition.
  - csr-active-config by default is set to `00000000` which enables System Integrity Protection. You can choose a number of different values but overall we recommend keeping this enabled for best security practices. More info can be found in our troubleshooting page: Disabling SIP
- **run-efi-updater**: `No`
  - This is used to prevent Apple's firmware update packages from installing and breaking boot order; this is important as these firmware updates (meant for Macs) will not work.
- **prev-lang:kbd**: <>
  - Needed for non-latin keyboards in the format of `lang-COUNTRY:keyboard`, recommended to keep blank though you can specify it(**Default in Sample config is Russian**):
  - American: `en-US:0` (`656e2d55533a30` in HEX)
  - Full list can be found in AppleKeyboardLayouts.txt
  - Hint: `prev-lang:kbd` can be changed into a String so you can input `en-US:0` directly instead of converting to HEX
  - Hint 2: `prev-lang:kbd` can be set to a blank variable (eg. `<>`) which will force the Language Picker to appear instead at first boot up.

Key	Type	Value
prev-lang:kbd	String	en-US:0

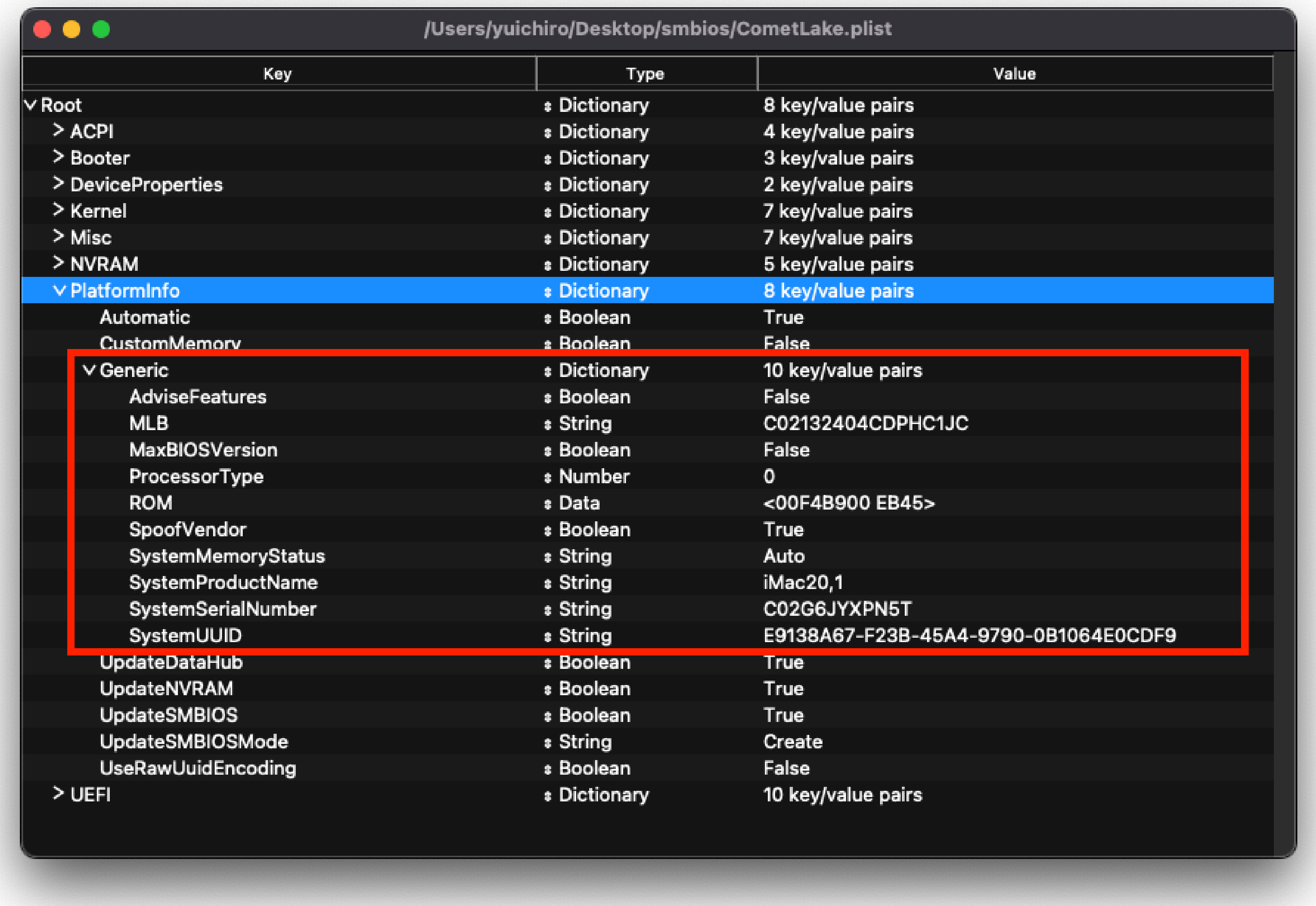
Delete

Info

Quirk	Enabled
WriteFlash	YES

▶ More in-depth Info

PlatformInfo



/Users/yuichiro/Desktop/smbios/CometLake.plist

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	7 key/value pairs
> NVRAM	Dictionary	5 key/value pairs
∨ PlatformInfo	Dictionary	8 key/value pairs
Automatic	Boolean	True
CustomMemory	Boolean	False
∨ Generic	Dictionary	10 key/value pairs
AdviseFeatures	Boolean	False
MLB	String	C02132404CDPHC1JC
MaxBIOSVersion	Boolean	False
ProcessorType	Number	0
ROM	Data	<00F4B900 EB45>
SpoofVendor	Boolean	True
SystemMemoryStatus	String	Auto
SystemProductName	String	iMac20,1
SystemSerialNumber	String	C02G6JYXPN5T
SystemUUID	String	E9138A67-F23B-45A4-9790-0B1064E0CDF9
UpdateDataHub	Boolean	True
UpdateNVRAM	Boolean	True
UpdateSMBIOS	Boolean	True
UpdateSMBIOSMode	String	Create
UseRawUuidEncoding	Boolean	False
> UEFI	Dictionary	10 key/value pairs

Info

For setting up the SMBIOS info, we'll use CorpNewt's GenSMBIOS application.

For this Comet Lake example, we'll chose the iMac20,1 SMBIOS - this is done intentionally for compatibility's sake. There are two main SMBIOS used for Comet Lake:

SMBIOS	Hardware
iMac20,1	i7-10700K and lower(ie. 8 core and lower)
iMac20,2	i9-10850K and higher(ie. 10 core)

Run GenSMBIOS, pick option 1 for downloading MacSerial and Option 3 for selecting out SMBIOS. This will give us an output similar to the following:

```sh
  #######################################################
 #               iMac20,1 SMBIOS Info                  #
#######################################################

Type:         iMac20,1
Serial:       C02XG0FDH7JY
Board Serial: C028393O3QXH69FJA
SmUUID:       DBB364D6-44B2-4A02-B922-AB4396F16DA8
```

- **Note**: MacSerial currently does not support Linux, so you must grab a Windows or macOS machine to generate the values

The `Type` part gets copied to Generic -> SystemProductName.

The `Serial` part gets copied to Generic -> SystemSerialNumber.

The `Board Serial` part gets copied to Generic -> MLB.

The `SmUUID` part gets copied to Generic -> SystemUUID.

We set Generic -> ROM to either an Apple ROM (dumped from a real Mac), your NIC MAC address, or any random MAC address (could be just 6 random bytes, for this guide we'll use `11223300 0000`. After install follow the Fixing iServices page on how to find your real MAC Address)

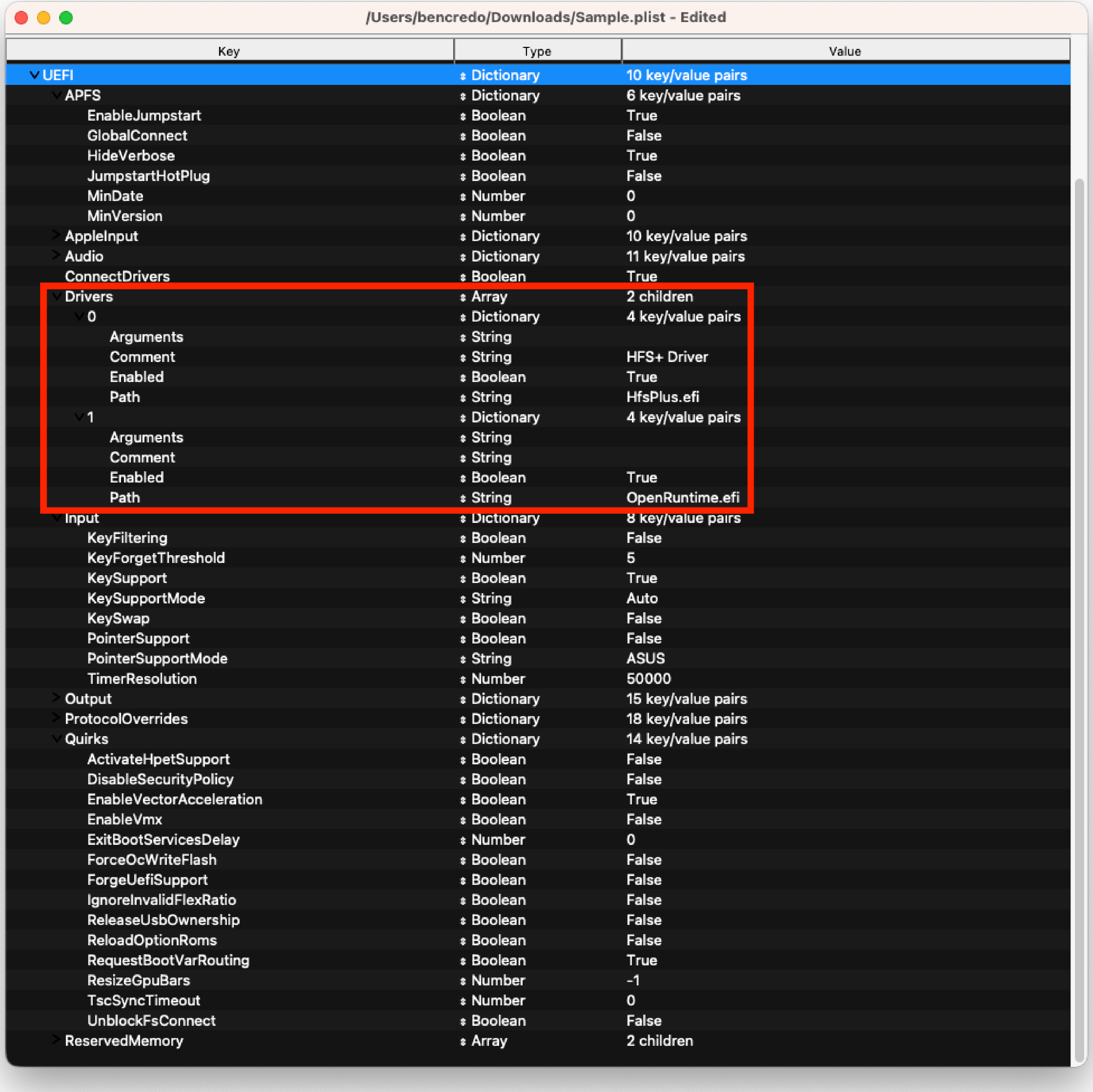
Reminder that you need an invalid serial! When inputting your serial number in Apple's Check Coverage Page, you should get a message such as "Unable to check coverage for this serial number."

Automatic: YES

- Generates PlatformInfo based on Generic section instead of DataHub, NVRAM, and SMBIOS sections

Generic

UEFI



/Users/bencredo/Downloads/Sample.plist - Edited

Key	Type	Value
UEFI	Dictionary	10 key/value pairs
APFS	Dictionary	6 key/value pairs
EnableJumpstart	Boolean	True
GlobalConnect	Boolean	False
HideVerbose	Boolean	True
JumpstartHotPlug	Boolean	False
MinDate	Number	0
MinVersion	Number	0
AppleInput	Dictionary	10 key/value pairs
Audio	Dictionary	11 key/value pairs
ConnectDrivers	Boolean	True
Drivers	Array	2 children
0	Dictionary	4 key/value pairs
Arguments	String	
Comment	String	HFS+ Driver
Enabled	Boolean	True
Path	String	HfsPlus.efi
1	Dictionary	4 key/value pairs
Arguments	String	
Comment	String	
Enabled	Boolean	True
Path	String	OpenRuntime.efi
Input	Dictionary	8 key/value pairs
KeyFiltering	Boolean	False
KeyForgetThreshold	Number	5
KeySupport	Boolean	True
KeySupportMode	String	Auto
KeySwap	Boolean	False
PointerSupport	Boolean	False
PointerSupportMode	String	ASUS
TimerResolution	Number	50000
Output	Dictionary	15 key/value pairs
ProtocolOverrides	Dictionary	18 key/value pairs
Quirks	Dictionary	14 key/value pairs
ActivateHpetSupport	Boolean	False
DisableSecurityPolicy	Boolean	False
EnableVectorAcceleration	Boolean	True
EnableVmx	Boolean	False
ExitBootServicesDelay	Number	0
ForceOcWriteFlash	Boolean	False
ForgeUefiSupport	Boolean	False
IgnoreInvalidFlexRatio	Boolean	False
ReleaseUsbOwnership	Boolean	False
ReloadOptionRoms	Boolean	False
RequestBootVarRouting	Boolean	True
ResizeGpuBars	Number	-1
TscSyncTimeout	Number	0
UnblockFsConnect	Boolean	False
ReservedMemory	Array	2 children

ConnectDrivers: YES

- Forces .efi drivers, change to NO will automatically connect added UEFI drivers. This can make booting slightly faster, but not all drivers connect themselves. E.g. certain file system drivers may not load.

Drivers

Add your .efi drivers here.

Only drivers present here should be:

- HfsPlus.efi
- OpenRuntime.efi

▶ More in-depth Info

APFS

By default, OpenCore only loads APFS drivers from macOS Big Sur and newer. If you are booting macOS Catalina or earlier, you may need to set a new minimum version/date. Not setting this can result in OpenCore not finding your macOS partition!

macOS Sierra and earlier use HFS instead of APFS. You can skip this section if booting older versions of macOS.

APFS Versions

Both MinVersion and MinDate need to be set if changing the minimum version.

macOS Version	Min Version	Min Date
High Sierra (`10.13.6`)	`748077008000000`	`20180621`
Mojave (`10.14.6`)	`945275007000000`	`20190820`
Catalina (`10.15.4`)	`1412101001000000`	`20200306`
No restriction	`-1`	`-1`

Audio

Related to AudioDxe settings, for us we'll be ignoring(leave as default). This is unrelated to audio support in macOS.

- For further use of AudioDxe and the Audio section, please see the Post Install page: Add GUI and Boot-chime

Why OpenCore over Clover and others
USB Creation
Creating the USB
Making the installer in macOS
Making the installer in Windows
Making the installer in Linux
Adding The Base OpenCore Files
Gathering files
Getting started with ACPI
config.plist Setup
Configs
Intel Desktop config.plist
Penryn
Clarkdale
Sandy Bridge
Ivy Bridge
Haswell
Skylake
Kaby Lake
Coffee Lake
Comet Lake
Starting Point
ACPI
Booter
DeviceProperties
Kernel
Misc
NVRAM
PlatformInfo
UEFI
Cleaning up
Intel BIOS settings
Intel Laptop config.plist
Intel HEDT config.plist
AMD Desktop config.plist
Apple Secure Boot
Installation
Installation Process
Troubleshooting
General Troubleshooting
OpenCore Boot Issues
Kernel Issues
Userspace Issues
Post-Install Issues
Miscellaneous Issues
OpenCore Debugging
Understanding the macOS Boot Process
System Debugging: In-depth
Post Install
Post-Install
Universal
Laptop Specifics
Cosmetics
Multiboot
Miscellaneous
Extras
Fixing KASLR slide values
Disabling GPU
macOS 13: Ventura
Clover Conversion
Choosing the right SMBIOS
Misc
Supporting the guides
Credits

Input

Related to boot.efi keyboard passthrough used for FileVault and Hotkey support, leave everything here as default as we have no use for these quirks. See here for more details: Security and FileVault

Output

Relating to OpenCore's visual output, leave everything here as default as we have no use for these quirks.

▶ More in-depth Info

ProtocolOverrides

Mainly relevant for Virtual machines, legacy macs and FileVault users. See here for more details: Security and FileVault

Quirks

Info

Relating to quirks with the UEFI environment, for us we'll be changing the following:

Quirk	Enabled	Comment
UnblockFsConnect	NO	Needed mainly by HP motherboards

▶ More in-depth Info

ReservedMemory

Used for exempting certain memory regions from OSes to use, mainly relevant for Sandy Bridge iGPUs or systems with faulty memory. Use of this quirk is not covered in this guide

Cleaning up

And now you're ready to save and place it into your EFI under EFI/OC.

For those having booting issues, please make sure to read the Troubleshooting section first and if your questions are still unanswered we have plenty of resources at your disposal:

- r/Hackintosh Subreddit
- r/Hackintosh Discord

Intel BIOS settings

- Note: Most of these options may not be present in your firmware, we recommend matching up as closely as possible but don't be too concerned if many of these options are not available in your BIOS

Disable

- Fast Boot
- Secure Boot
- Serial/COM Port
- Parallel Port
- VT-d (can be enabled if you set `DisableIoMapper` to YES)
- Compatibility Support Module (CSM) (**Must be off in most cases, GPU errors/stalls like `gIO` are common when this option is enabled**)
- Thunderbolt (For initial install, as Thunderbolt can cause issues if not setup correctly)
- Intel SGX
- Intel Platform Trust
- CFG Lock (MSR 0xE2 write protection)(**This must be off, if you can't find the option then enable `AppleXcpmCfgLock` under Kernel -> Quirks. Your hack will not boot with CFG-Lock enabled**)

Enable

- VT-x
- Above 4G Decoding
  - 2020+ BIOS Notes: When enabling Above4G, Resizable BAR Support may become an available on some Z490 and newer motherboards. Please ensure that Booter -> Quirks -> ResizeAppleGpuBars is set to `0` if this is enabled.
- Hyper-Threading
- Execute Disable Bit
- EHCI/XHCI Hand-off
- OS type: Windows 8.1/10 UEFI Mode (some motherboards may require "Other OS" instead)
- DVMT Pre-Allocated(iGPU Memory): 64MB or higher
- SATA Mode: AHCI

Once done here, we need to edit a couple extra values. Head to the Apple Secure Boot Page

Help us improve this page!

Last Updated: 7/11/2023, 12:59:44 AM

← Coffee Lake

Arrandale →