

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
  - Starting Point
  - ACPI
  - Booter
  - DeviceProperties
  - Kernel
  - Misc
  - NVRAM
  - PlatformInfo
  - UEFI
  - Cleaning up
  - Intel BIOS settings
- Haswell
- Skylake
- Kaby Lake
- Coffee Lake
- Comet Lake

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 Ivy Bridge




Support	Version
Initial macOS Support	OS X 10.7, Lion
Last Supported OS	macOS 12 Monterey
Note 1	For Ventura information, see <a href="#">macOS 13 Ventura</a>
Note 2	Ivy Bridge iGPUs are only supported up-to macOS 11

## 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!

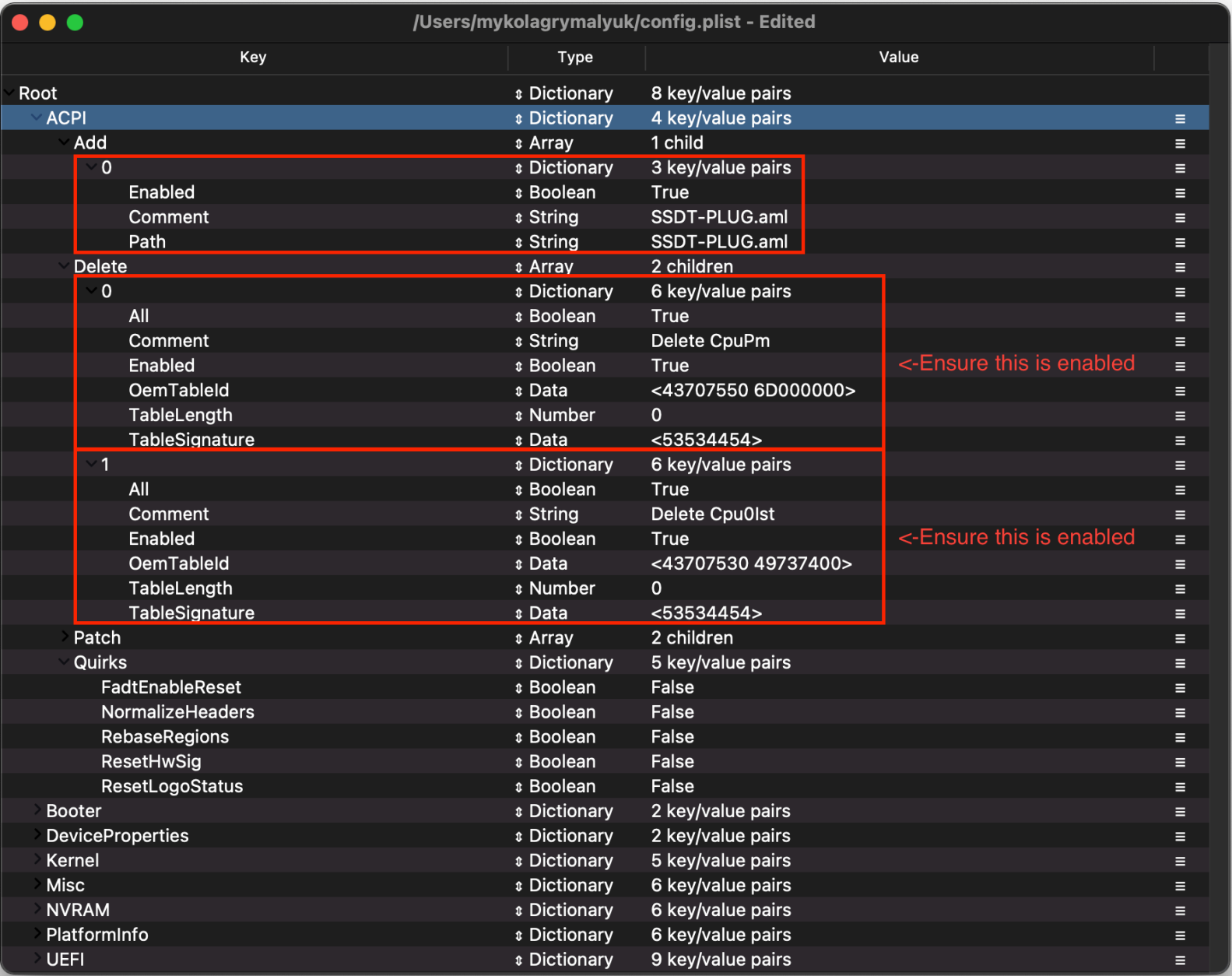
Now with all that, a quick reminder of the tools we need

- [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



## 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:

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

Starting Point

ACPI

Booter

DeviceProperties

Kernel

Misc

NVRAM

PlatformInfo

UEFI

Cleaning up

Intel BIOS settings

Haswell

Skylake

Kaby Lake

Coffee Lake

Comet Lake

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

SSDT-PM	Needed for proper CPU power management, you will need to run Pike's ssdtPRGen.sh script to generate this file. This will be run in <a href="#">post install</a> .
SSDT-EC	Fixes the embedded controller, see <a href="#">Getting Started With ACPI Guide</a> for more details.
SSDT-IMEI	Needed to add a missing IMEI device on Ivy Bridge CPU with 6 series motherboards

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

#### Info

This blocks certain ACPI tables from loading, for us we really care about this. Main reason is that Apple's XCPM does not support IvyBridge all too well and can cause AppleIntelCPUPowerManagement panics on boot. To avoid this we make our own PM SSDT in [Post-Install](#) and drop the old tables(Note that this is only temporary until we've made our SSDT-PM, we'll re-enable these tables later):

Removing CpuPm:

Key	Type	Value
All	Boolean	YES
Comment	String	Delete CpuPm
Enabled	Boolean	YES
OemTableId	Data	437075506d000000
TableLength	Number	0
TableSignature	Data	53534454

Removing Cpu0Ist:

Key	Type	Value
All	Boolean	YES
Comment	String	Delete Cpu0Ist
Enabled	Boolean	YES
OemTableId	Data	4370753049737400
TableLength	Number	0
TableSignature	Data	53534454

### 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



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

Starting Point

ACPI

Booter

DeviceProperties

Kernel

Misc

NVRAM

PlatformInfo

UEFI

Cleaning up

Intel BIOS settings

Haswell

Skylake

Kaby Lake

Coffee Lake

Comet Lake

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

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	False
DisableSingleUser	⌘ Boolean	False
DisableVariableWrite	⌘ Boolean	False
DiscardHibernateMap	⌘ Boolean	False
EnableSafeModeSlide	⌘ Boolean	True
EnableWriteUnprotector	⌘ Boolean	True
ForceBooterSignature	⌘ Boolean	False
ForceExitBootServices	⌘ Boolean	False
ProtectMemoryRegions	⌘ Boolean	False
ProtectSecureBoot	⌘ Boolean	False
ProtectUefiServices	⌘ Boolean	False
ProvideCustomSlide	⌘ Boolean	True
ProvideMaxSlide	⌘ Number	0
RebuildAppleMemoryMap	⌘ Boolean	False
ResizeAppleGpuBars	⌘ Number	-1
SetupVirtualMap	⌘ Boolean	True
SignalAppleOS	⌘ Boolean	False
SyncRuntimePermissions	⌘ Boolean	False
> 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 spaces to be passthrough to macOS that are generally ignored, useful when paired with

DevirtualiseMmio

Quirks

Info

Settings relating to boot.efi patching and firmware fixes, for us, we leave it as default

▶ More in-depth Info

DeviceProperties

config.plist — Edited		
config.plist > No Selection		
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(0x1b,0x0)	Dictionary	(1 item)
layout-id	Data	<01000000>
▼ PciRoot(0x0)/Pci(0x2,0x0)	Dictionary	(1 item)
AAPL,ig-platform-id	Data	<0A006601>
▼ PciRoot(0x0)/Pci(0x16,0x0)	Dictionary	(1 item)
device-id	Data	<3A1C0000>
> Delete	Dictionary	(0 items)
> Kernel	Dictionary	(5 items)
> Misc	Dictionary	(6 items)
> NVRAM	Dictionary	(6 items)
> PlatformInfo	Dictionary	(6 items)
> UEFI	Dictionary	(9 items)

<- device-id is only for 6 series motherboards

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.

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

The AAPL,ig-platform-id we use is as follows:

AAPL,ig-platform-id	Comment
0A006601	Used when the iGPU is used to drive a display

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

Starting Point

ACPI

Booter

DeviceProperties

Kernel

Misc

NVRAM

PlatformInfo

UEFI

Cleaning up

Intel BIOS settings

Haswell

Skylake

Kaby Lake

Coffee Lake

Comet Lake

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

<b>07006201</b>	Used when the iGPU is only used for computing tasks and doesn't drive a display
-----------------	---

Example setup:

Key	Type	Value
AAPL,ig-platform-id	Data	0A006601

(This is an example for a desktop HD 4000 without a dGPU)

### PciRoot(0x0)/Pci(0x16,0x0)


This is needed if you're pairing an Ivy Bridge CPU with a 6 series motherboard(ie. H61, B65, Q65, P67, H67, Q67, Z68), specifically needed to spoof your IMEI device into being supported. Note this property is still required with or without SSDT-IMEI.


Key	Type	Value
device-id	Data	3A1E0000

**Note:** This is not needed if you have a 7 series motherboard(ie. B75, Q75, Z75, H77, Q77, Z77)

### 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

/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
> Add	Array	2 children
> Block	Array	1 child
> Emulate	Dictionary	5 key/value pairs
> Force	Array	1 child
> Patch	Array	10 children
√ Quirks	Dictionary	22 key/value pairs
AppleCpuPmCfgLock	Boolean	True
AppleXcpmCfgLock	Boolean	True
AppleXcpmExtraMsr	Boolean	False
AppleXcpmForceBoost	Boolean	False
CustomPciSerialDevice	Boolean	False
CustomSMBIOSGuid	Boolean	False
DisableIoMapper	Boolean	True
DisableLinkeditJettison	Boolean	True
DisableRtcChecksum	Boolean	False
ExtendBTFeatureFlags	Boolean	False
ExternalDiskIcons	Boolean	False
ForceAquantiaEthernet	Boolean	False
ForceSecureBootScheme	Boolean	False
IncreasePciBarSize	Boolean	False
LapicKernelPanic	Boolean	False
LegacyCommpage	Boolean	False
PanicNoKextDump	Boolean	True
PowerTimeoutKernelPanic	Boolean	True
ProvideCurrentCpuInfo	Boolean	False
SetApfsTrimTimeout	Number	-1
ThirdPartyDrives	Boolean	False
XhciPortLimit	Boolean	False
> Scheme	Dictionary	4 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

## Add

Here's where we specify which kexts to load, in what specific order to load, and what architectures each kext is meant for. By default we recommend leaving what ProperTree has done, however for 32-bit CPUs please see below:

- More in-depth Info

## Emulate



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

Starting Point

ACPI

Booter

DeviceProperties

Kernel

Misc

NVRAM

PlatformInfo

UEFI

Cleaning up

Intel BIOS settings

Haswell

Skylake

Kaby Lake

Coffee Lake

Comet Lake

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

• **CpuidMask:** Leave this blank

- **Cpuid1Data:** Leave this blank

## 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.

## Quirks

Info		
Settings relating to the kernel, for us we'll be enabling the following:		
Quirk	Enabled	Comment
AppleCpuPmCfgLock	YES	Not needed if <code>CFG-Lock</code> is disabled in the BIOS
DisableIoMapper	YES	Not needed if <code>VT-D</code> 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

## Misc

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

Starting Point

ACPI

Booter

DeviceProperties

Kernel

Misc

NVRAM

PlatformInfo

UEFI

Cleaning up

Intel BIOS settings

Haswell

Skylake

Kaby Lake

Coffee Lake

Comet Lake

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



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:

Quirk	Enabled	Comment
AllowSetDefault	YES	
BlacklistAppleUpdate	YES	

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

Starting Point

ACPI

Booter

DeviceProperties

Kernel

Misc

NVRAM

PlatformInfo

UEFI

Cleaning up

Intel BIOS settings

Haswell

Skylake

Kaby Lake

Coffee Lake

Comet Lake

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

ScanPolicy	0	
SecureBootModel	Default	Leave this as <code>Default</code> 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 <code>Optional</code> , 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-9CC6-4BCCA8B301C	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-en-updater	String	No	
√ Delete	Dictionary	3 key/value pairs	
√ 4D1EDE05-38C7-4A6A-9CC6-4BCCA8B38C	Array	1 child	
0	String	DefaultBackgroundColor	
√ 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B301C	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

- General Purpose boot-args:



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

Starting Point

ACPI

Booter

DeviceProperties

Kernel

Misc

NVRAM

PlatformInfo

UEFI

Cleaning up

Intel BIOS settings

Haswell

Skylake

Kaby Lake

Coffee Lake

Comet Lake

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


-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 <i>hopefully</i> 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 <a href="#">supported codecs</a>  to figure out which layout to use for your specific system. More info on this is covered in the <a href="#">Post-Install Page</a> 

- 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. <b>Don't use if you don't have Navi</b> (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

Forcibly rewrites NVRAM variables, do note that `Add` **will not overwrite** values already present in NVRAM so values like `boot-args` should be left alone. For us, we'll be changing the following:

Quirk	Enabled
WriteFlash	YES

- More in-depth Info

PlatformInfo



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

Starting Point

ACPI

Booter

DeviceProperties

Kernel

Misc

NVRAM

PlatformInfo

UEFI

Cleaning up

Intel BIOS settings

Haswell

Skylake

Kaby Lake

Coffee Lake

Comet Lake

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

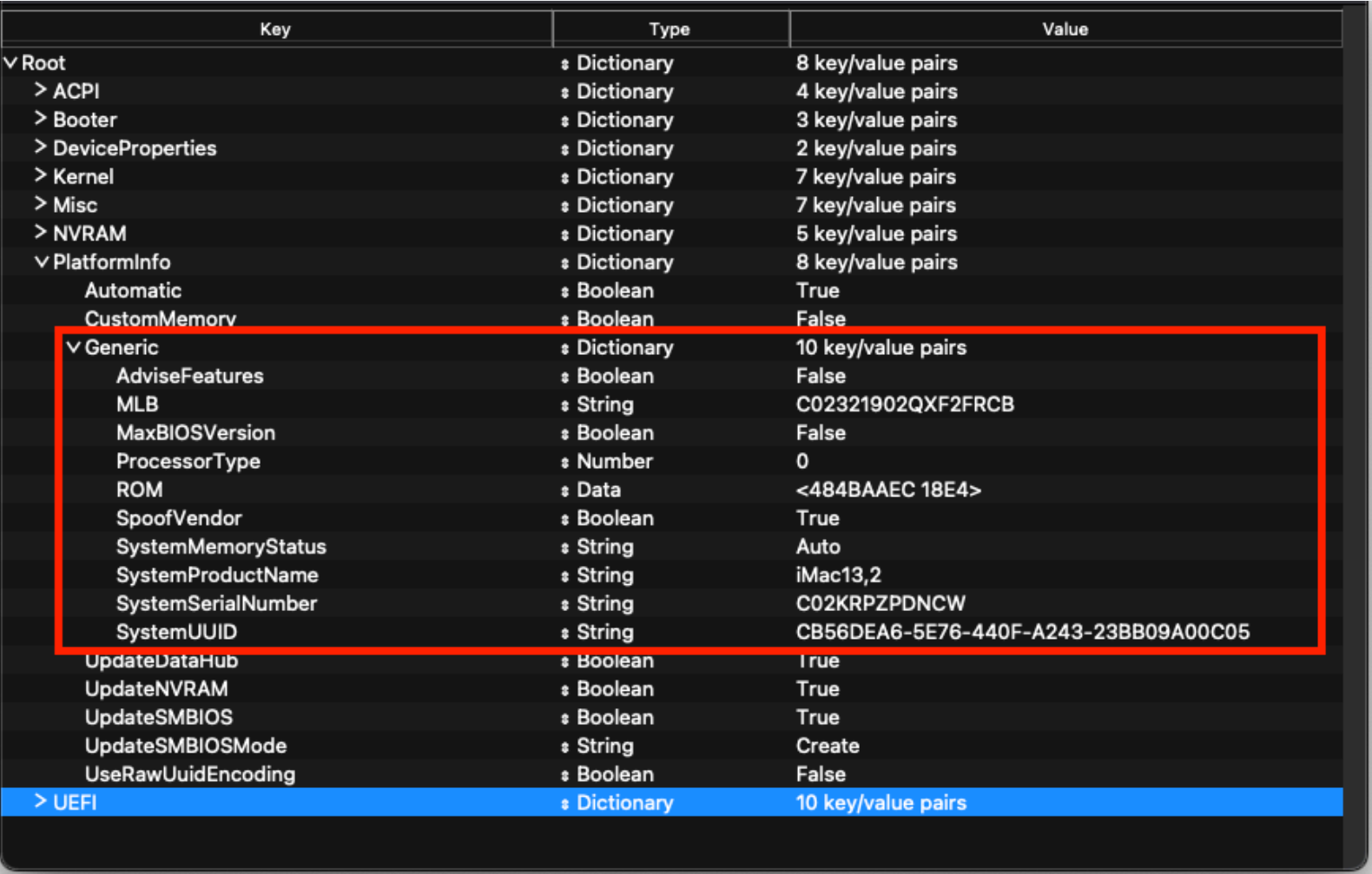
Clover Conversion

Choosing the right SMBIOS

Misc

Supporting the guides

Credits



Info

For setting up the SMBIOS info, we'll use CorpNewt's [GenSMBIOS](#) application.

For this Ivy Bridge example, we'll chose the iMac13,2 SMBIOS - this is done intentionally for compatibility's sake.

There are 3 main SMBIOS used for Ivy Bridge:

SMBIOS	Hardware
iMac13,1	Used for computers utilizing the iGPU for displaying
iMac13,2	Used for computers using a dGPU for displaying, and an iGPU for computing tasks only

**Note:** The following SMBIOS are only supported up-to and including macOS 10.15, Catalina. For cases where you must boot Big Sur or Monterey, see below:

► Big Sur/Monterey SMBIOS table

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:

```
#####
#                               iMac13,2  SMBIOS  Info                               #
#####

Type:      iMac13,2
Serial:    C02KCYZLDNCW
Board Serial: C02309301QXF2FRJC
SmUUID:    A154B586-874B-4E57-A1FF-9D6E503E4580
```

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

► More in-depth Info

UEFI

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

Starting Point

ACPI

Booter

DeviceProperties

Kernel

Misc

NVRAM

PlatformInfo

UEFI

Cleaning up

Intel BIOS settings

Haswell

Skylake

Kaby Lake

Coffee Lake

Comet Lake

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

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
√ UEFI	Dictionary	10 key/value pairs
> APFS	Dictionary	6 key/value pairs
> AppleInput	Dictionary	10 key/value pairs
> Audio	Dictionary	11 key/value pairs
ConnectDrivers	Boolean	True
√ Drivers	Array	2 children
√ 0	Dictionary	5 key/value pairs
Arguments	String	
Comment	String	
Enabled	Boolean	True
LoadEarly	Boolean	False
Path	String	OpenRuntime.efi
√ 1	Dictionary	5 key/value pairs
Arguments	String	
Comment	String	HFS+ Driver
Enabled	Boolean	True
LoadEarly	Boolean	False
Path	String	HfsPlus.efi
> Input	Dictionary	8 key/value pairs
> 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
ReleaseUsbOwnership	Boolean	False
IgnoreInvalidFlexRatio	Boolean	True
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🔗](#)

Input



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

Starting Point

ACPI

Booter

DeviceProperties

Kernel

Misc

NVRAM

PlatformInfo

UEFI

Cleaning up

Intel BIOS settings

Haswell

Skylake

Kaby Lake

Coffee Lake

Comet Lake

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

## 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
IgnoreInvalidFlexRatio	YES	
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 `gI0` 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 `AppleCpuPmCfgLock` under Kernel -> Quirks. Your hack will not boot with CFG-Lock enabled**)

### Enable

- VT-x
- Above 4G Decoding
- 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): 32MB or higher
- SATA Mode: AHCI

Once done here, we need to edit a couple extra values.  
Head to the [Apple Secure Boot Page](#)

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

        Starting Point

        ACPI

        Booter

        DeviceProperties

        Kernel

        Misc

        NVRAM

        PlatformInfo

        UEFI

        Cleaning up

        Intel BIOS settings

    Haswell

    Skylake

    Kaby Lake

    Coffee Lake

    Comet Lake

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

← Sandy Bridge

Haswell →