

# 精解 OpenCore

此 PDF 副本仅限 OpenCore 技术交流群内使用, 未经允许严禁转载/发.

© 2019 Bat.bat & Daliansky 许可协议 [CC-BY-NC-SA](#)



## 什么是 OpenCore

OpenCore 是一个着眼于未来开源引导工具, 最初诞生于 HermitCrabs 实验室, 现在接手于 Acidanthera, 其目的是创建一个更加严谨的模组化的轻量引导系统。尽管 OpenCore 的主要用途是黑苹果, 它也支持其它操作系统的引导。

OpenCore 现在已经处于公测 Beta 阶段, 引导相关功能也已非常稳定。但是除非你想站在 Hackintosh 开发的最前沿或计划为该项目做贡献, 在当前已经拥有稳定系统的情况下, 不建议将黑苹果作为生产力的朋友折腾。

这个教程只能作为你探索的起点。请仔细阅读并时刻牢记你的硬件可能有不同的配置要求。

## 为什么选择 OpenCore

1. OpenCore 启动黑苹果 (尤其是 macOS 10.15) 会比 Clover 快不少
2. OpenCore 提供强大的排错功能和详细的日志
3. OpenCore 更加注重系统的安全性同时对文件保险箱 (FileVault) 有更强大的支持
4. OpenCore 支持基于 `boot.efi` 的原生开机快捷键支持
  - 按住 `Option` 或 `ESC` 来选择启动项
  - 按住 `Command + R` 来进入恢复模式
  - 按住 `Command + Option + R + R` 来重置 NVRAM
5. OpenCore 使用更加先进的方法注入内核扩展 (Kext) 且与此同时不会破坏系统完整性保护。
6. OpenCore 通过读取启动磁盘设置的 NVRAM 变量, 可以像白苹果一样支持在设置的启动磁盘切换默认引导项
7. 支持给其它 .efi 驱动或引导工具加入参数
8. 如下 .efi 驱动被合并入 OpenCore, 未来的开发直接与 OpenCore 绑定, 且不再支持 Clover
  - `AptioMemoryFix` 已经被合并入 OpenCore 并将未合并的模组更名为 `FwRuntimeVariable.efi`, 提供更多可按需求定制 Slide 值计算的 Quirk
  - `AppleGenericInput` (原来的 `AptioInputFix`) 为了提高兼容性和 `ALT` / `Command` 键的对调即将合并入 OpenCore 并增加更多的 Quirk
  - `AppleUISupport` 为了提供更好的 FileVault 体验, 已合并入 OpenCore 并增加了更多的 Quirk
  - `ApfsDriverLoader` 计划在未来合并苹果最新的 APFS 驱动源码并支持 Fusion Drive, 有可能并入 OpenCore
9. 从 2019 年 9 月以后, Acidanthera 开发的内核驱动 (Lilu, AppleALC 等等) 不再会在 Clover 上做兼容性测试

## • OpenCore 常见疑问

1. OpenCore 没有精美的主题, 以后会添加吗?
  - 可能但是短时间内不会有, OpenCore 的开发成员 vit9696 已经声明不会开发官方主题, 但是保留了 GUI 接口, 其他开发者可以自己制作

- OpenCore 自带的 GUI 仅用于 Debug, 正常使用的时候是像白苹果一样隐藏的。OpenCore 已经提供了大量可以绕开 GUI 的功能, 将 `ShowPicker` 设置为 `NO` 隐藏菜单之后, 可以通过系统自带的启动磁盘设置来切换默认系统, 用苹果原生快捷键来清空 NVRAM 和添加引导表示符, 需要其它功能的时候可以随时按 ESC/Option 调出, 主题存在的意义不大
2. OpenCore 为什么不会为其它操作系统忽略 ACPI 补丁?
- 真正正确的 ACPI (DSDT/SSDT) 应当适用于所有的操作系统, 单独为一个操作系统制作 ACPI 是不合理的, 因此 OpenCore 基于 ACPI 标准, 不会在 ACPI 上区别对待操作系统, (OpenCore ACPI 教程正在施工中, 敬请期待)
3. OpenCore 的配置为什么看起来这复杂?
- OpenCore 为了提高兼容性, 为用户开放了更多底层的 Quirk
  - OpenCore 现阶段没有可用的非常直观的 GUI 编辑器
  - OpenCore 由于经常更新 Config, 现阶段还没有人为其制作通用 Config
- 换位思考, 如果用 Xcode 来编辑 Clover 安装包内自带的, OpenCore 的配置就会显得简单明了很多, OpenCore 很多功能
- 都有且只有一处设置, 不像 Clover 有大量都干扰组合

## 配置 OpenCore

讲了这么多, 终于到真正动手的环节了

### 准备工作

我们先来做些简单的准备工作

#### 推荐 BIOS 设置

禁用:

英文	中文
Fast Boot	快速启动
CFG Lock (MSR 0xE2 write protection)	CFG 锁 (MSR 0xE2 写入保护)
VT-d	<a href="#">VT-d</a>
CSM	兼容性支持模块

启用:

英文	中文
VT-x	<a href="#">VT-x</a>
Above 4G decoding	大于 4G 地址空间解码
Hyper Threading	处理器超线程
Execute Disable Bit	执行禁止位
EHCI/XHCI Hand-off	接手 EHCI/XHCI 控制
OS type: Windows 8.1/10	操作系统类型: Windows 8.1/10

#### 下载如下文件

- [OpenCorePkg](#) (建议从 Acidanthera 的官方例子开始构建, 并使用官方 Release, 因为 OpenCore 的配置文件需要不断随着主程序的更新而更新)
- [MacInfoPkg](#)
- [AppleSupportPkg](#)
- Plist 编辑器:
  - [Xcode](#) (Acidanthera 官方推荐)
  - [ProperTree](#) (基于 Python 的轻量级跨平台 Plist 编辑器, 针对 OpenCore 有优化, Acidanthera 官方推荐)
  - [Plist Editor Pro](#) (使用远古 Plist 标准, 会破坏 XML 结构而且不会自动转换 Base64, 可用但是不推荐)
- [OpenCore Configurator \[1\]](#), [OpenCore Configurator \[2\]](#) 是两个第三方配置工具, 但是 vit9696 [多次声明](#) 他不建议使用
- 这些工具因为它们在编辑时会破坏 plist 的文件格式并且会造成一系列未知问题。目前 (2019.10.1) 二者的更新已经远远落后于 OpenCore, 使用会导致整个 Config 直接损坏!

## - 需要具备的条件

- 一个 U 盘
- 对黑苹果有一定的知识基础, 至少会自己配置 Clover, 清楚各个 内核驱动 `.kext` 和 UEFI `.efi` 驱动的用途, 不清楚可以查看[附录\[1\]](#)和 [附录\[2\]](#)
- 一个完全精简过的 Clover EFI
- 一个正常稳定工作的黑苹果测试平台
- EFI 挂载工具
  - [mountEFI](#)
  - [Hackintool](#)

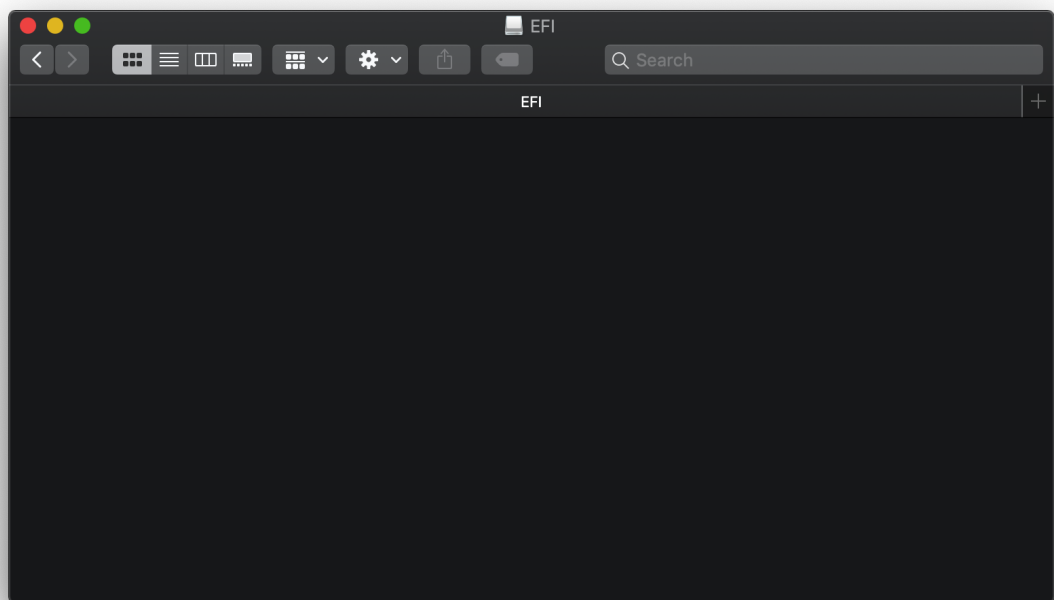
## • 创建 USB 启动盘

1. 为了不干扰当前引导工具, 建议单独创建一个 USB 启动盘, 格式化为 GUID 分区图, 分区类型为 macOS 日志式。



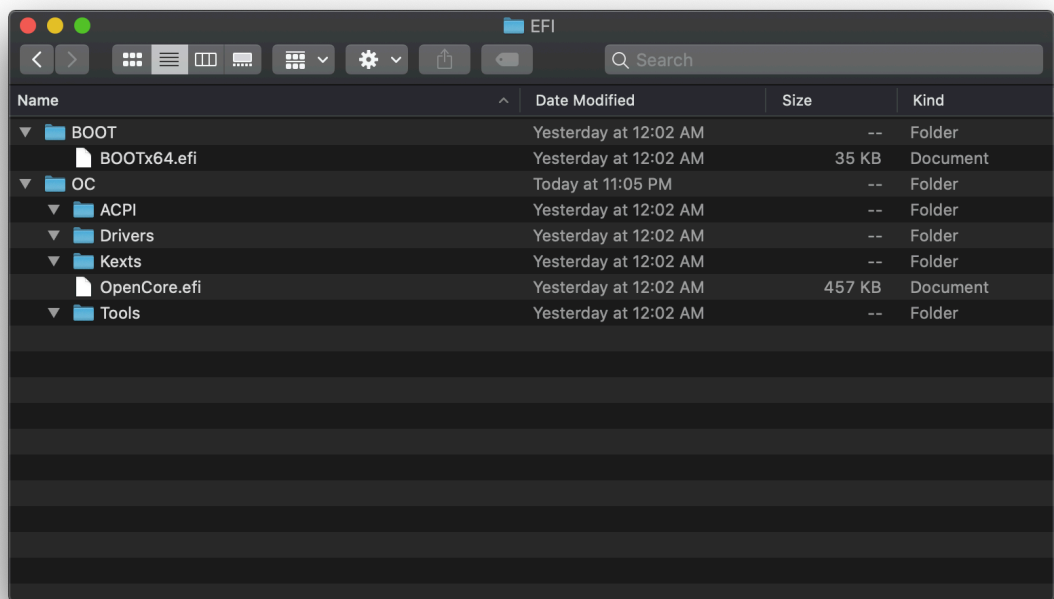
## 2. 挂载 EFI 分区

你会注意到, U 盘的 EFI 分区是空的, 于是正真有意思的部分就开始了。

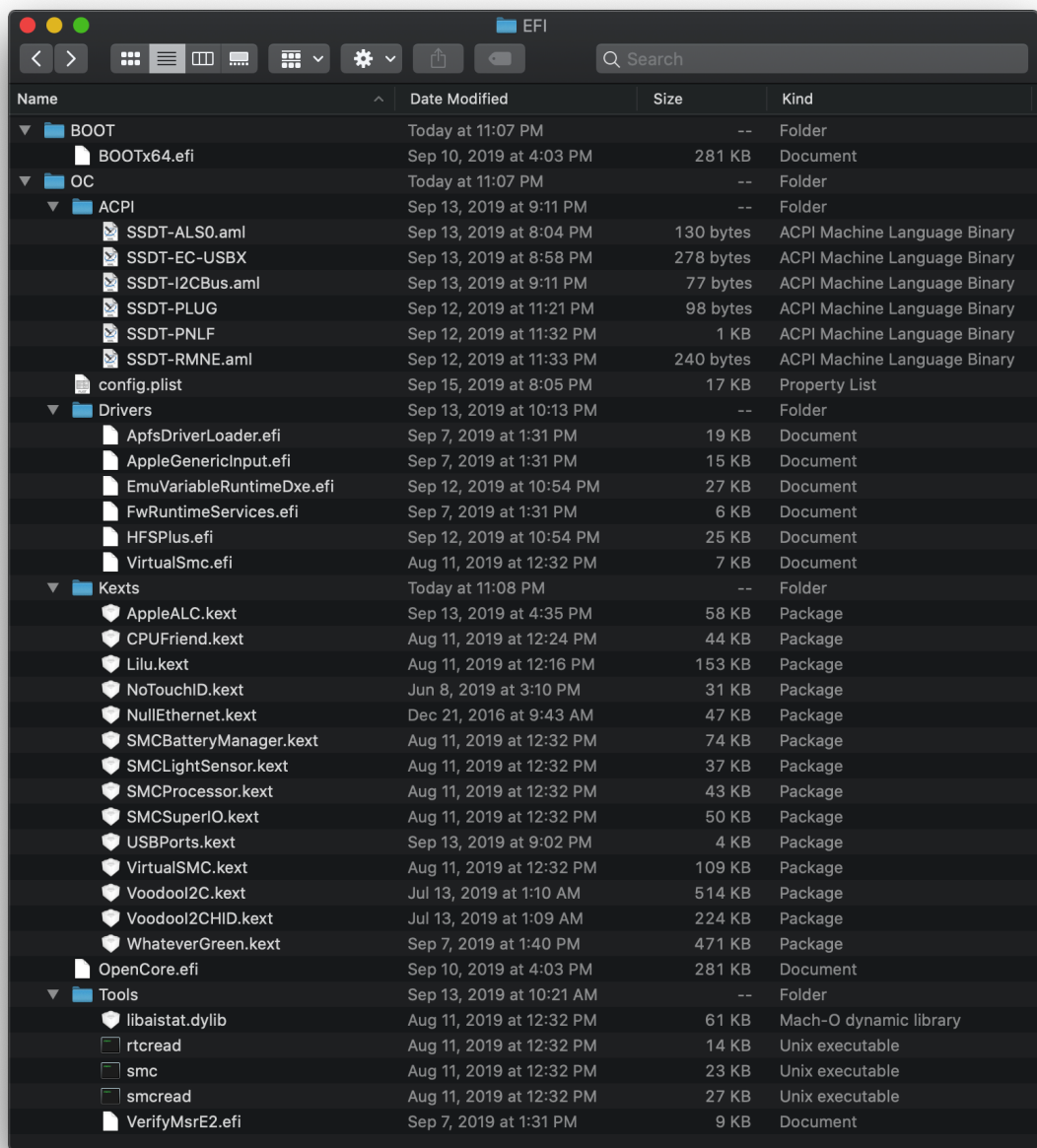


- 基础文件夹结构

1. OpenCore 的正确文件结构如下所示



2. 现在, 你可以将 AppleSupportPkg 和 VirtualSMC 中必需的 `.efi` 驱动程序放入 `drivers` 文件夹, 将 你的 `Kext` 和 `DSDT / SSDT` 放入各自的文件夹中。请注意, OpenCore 不支持支持列表以外的 UEFI 驱动程序!
3. 完成后的效果 (请忽略这些随便放的驱动):

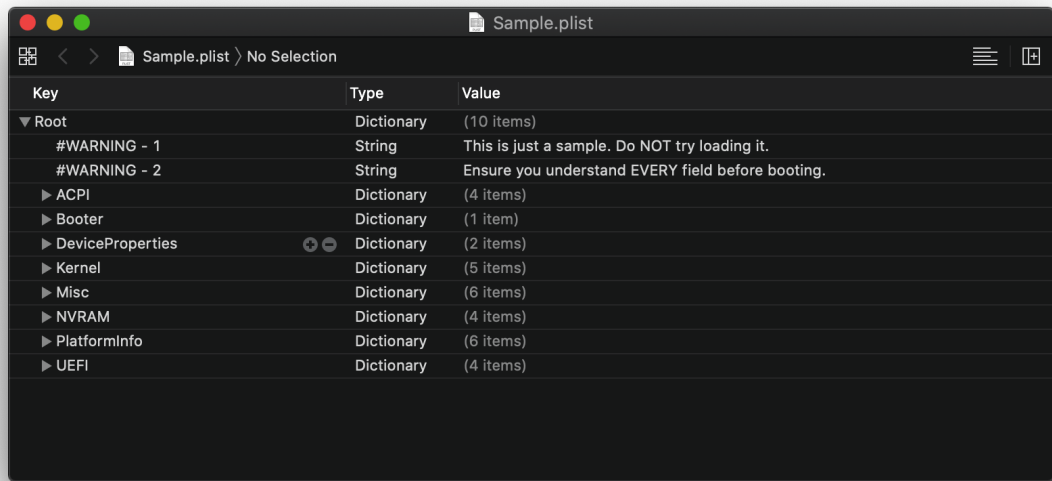


## • 配置 Config.plist

- 请记住 OpenCore 中的 Config.plist, 与 Clover 的 Config.plist 尽管名称相同但是内容和结构完全不同。它们不能混合滥用。
- 如果你不清楚某一个项里应该填什么类型的值, 请参考 [OpenCore 官方文档](#)
- 下面的 [Quirk](#) 给的都是推荐值, 请根据自身情况修改

■ 由于精力有限, 部分 Quirk 和 参数未在文中提及

1. 复制 `Sample.plist`, 将副本重命名为 `Config.plist` 并在 Xcode 中打开。



2. 里面躺着所有我们需要配置的子项:

- **ACPI** : 用于加载, 屏蔽和修补 ACPI (DSDT, SSDT) 表
- **Booter** : 用于设置 FwRuntimeServices.efi (Slide 值计算, KASLR)
- **DeviceProperties** : 用于设置 PCI 设备属性, 如英特尔缓冲帧补丁, 声卡 Layout ID
- **Kernel** : 用于说明 OpenCore 的具体加密信息, 配置 Kext 加载顺序以及屏蔽驱动
- **Misc** : OpenCore 的自身设置
- **NVRAM** : 用于注入 NVRAM (如引导标识符和 SIP)
- **Platforminfo** : 用于设置 SMBIOS 机型信息
- **UEFI** : 用于加载 UEFI 驱动以及以何种顺序加载

我们可以删除 #WARNING - 1 和 #WARNING - 2 来让整个配置文件看上去更整洁一些。

## • ACPI

Add: 你需要把这些而例子里内容修改/创建为你的 **EFI/OC/ACPI/** 下的文件

Block: 我们不会在这里做任何修改

Patch: 对 DSDT (SSDT) 的内容进行查找和替换

Quirks: ACPI 相关设置

- **FadtEnableReset**: **NO**
  - 在旧硬件上修复重启和关机, 除非需要, 否则不推荐开启
- **NormalizeHeaders**: **NO**
  - 清除 ACPI 头字段, 只有 macOS 10.13 需要
- **RebaseRegions**: **YES**
  - 尝试试探性地重新定位 ACPI 内存区域, 使用自定义 DSDT 则必须开启
- **ResetHwSig**: **NO**
  - 存在重新启动后因无法维持硬件签名而导致从休眠中唤醒的问题的硬件需要开启
- **ResetLogoStatus**: **NO**
  - 无法在有 **BGRT** 表的系统上显示 OEM Windows 标志的硬件需要开启

## • Booter 启动器

Quirks:

- **AvoidRuntimeDefrag**: **YES**

- 开启后会修复 UEFI 的运行服务, 例如日期, 时间, NVRAM, 电源控制等
- **DevirtualiseMmio:** NO
  - 开启后会减少 Stolen 内存占用空间, 扩大 `Slide = N` 值的范围, 但可能与主板不兼容。通常用于 APTIO V 固件 (Broadwell +)
- **DisableSingleUser:** NO
  - 开启后会禁止 `Cmd + S` 和 `-s` 的使用, 使设备更加接近于 T2 白苹果
- **DisableVariableWrite:** NO
  - 开启后会禁止 NVRAM 写入, 在 Z390/HM370 等没有原生 macOS 支持 NVRAM 的主板上需要开启
- **DiscardHibernateMap:** NO
  - 开启后会重用原始休眠内存映射, 仅某些旧硬件需要
- **EnableSafeModeSlide:** YES
  - 开启后会允许在安全模式下使用 Slide 值
- **EnableWriteUnprotector:** YES
  - 开启后会在执行期间删除 CR0 寄存器中的写入保护
- **ForceExitBootServices:** NO
  - 开启后会确保 ExitBootServices 即使在 MemoryMap 发生更改时也能调用成功, 除非有必要, 否则请勿使用
- **ProtectCsmRegion:** NO
  - 开启后会用于修复人为制造和睡眠唤醒的问题, AvoidRuntimeDefrag 已经修复了这个问题所以请尽可能避免使用这个 Quirk
- **ProvideCustomSlide:** YES
  - 如果 Slide 值存在冲突, 此选项将强制 macOS 执行以下操作:
    - 使用一个伪随机值。只有在遇到 `Only N/256 slide values are usable!` 时需要
- **SetupVirtualMap:** YES
  - 开启后会将 SetVirtualAddresses 调用修复为虚拟地址
- **ShrinkMemoryMap:** NO
  - 有巨大且不兼容内存映射的主板需要开启, 非必须不要使用

## • DeviceProperties 设备属性

– 不同的设备硬件地址不一样! 你需要先通过 Hackintool 或者 Windows 设备管理器 等工具查看 PCI 设备地址

■ 这里的内容可以直接从 Clover 的 Config.plist 复制过来, 请提前确保内容无误

Add: 设置设备属性

`PciRoot(0x0)/ Pci(0x2,0x0)` → `AAPL, ig-platform-id`

- 应用缓冲帧补丁, 查看[此处](#)来确认你需要的缓冲帧补丁

`PciRoot(0x0)/ Pci(0x1f, 0x3)` → `Layout-id`

- 应用 AppleALC 音频 Layout ID 注入

屏蔽: 用于删除设备属性 (可以删除此项, 与大多数用户都用不到)

## • Kernel 内核

– Add

这里是你指定要加载哪些 Kext 以及仿冒 CPU ID 的地方, 这里的顺序非常重要, 所以请确保 Lilu.Kext 始终在第一位! 其他优先级更高的 Kext 为 Lilu 的插件, 如 VirtualSMC, AppleALC, WhateverGreen 等。(有些驱动里面还包含插件驱动, 如: VoodooI2C, VoodooPS2 注意要把里面的插件也全部列出)



- **BundlePath**
  - 这里填入 Kext 的名称
  - 如: `Lilu.kext`
- **Enabled**
  - 控制 Kext 的启用禁用
- **ExecutablePath**
  - 隐藏在 kext 中的实际可执行文件的路径, 可以通过右键单击并选择 `显示包内容` 来查看 Kext 的路径。通常为 `Contents/MacOS/(Kext)`, 但有的时候 `Plugin` 文件夹下也会有 Kext。注意空壳 Kext 不需要填写此处。
  - 如: `Contents/MacOS/Lilu`
- **PlistPath**
  - 隐藏在 Kext 中的 `info.plist` 路径
  - 如: `Contents/Info.plist`

**Emulate:** 仿冒不支持的 CPU, 例如奔腾和赛扬

- **CpuidMask:** 设置为零时, 将使用原始 CPU 位
  - `<Clover_FCPUID_Extended_to_4_bytes_Swapped_Bytes> | 00 00 00 00 | 00 00 00 00 | 00 00 00 00`
  - 例如: CPUID `0x0306A9` 就是 `A9 06 03 00 | 00 00 00 00 | 00 00 00 00 | 00 00 00 00`
- **CpuidData:** CPU 的被仿冒位
  - `FF FF FF FF | 00 00 00 00 | 00 00 00 00 | 00 00 00 00`
  - 如果需要替换更长的位, 将 `00` 替换为 `FF`

**Block:** 屏蔽系统里的 Kext

**Patch:** 这是你要添加系统内核补丁, Kext 补丁, 和 AMD CPU 补丁的地方。(等同于 Clover 的 `KextToPatch` 和 `KernelToPatch`)

**Quirks:**

- **AppleCpuPmCfgLock:** `N0`
  - 仅在 BIOS 中无法禁用 CFG-Lock 时才需要
- **AppleXcpmCfgLock:** `N0`
  - 仅在 BIOS 中无法禁用 CFG-Lock 时才需要
- **AppleXcpmExtraMsrs:** `N0`
  - 禁用奔腾和某些至强等不支持 CPU 所需的多个 MSR 访问
- **CustomSMBIOSGuid:** `N0`
  - 对 UpdateSMBIOSMode 自定义模式执行 GUID 修补, 用于戴尔笔记本电脑 (等同于 Clover 的 `DellSMBIOSPatch`)
- **DisbaleIOMapper:** `N0`
  - 需要绕过 VT-d 且 BIOS 中禁用时使用
- **ExternalDiskIcons:** `YES`
  - 硬盘图标补丁, macOS 将内部硬盘视为外接硬盘 (黄色) 时使用
- **LapicKernelPanic:** `N0`
  - 禁用由 AP 核心 lapic 中断造成的内核崩溃, 通常用于惠普电脑 (等同于 Clover 的 `Kernel LAPIC`)
- **PanicNoKextDump:** `YES`
  - 在发生内核崩溃时阻止输出 Kext 列表, 提供可供排错参考的日志
- **ThirdPartyTrim:** `N0`
  - 为 SSD 启用 TRIM 指令, NVMe SSD 会自动被 macOS 加载因此不需要
- **XhciPortLimit:** `YES`
  - 这实际上是 15 端口限制补丁, 不建议依赖, 因为这不是 USB 的最佳解决方案。有能力的情况下请选择定制 [USB](#), 这个选项用于没有定制 USB 的设备



## • Misc 杂项

Boot: 引导界面的设置 (保持原样, 除非你知道你在做什么)

- HibernateMode: `None`
  - 最好避免与黑苹果一同休眠
- HideSelf: `YES`
  - 在 OpenCore 的启动选择中隐藏自身 EFI 分区的启动项
- PollAppleHotKeys: `YES`
  - 允许在引导过程中使用苹果原生快捷键, 需要与 `AppleGenericInput.efi` 或 `UsbKbDxe.efi` 结合使用, 具体体验取决于固件。常用组合:
    - `Cmd + V` : 启用 `-v` 跑码
    - `Cmd + Opt + P + R` : 重置 NVRAM
    - `Cmd + R` : 启动恢复分区
    - `Cmd + S` : 启动至单用户模式
    - `Option / ALT` : 在 `ShowPicker` 设置成 `NO` 时显示引导项, ALT 不可用时可用 `ESC` 键代替
- Timeout: `5`
  - 设置引导项等待时间
- ShowPicker: `YES`
  - 显示 OpenCore 的 UI, 用于查看可用引导项, 设置为 `NO` 可以和 `PollAppleHotKeys` 配合提升体验
- UsePicker: `YES`
  - 使用 OpenCore 的默认 GUI, 如果您希望使用其他 GUI (暂时没有), 则设置为 `NO`

Debug: Debug 有特殊用途, 除非你知道你在做什么, 否则保持原样。

- DisableWatchDog: `NO`
  - 如果 macOS 在启动时卡在某些地方, 可能需要设置为 `YES`, 通常用于避免无用的排错信息

Security: 安全

- AllowNvramReset: `YES`
  - 允许在引导选择界面和快捷键 `Cmd + Opt + P + R` 按下时重置 NVRAM
- RequireSignature: `NO`
  - 我们不会使用 `Vault.plist`, 所以我们可以忽略
- RequireVault: `NO`
  - 我们不会使用 `Vault.plist`, 所以我们可以忽略
- ScanPolicy: `0`
  - `0` (允许显示所有可用的硬盘)

Tools 用于运行 OC 调试工具, 例如验证 CFG 锁 (VerifyMsrE2)

- Name
  - OpenCore 启动项中显示的名称
- Enabled
  - 启用或禁用
- Path
  - `Tools` 文件夹下的文件路径
  - 如: [Shell.efi](#)

Entires: 用于指定 OpenCore 无法自动找到的无规律引导路径

- Name
  - OpenCore 启动项中显示的名称
- Enabled
  - 启用或禁用

- Path

- 引导磁盘的 PCI 路径, 可以通过 `OpenCoreShell` 的 `map` 命令找到
- 如: `PciRoot(0x0)/Pci(0x1D,0x4)/Pci(0x0,0x0)/NVMe(0x1,09-63-E3-44-8B-44-1B-00)/HD(1,GPT,11F42760-7AB1-4DB5-924B-D12C52895FA9,0x28,0x64000)/\EFI\Microsoft\Boot\bootmgfw.efi`

- NVRAM

Add:

- 4D1EDE05-38C7-4A6A-9CC6-4BCCA8B38C14
- UIScale: OpenCore UI 和引导第一阶段缩放
  - 01: 正常大小
  - 02: HIDPI (使 FileVault 在较小的显示器上正常显示)
- 7C436110-AB2A-4BBB-A880-FE41995C9F82 (系统完整性保护位掩码)
  - boot-args: `-v dart=0 debug=0x100 keepsyms=1` 等 (引导标识符, 更多请查看 [附录\[3\]](#))
  - csr-active-config: `<00000000>` (SIP 设置)
    - 00000000 - SIP 完全开启
    - 30000000 - 允许未签名的 Kext 加载并允许写入受保护的文件系统路径
    - E7030000 - SIP 完全关闭
    - 67000000 —不再推荐使用
  - nvda\_drv: `<>` (设为 31 时启用 NVIDIA WebDrivers)
  - prev-lang:kbd: `<0x7a682d48616e733a323532>` (将默认语言设置为简体中文, 留空为英文)

Block: 强制重写 NVRAM 变量, 一般不需要, 因为 `sudo nvram` 是首选, 但对那些极端情况很有用。请注意, Add 不会覆盖 NVRAM 中已经存在的值

LegacyEnable: NO

- 允许将 NVRAM 存储在 nvram.plist 中, 没有原生 NVRAM 的设备必须开启

LegacySchema:

- 用于分配 NVRAM 变量, 与 LegacyEnable 配合使用

- PlatformInfo

Auto: YES (基于 Generic 部分生成 PlatformInfo, 而不是 DataHub, NVRAM 和 SMBIOS 部分)

OpenCore 提供了大量方法来注入 PlatformInfo, 这里只介绍最简单的方法

Generic: (通用)

- SpoofVendor: YES (仿冒制造商为 Acidanthera 来避免出现冲突)
- SystemUUID: 建议留空并让 OpenCore 自动注入设备自带 UUID 或填入设备自带 UUID 以防出现 Windows 激活问题
- MLB: 用 macserial 读取或生成
- ROM: `<>` 可以是任意 6 Byte MAC 地址, 如 `0x112233000000`
- SystemProductName: 用 macserial 读取或生成
- SystemSerialNumber: 用 macserial 读取或生成

UpdateDataHub: YES (更新 DataHub)

UpdateNVRAM: YES (更新 NVRAM)

UpdateSMBIOS: YES (更新 SMBIOS)

UpdateSMBIOSMode: Create (用新分配的 EfiReservedMemoryType 替换原有的表, 戴尔笔记本需要使用 Custom 并开启 CustomSMBIOSGuid)

## - macserial

使用 macserial 查看原有 PlatformInfo:

1. 打开下载好的 MacInfoPkg
2. 找到里面的 `macserial` 并将其复制到桌面
3. 终端输入:

```
cd ~/Desktop
./macserial
```

4. 此时你现有的 PlatformInfo 就已经列出来了, 对着 Config 对号入座即可

使用 macserial 生成新的 PlatformInfo:

1. 如果要生成新的 PlatformInfo 请输入如下命令来列出所有支持的设备型号:

```
./macserial -l | grep -i "Model:"
```

2. 然后输入如下命令来生成序列号和 MLB

```
./macserial -m <设备型号>
```

请忽略 `< >`, 例子: `./macserial -m MacBookPro15,3`

其中 `|` 前面的是序列号 (SystemSerialNumber), 后面的是 MLB

## • UEFI

ConnectDrivers: `YES`

- 强制加载 `.efi` 驱动程序, 更改为 `NO` 将自动连接添加的 UEFI 驱动程序, 这样以获得更快的启动速度, 但并非所有驱动程序都可以自行连接, 某些文件系统驱动程序可能无法加载

Drivers: 在这里添加你的 `.efi` 驱动

Protocols: (协议)

- AppleBootPolicy: `NO`
  - 用于确保虚拟机或旧版 Mac 上的 APFS 兼容性
- ConsoleControl: `YES`
  - macOS 引导加载程序需要用于文本输出的控制台控制协议, 某些固件缺少该协议。当协议已经在固件中可用时, 需要设置此选项, 并且使用其他控制台控制选项, 例如 `IgnoreTextInGraphics`, `SanitiseClearScreen` 以及 `ConsoleBehaviourUi` 的 `ConsoleBehaviourOs`
- DataHub: `NO`
  - 重新安装数据库
- DeviceProperties: `NO`
  - 确保在 VM 或旧白苹果上完全兼容
- FirmwareVolume: `NO`
  - 修复 FileVault 的 UI 问题, 设置为 `YES` 可以获得更好的 FileVault 兼容性
- HashServices: `NO`
  - 修复运行 FileVault 时鼠标光标大小不正确的问题, 设置为 `YES` 可以更好兼容 FileVault
- UnicodeCollation: `NO`

- 一些较旧的固件破坏了 Unicode 排序规则, 开启以修复这些系统上 UEFI Shell 的兼容性 (通常为用于 IvyBridge 和更旧的设备)

#### Quirks:

- **AvoidHighAlloc:** NO
  - 主板无法正确访问 UEFI Boot Services 中更高内存的解决方法。除非有必要, 否则避免使用
- **ExitBootServicesDelay:** 0
- **IgnoreInvalidFlexRatio:** NO
  - BIOS 中无法禁用 MSR\_FLEX\_RATIO(0x194) 时开启
- **IgnoreTextInGraphics:** NO
  - 修复不用 -v 开机时日志覆盖苹果标志输出的问题
- **ProvideConsoleGop:** YES
  - macOS 引导加载程序要求 GOP (图形输出协议) 存在于控制台句柄上。启用后会在缺少该协议的设备上安装。
- **ReleaseUsbOwnership:** NO
  - 从固件驱动程序中释放 USB 控制器所有权, 除非您不知道自己在做什么, 否则避免使用。Clover 的等效设置是 FixOwnership
- **RequestBootVarRouting:** NO
  - 从 EFI\_GLOBAL\_VARIABLE\_GUID 中为 OC\_VENDOR\_VARIABLE\_GUID 请求 redirectBoot 前缀变量
  - 启用此项以便能够在设计上与 macOS 引导项不兼容的固件中可靠地使用 启动磁盘 设置
- **ReplaceTabWithSpace:** NO
  - 取决于固件, 某些设备在使用 Tab 编辑 UEFI Shell 中的文件出问题启用。注意, 此选项需要将 ConsoleControl 设置为 YES
- **SanitiseClearScreen:** YES
  - 修复 OpenCore 在高分屏中以 1024x768 显示的问题, 注意要同时开启 ConsoleControl 并将 ConsoleMode 的内容留空
- **ClearScreenOnModeSwitch:** NO
  - 从图形模式切换到文本模式时, 某些固件仅清除屏幕的一部分, 导致屏幕上残留之前绘制的图片。此选项会在切换到文本模式之前用黑色填充整个屏幕

你的 EFI 现在应该是什么样子:

EFI			
Name	Date Modified	Size	Kind
▼ BOOT	Today at 11:07 PM	--	Folder
BOOTx64.efi	Sep 10, 2019 at 4:03 PM	281 KB	Document
▼ OC	Today at 11:07 PM	--	Folder
▼ ACPI	Sep 13, 2019 at 9:11 PM	--	Folder
SSDT-ALS0.aml	Sep 13, 2019 at 8:04 PM	130 bytes	ACPI Machine Language Binary
SSDT-EC-USBX	Sep 13, 2019 at 8:58 PM	278 bytes	ACPI Machine Language Binary
SSDT-I2CBus.aml	Sep 13, 2019 at 9:11 PM	77 bytes	ACPI Machine Language Binary
SSDT-PLUG	Sep 12, 2019 at 11:21 PM	98 bytes	ACPI Machine Language Binary
SSDT-PNLF	Sep 12, 2019 at 11:32 PM	1 KB	ACPI Machine Language Binary
SSDT-RMNE.aml	Sep 12, 2019 at 11:33 PM	240 bytes	ACPI Machine Language Binary
config.plist	Sep 15, 2019 at 8:05 PM	17 KB	Property List
▼ Drivers	Sep 13, 2019 at 10:13 PM	--	Folder
ApfsDriverLoader.efi	Sep 7, 2019 at 1:31 PM	19 KB	Document
AppleGenericInput.efi	Sep 7, 2019 at 1:31 PM	15 KB	Document
EmuVariableRuntimeDxe.efi	Sep 12, 2019 at 10:54 PM	27 KB	Document
FwRuntimeServices.efi	Sep 7, 2019 at 1:31 PM	6 KB	Document
HFSPlus.efi	Sep 12, 2019 at 10:54 PM	25 KB	Document
VirtualSmc.efi	Aug 11, 2019 at 12:32 PM	7 KB	Document
▼ Kexts	Today at 11:08 PM	--	Folder
AppleALC.kext	Sep 13, 2019 at 4:35 PM	58 KB	Package
CPUFriend.kext	Aug 11, 2019 at 12:24 PM	44 KB	Package
Lilu.kext	Aug 11, 2019 at 12:16 PM	153 KB	Package
NoTouchID.kext	Jun 8, 2019 at 3:10 PM	31 KB	Package
NullEthernet.kext	Dec 21, 2016 at 9:43 AM	47 KB	Package
SMCBatteryManager.kext	Aug 11, 2019 at 12:32 PM	74 KB	Package
SMCLightSensor.kext	Aug 11, 2019 at 12:32 PM	37 KB	Package
SMCProcessor.kext	Aug 11, 2019 at 12:32 PM	43 KB	Package
SMCSuperIO.kext	Aug 11, 2019 at 12:32 PM	50 KB	Package
USBPorts.kext	Sep 13, 2019 at 9:02 PM	4 KB	Package
VirtualSMC.kext	Aug 11, 2019 at 12:32 PM	109 KB	Package
VoodooI2C.kext	Jul 13, 2019 at 1:10 AM	514 KB	Package
VoodooI2CHID.kext	Jul 13, 2019 at 1:09 AM	224 KB	Package
WhateverGreen.kext	Sep 7, 2019 at 1:40 PM	471 KB	Package
OpenCore.efi	Sep 10, 2019 at 4:03 PM	281 KB	Document
▼ Tools	Sep 13, 2019 at 10:21 AM	--	Folder
libaistat.dylib	Aug 11, 2019 at 12:32 PM	61 KB	Mach-O dynamic library
rtcread	Aug 11, 2019 at 12:32 PM	14 KB	Unix executable
smc	Aug 11, 2019 at 12:32 PM	23 KB	Unix executable
smcread	Aug 11, 2019 at 12:32 PM	27 KB	Unix executable
VerifyMsrE2.efi	Sep 7, 2019 at 1:31 PM	9 KB	Document

- 现在是时候开机见分晓了



## • 解决 Clover 和 OpenCore 的冲突

### - 清理 Clover 的模拟 NVRAM RC 脚本并换用 OpenCore 自带的 LogoutHook (请按如下顺序执行)

1. 对于 RC 脚本, 需要删除 macOS 下的如下文件(夹):

- `/Volumes/EFI/nvram.plist`
- `/etc/rc.clover.lib`
- `/etc/rc.boot.d/10.save_and_rotate_boot_log.local`
- `/etc/rc.boot.d/20.mount_ESP.local`
- `/etc/rc.boot.d/70.disable_sleep_proxy_client.local.disabled`
- `/etc/rc.boot.d/80.save_nvram.plist.local`
- `/etc/rc.boot.d`
- `/etc/rc.shutdown.d`

■ 经群友证实, RC 脚本同样适用于 OpenCore

### - 重置原生 NVRAM

■ 为了尽可能减少问题出现的概率, 建议在 Clover 和 OpenCore 之间切换时重置 NVRAM

1. 通过 OpenCore 引导界面的重置 NVRAM 选项或者快捷键 `Command + Option + R + R`
2. 通过 Clover F11 清除

### - Clover 设置面板会和 OpenCore 产生冲突, 需要删除

Clover 设置面板位于 `/Library/PreferencePanes/Clover.prefPane`

可以在终端输入 `sudo rm -rf /Library/PreferencePanes/Clover.prefPane` 删除

## 附录[1] -- OpenCore 支持的内核驱动 (Kext) 及其用途

- 参考 <https://github.com/acidanthera/OpenCorePkg/blob/master/Docs/Kexts.md>
- GitHub 项目的下载地址位于 Release 下, 驱动具体支持的设备请进入项目地址查看

- 有线网卡

驱动及项目地址	备注
<a href="#">AppleRTL8169Ethernet</a>	Realtek RTL8169 官方驱动
<a href="#">AtherosE2200Ethernet.kext</a>	高通 Atheros Killer E2200 系列驱动
<a href="#">AtherosL1cEthernet.kext</a>	高通 Atheros AR813x/815x 驱动
<a href="#">IntelMausi.kext</a>	英特尔有线网卡 Acidanthera 分支
<a href="#">IntelMausiEthernet.kext</a>	英特尔有线网卡原作者
<a href="#">NullEthernetInjector.kext</a>	仿冒内建网卡 (没有可用的内建网卡时使用)
<a href="#">RealtekR1000SL.kext</a>	Realtek 8111B/C/D/E/EP/F/G/GU/8411B 系列驱动
<a href="#">RealtekRTL8100.kext</a>	Realtek RTL810X 系列驱动
<a href="#">RealtekRTL8111.kext</a>	Realtek RTL8111/8168 系列驱动

- Wi-Fi 和蓝牙

驱动及项目地址	备注
<a href="#">AirPortAtheros40.kext</a>	高通 Atheros AR92xx/AR93xx 驱动, 仅用于 10.14+
<a href="#">AirportBrcmFixup.kext</a>	非苹果官方博通网卡修复
<a href="#">ATH9KFixup.kext</a>	高通 Atheros AR9xxx 无线网卡修复
<a href="#">BrcmPatchRAM.kext</a>	博通网卡蓝牙固件上传
<a href="#">BT4LEContinuityFixup.kext</a>	IOBluetoothFamily 修补
<a href="#">MT7610</a>	联发科 MT7610
<a href="#">RT5370</a>	联发科 RT5370
<a href="#">RTL8192CU</a>	Realtek RTL8192CU 驱动 (链接为谷歌云盘)

- 键盘, 鼠标和触摸设备



驱动及项目地址	备注
<a href="#">ApplePS2SmartTouchPad.kext</a>	触摸板和键盘
<a href="#">AsusSMC.kext</a>	华硕 Fn 键, 键盘背光灯和环境光传感器 驱动
<a href="#">NoTouchID.kext</a>	禁用 Touch ID 检测, 修复输密码时卡顿
<a href="#">SerialMouse.kext</a>	使用 Microsoft 串行鼠标协议的串行鼠标驱动
<a href="#">VoodooI2C.kext</a>	I2C 触摸板/屏 驱动
<a href="#">VoodooPS2Controller.kext</a>	PS2 键盘/触摸板 驱动

## • 显卡和声卡

驱动及项目地址	备注
<a href="#">AppleALC.kext</a>	定制万能声卡驱动
<a href="#">NVIDIA CUDA drivers</a>	NVIDIA CUDA 驱动
<a href="#">NVIDIA Web-drivers</a>	NVIDIA 显卡驱动
<a href="#">SNBGraphicsMojaveInstaller</a>	二代酷睿核显驱动, 仅用于 10.14+
<a href="#">VoodooHDA.kext</a>	万能声卡驱动
<a href="#">WhateverGreen.kext</a>	显卡补丁驱动
<a href="#">Polaris22Fixup.kext</a>	Polaris22/VegaM 显卡修复

## • CPU 和 SMC

驱动及项目地址	备注
<a href="#">CPUFriend.kext</a>	CPU 变频管理
<a href="#">FakeSMC.kext and sensors</a>	Clover 官方 FakeSMC
<a href="#">HWPEnabler.kext</a>	启用 HWP
<a href="#">NullCPUPowerManagement.kext</a>	AMD 和虚拟机专用版本
<a href="#">OpcodeEmulator.kext</a>	Opcode 模拟驱动
<a href="#">TSCAdjustReset.kext</a>	TSC 频率同步驱动
<a href="#">VirtualSMC.kext 及传感器</a>	Acidanthera 的 SMC 和传感器驱动
<a href="#">VoodooTSCSync.kext</a>	TSC 频率同步驱动

## • USB 和 其它接口

驱动及项目地址	备注
<a href="#">IOElectrify.kext</a>	在雷电 3 设备上启用常开电源
<a href="#">USBWakeFixup.kext</a>	修复 Skylake 平台 USB 唤醒黑屏
<a href="#">SASMegaRAID.kext</a>	LSI MegaRAID SAS 系列 RAID 控制器驱动
<a href="#">Sinetek-rtsx.kext</a>	Realtek <code>RTSX</code> SDHC 读卡器驱动
<a href="#">VoodooSDHC.kext</a>	SDHC 读卡器驱动

## • 其它驱动

驱动及项目地址	备注
<a href="#">AppleIntelInfo.kext</a>	CPU / 核显 变频测试
<a href="#">HibernationFixup.kext</a>	修复因 RTC 变量和 NVRAM 造成的睡眠问题
<a href="#">Lilu.kext</a>	SDK & Library
<a href="#">LiluFriend.kext</a>	用于确保 Lilu 在 L/E 下正常加载
<a href="#">RTCMemoryFixup.kext</a>	修复 BIOS CMOS (RTC) 内存和 AppleRTC 之间的冲突问题
<a href="#">NightShiftUnlocker.kext</a>	解锁 NightShift
<a href="#">WebCamera.kext</a>	某些旧设备的摄像头驱动

## 附录[2] -- UEFI 驱动 (.efi) 及其用途

### • OpenCore 支持的 UEFI 驱动

- [ApfsDriverLoader](#)
  - APFS 文件系统引导驱动
- [AppleGenericInput](#)
  - 添加了对 `AppleKeyMapAggregator` 协议支持的用户输入驱动。此外, 还解决了某些固件上的鼠标输入问题, 这是 `UsbKbDxe` 的等效驱动, 根据固件的不同, 实际效果会更好或更坏
- [CrScreenshotDxe](#)
  - 增加 OpenCore UI 内截屏功能, 快捷键为 `LCtrl + LAlt + F12` (左Ctrl + 左Alt + F12)
- [EmuVariableRuntimeDxe](#)
  - EDK II ([UDK2018](#)) 官方的模拟 NVRAM 驱动, 需要配合 `FwRuntimeServices` (.efi) 和 `DisableVariableWrite` (Quirk), OpenCore 可能随时停止支持此驱动
- [FwRuntimeServices](#)
  - 通过支持只读和只写 NVRAM 变量来提高 OpenCore 和 Lilu 的安全性。有些设置项, 例如 `RequestBootVarRouting`, 需要此驱动程序才能正常运行
- [HFSPlus](#)
  - 苹果自带的闭源 HFS 驱动, 不具有 `bless` 和其它功能, 但是启动速度比它的等效驱动 `VBoxHfs` 快 3 倍
- [NvmExpressDxe](#)
  - 从 Broadwell 开始的大多数固件中都包含此驱动程序。对于 Haswell 及更早的固件, 如果安装了 NVMe SSD 就需要使用
- [MemoryAllocation.efi](#)

- 为 Z390/X99 等主板预留第一组 512MB 内存, 帮助引导工具注入内核以及内核缓存至第一组 512MB 内存, 需要配合 FwRuntimeServices 和引导标识符 `slide=1`
- [UsbKbDxe](#)
  - 添加了对 `AppleKeyMapAggregator` 协议支持的 USB 键盘驱动, 这是 `AppleGenericInput` 的等效驱动, 根据固件的不同, 实际效果会更好或更坏
- [VariableRuntimeDxe](#)
  - EDK II 官方的 NVRAM 驱动, OpenCore 中用于模拟 NVRAM, 需要配合 `FwRuntimeServices` (.efi) 和 `DisableVariableWrite` (Quirk), 在某些固件上可能报错 ([UDK2018](#) 下的 `EmuVariableRuntimeDxe` 是与其合并之前的版本, 报错概率较低)
- [VirtualSmc.efi](#)
  - UEFI SMC 驱动程序, 对于 FileVault 2 和 macOS 的各个功能是必需的。OpenCore 使用 FakeSMC 时, 需要手动添加 NVRAM 变量, 并且应继续使用 `VirtualSmc` .efi 驱动
- [VBoxHfs](#)
  - 具有 `bless` 支持的 HFS 文件系统驱动程序。该驱动程序可以等效替代 Apple 固件中常见的闭源 HFSPlus 驱动。此驱动虽然功能更加完善, 但启动速度相比大约慢 3 倍, 并且尚未经过安全审核
- [XhciDxe](#)
  - 来自 MdeModulePkg 的 XHCI USB 控制器支持驱动程序。从 Sandy Bridge 开始的大多数固件中都包含此驱动程序。在较旧的固件上可以用于支持 USB 3.0 PCI 卡

## • OpenCore 不兼容的 UEFI 驱动

- [AppleImageCodec](#)
  - 为 Clover 启动 FileVault 2 解码 PNG 和 BMP, OpenCore 已集成
- [AppleKeyAggregator](#)
  - 为 Clover 支持 FileVault 2 启动 UI 的驱动, OpenCore 已集成
- [AppleKeyFeeder](#)
  - 为 Clover 支持 FileVault 2 内 PS/2 键盘输入的驱动, OpenCore 不兼容, 替代品为 `AppleGenericInput`
- [AppleUITheme](#)
  - 为 Clover 支持 FileVault 2 启动 UI 主题的驱动, OpenCore 已集成
- [AptioInputFix](#)
  - 用于解决 UEFI 固件输入问题的驱动, 已更名为 `AppleGenericInput` 并计划在未来与 OpenCore 合并
- [AptioMemoryFix](#)
  - NVRAM 和内存驱动, 用于修复 UEFI 固件上的内存问题, 已与 OpenCore 合并为 `FwRuntimeServices`
- [AudioDxe](#)
  - 用于在 Clover 启动时播放声音的 HDA 驱动, OpenCore 兼容性未知
- [CsmVideoDxe](#)
  - 用于 Clover GUI 的显卡驱动, 允许使用更多分辨率, 基于 UEFI BIOS 中的 CSM 兼容模块, 并将启用所需的 CSM, OpenCore 不兼容
- [DataHubDxe](#)
  - macOS 必需的 DataHub 协议, OpenCore 自带且提供了这个 Quirk
- [EmuVariableUefi](#)
  - Clover 的模拟 NVRAM 驱动, OpenCore 不兼容, 替代品为 `EmuVariableRuntimeDxe` (UDK2018) 或 `VariableRuntimeDxe` (EDK II)
- [EnglishDxe](#)
  - 在 UEFI Shell 中支持 UnicodeCollation 协议, OpenCore 自带且提供了这个 Quirk
- [EnhancedFatDxe](#)
  - 这个驱动已存在于所有 UEFI 固件中, 无法从 OpenCore 直接使用。众所周知, 很多固件的 FAT 支持都有问题, 这会导致在尝试写入时损坏文件系统。如果在引导过程中写入 EFI 分区出现问题, 则需要将此驱动用 [UEFITool](#) 刷入固件中
- [FirmwareVolume](#)

- 为 Clover 启动 FileVault 2 创建 FirmwareVolume 光标的驱动, OpenCore 已集成
- [FSInject](#)
  - Clover 用于注入内核驱动 (Kext) 的驱动, OpenCore 自带且使用更先进的方法
- [HashServiceFix](#)
  - 修复 UEFI BIOS 中的哈希支持, OpenCore 自带且提供了这个 Quirk
- [OsxAptioFix3Drv](#)
  - Clover NVRAM 和内存驱动, 用于修复 UEFI 固件上的内存问题, 与 FwRuntimeServices 和 OpenCore 不兼容
- [OsxAptioFixDrv](#)
  - 旧的 Clover NVRAM 和内存驱动, 用于修复 UEFI 固件上的内存问题, 与 FwRuntimeServices 和 OpenCore 不兼容
- [OsxFatBinaryDrv](#)
  - Clover 用于支持 OS X 10.9 和更早版本的 FAT Binary 可执行文件的驱动, 与 FwRuntimeServices 和 OpenCore 不兼容
- [OsxLowMemFixDrv](#)
  - 精简版的 OsxAptioFixDrv, 用于修复 UEFI 固件上的内存问题, 与 FwRuntimeServices 和 OpenCore 不兼容
- [PartitionDxe](#)
  - 用于支持非常规分区图的驱动, 如: 混合 GPT/MBR 或 Apple 分区图, OpenCore 兼容性未知
- [Ps2MouseDxe](#)
  - PS/2 鼠标驱动, 这个驱动已存在于所有 UEFI 固件中
- [SMCHelper](#)
  - SMCHelper 是 VirtualSmc.efi 在 Clover 中的替代品, 用于与 FakeSMC 配合。与 VirtualSmc 和 OpenCore 不兼容
- 其它未在本文中提到的 UEFI 驱动一律不兼容
- UEFI 驱动下载地址

OpenCore UEFI Drivers 点击下载

## 附录[3] -- 常用内核引导标识符合集

引导标识符	作用
<code>-amd_no_dgpu_accel</code>	关闭 A 卡硬件加速
<code>cpus=#</code>	启用 <code>#</code> 个 CPU 核心
<code>darkwake=0</code>	禁用 Power Nap
<code>dart=0</code>	禁用 VT-d
<code>debug=0x100</code>	发生 KP 时不自动重启
<code>keepsyms=1</code>	发生 KP 时保留 Debug Symbols, 用于给开发者反馈问题
<code>kext-dev-mode=1</code>	启用 Kext 开发模式, 非开发者请勿使用
<code>-no_compat_check</code>	关闭兼容性检查
<code>npci=0x2000</code>	在旧设备上禁用 kIOPCIConfiguratorPFM64
<code>nvda_drv=1</code>	启用 N 卡驱动的老方法, 在 10.12 及以后失效
<code>nv_disable=1</code>	关闭 N 卡硬件加速
<code>-s</code>	单用户模式
<code>slide=#</code>	手动设置 KASLR slide 值为 <code>#</code>
<code>-v</code>	verbose 跑码模式
<code>-x</code>	安全模式

■ 驱动提供的引导标识符请去该驱动的 Readme 查看

## 参考及引用

- [Apple](#) 的 macOS
- [Acidanthera](#) 的 OpenCore 及其[官方文档](#)
- [khronokernel](#) 的 [Getting-Started-With-OpenCore](#) 和 [Opencore-Vanilla-Desktop-Guide](#)
- [Tonymacx86](#) 的 [Emulated-NVRAM-Uninstaller](#)

## 鸣谢

- [宪武](#) 提供的 OpenCore SSDT Hotpatch 典范 [OC-little](#)
- [xlivans](#), [子俊](#) 对 OpenCore 的大量跟进和研究
- [xjn](#) 对于内存分配的研究
- OpenCore 交流群成员进行的大量测试
- [Bat.bat](#) 对教程对审核更新完善

■ 教程更新于 2019.10.1, 基于 OpenCore 0.5.1 (2019.10.1) 自编译

■ 目前官方版本为 0.5.0