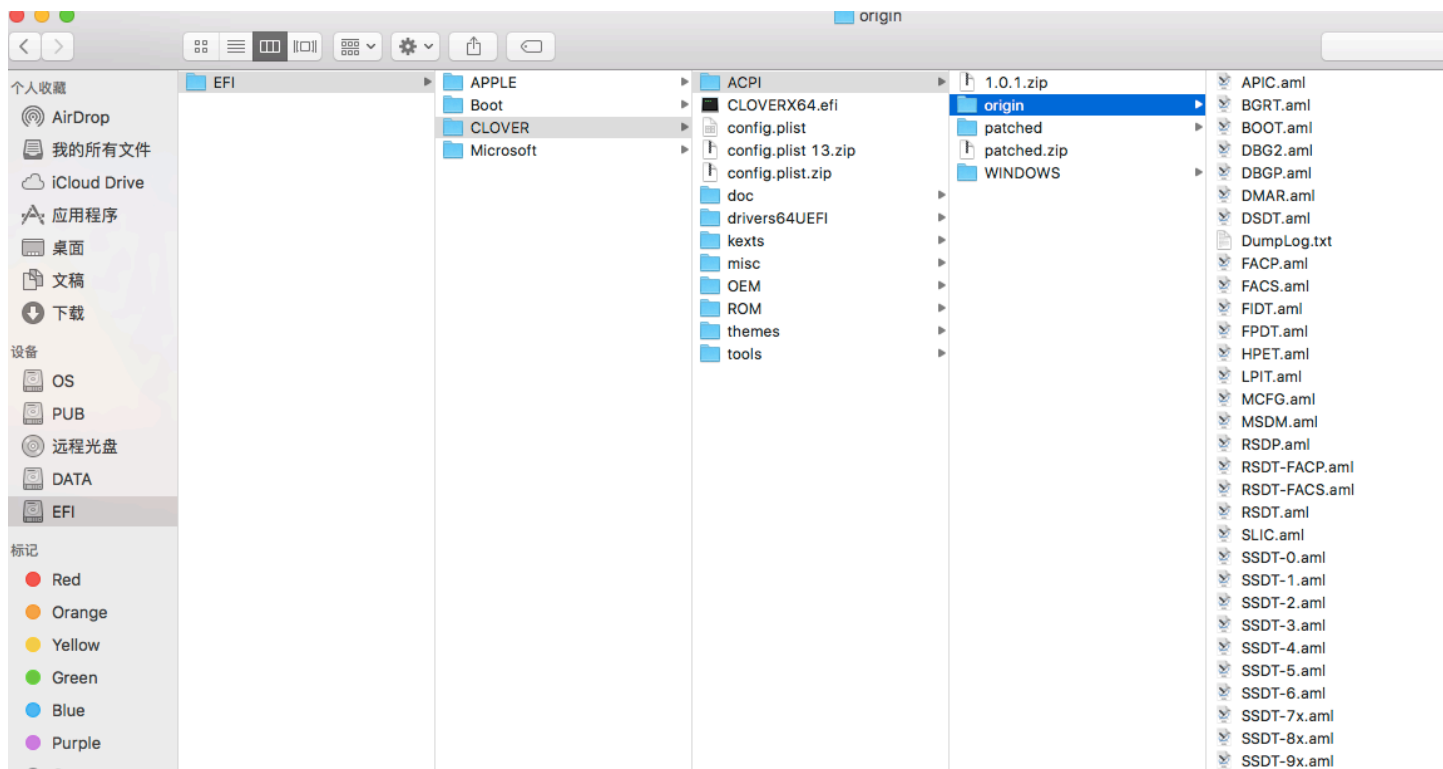


此教程面向具有一定计算机基础的新手，如果你尚不明白二进制和十六进制是什么，那么可以放弃了。

下面从 DSDT SSDT 补丁开始（参考原教程<https://www.tonymacx86.com/threads/guide-patching-laptop-dsdt-ssdts.152573/>）

提取DSDT SSDT 文件：在 clover 界面，按 F4会自动提取所有的相关的文件，存放目录在/EFI/CLOVER/APCI/origin



拆分 DSDT 及 SSDT 文件（反编译）：

复制 origin 文件夹到桌面，仅保留 DSDT.aml SSDT*.aml 删除多余文件

保存以下文本为refs.txt，并置于 origin 文件夹内

```
External(MDBG, MethodObj, 1)
External(_GPE.MMTB, MethodObj, 0)
External(_SB.PCI0.LPCB.H_EC.ECWT, MethodObj, 2)
External(_SB.PCI0.LPCB.H_EC.ECRD, MethodObj, 1)
External(_SB.PCI0.LPCB.H_EC.ECMD, MethodObj, 1)
External(_SB.PCI0.PEG0.PEGP.SGPO, MethodObj, 2)
External(_SB.PCI0.GFX0.DD02._BCM, MethodObj, 1)
External(_SB.PCI0.SAT0.SDSM, MethodObj, 4)
External(_GPE.VHOV, MethodObj, 3)
External(_SB.PCI0.XHC.RHUB.TPLD, MethodObj, 2)
```

origin 内包含的原始文件应如下:

名称	修改日期	大小	种类
SSDT-15.aml	今天 15:22	7 KB	ACPI...
SSDT-14.aml	今天 15:22	209 字节	ACPI...
SSDT-13x.aml	今天 15:22	304 字节	ACPI...
SSDT-12x.aml	今天 15:22	281 字节	ACPI...
SSDT-11x.aml	今天 15:22	142 字节	ACPI...
SSDT-10x.aml	今天 15:22	281 字节	ACPI...
SSDT-9x.aml	今天 15:22	895 字节	ACPI...
SSDT-8x.aml	今天 15:22	1 KB	ACPI...
SSDT-7x.aml	今天 15:22	1 KB	ACPI...
SSDT-6.aml	今天 15:22	4 KB	ACPI...
SSDT-5.aml	今天 15:22	21 KB	ACPI...
SSDT-4.aml	今天 15:22	14 KB	ACPI...
SSDT-3.aml	今天 15:22	3 KB	ACPI...
SSDT-2.aml	今天 15:22	4 KB	ACPI...
SSDT-1.aml	今天 15:22	584 字节	ACPI...
SSDT-0.aml	今天 15:22	789 字节	ACPI...
refs	今天 23:42	426 字节	纯文本
DSDT.aml	今天 15:22	103 KB	ACPI...

打开终端, 进入 origin

```
$ cd Desktop/origin
```

#终端内查看文件, 与上面是相符的

```
$ ls
```

```
DSDT.aml      SSDT-10x.aml  SSDT-13x.aml  SSDT-2.aml    SSDT-5.aml    SSDT-8x.aml
SSDT-0.aml    SSDT-11x.aml SSDT-14.aml   SSDT-3.aml    SSDT-6.aml    SSDT-9x.aml
SSDT-1.aml    SSDT-12x.aml SSDT-15.aml   SSDT-4.aml    SSDT-7x.aml   refs.txt
```

开始反编译:

```
$ iasl -da -dl -fe refs.txt DSDT.aml SSDT*.aml
```

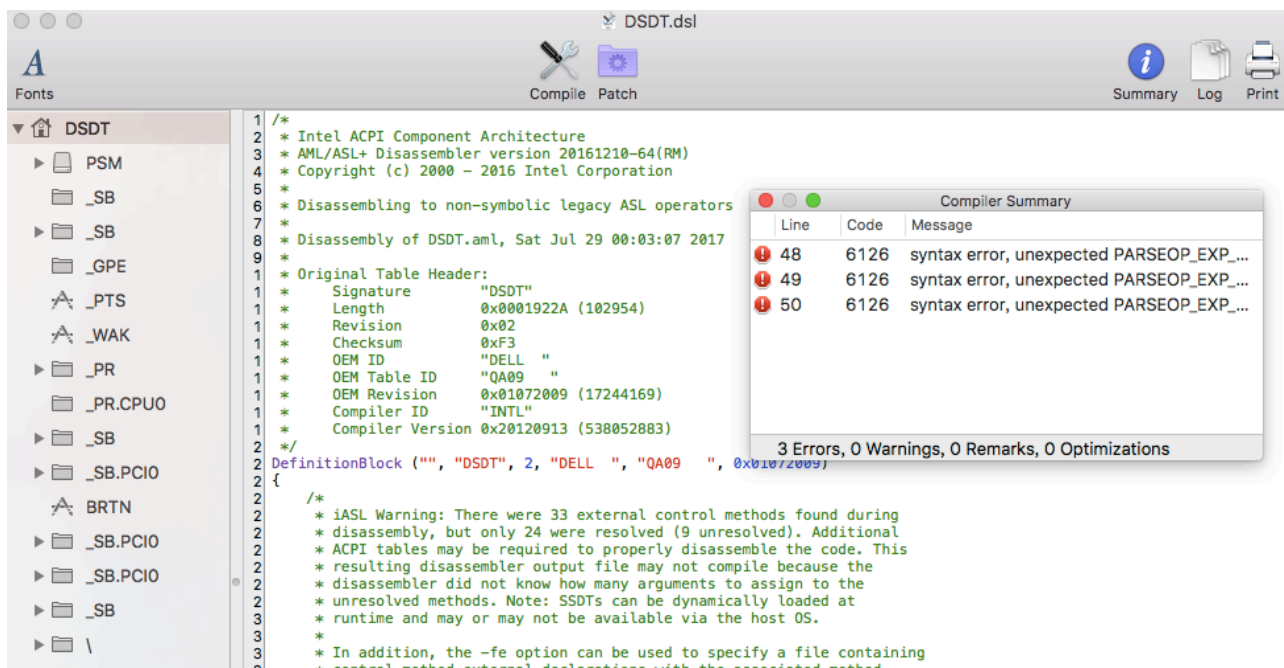
看到如下输出, 结束且没有报错即成功

```
Intel ACPI Component Architecture
ASL+ Optimizing Compiler version 20161210-64(RM)
Copyright (c) 2000 - 2016 Intel Corporation

Input file DSDT.aml, Length 0x1922A (102954) bytes
ACPI: DSDT 0x0000000000000000 01922A (v02 DELL QA09 01072009 INTL 20120913)
External object resolution file      SSDT-9x.aml
Input file SSDT-9x.aml, Length 0x37F (895) bytes
ACPI: SSDT 0x0000000000000000 00037F (v02 PmRef Cpu0Cst 00003001 INTL 20120913)
Pass 1 parse of [SSDT]
Pass 2 parse of [SSDT]
External object resolution file      SSDT-8x.aml
Input file SSDT-8x.aml, Length 0x5AA (1450) bytes
ACPI: SSDT 0x0000000000000000 0005AA (v02 PmRef ApIst 00003000 INTL 20120913)
```

初步错误修复 (MaciASL.app)

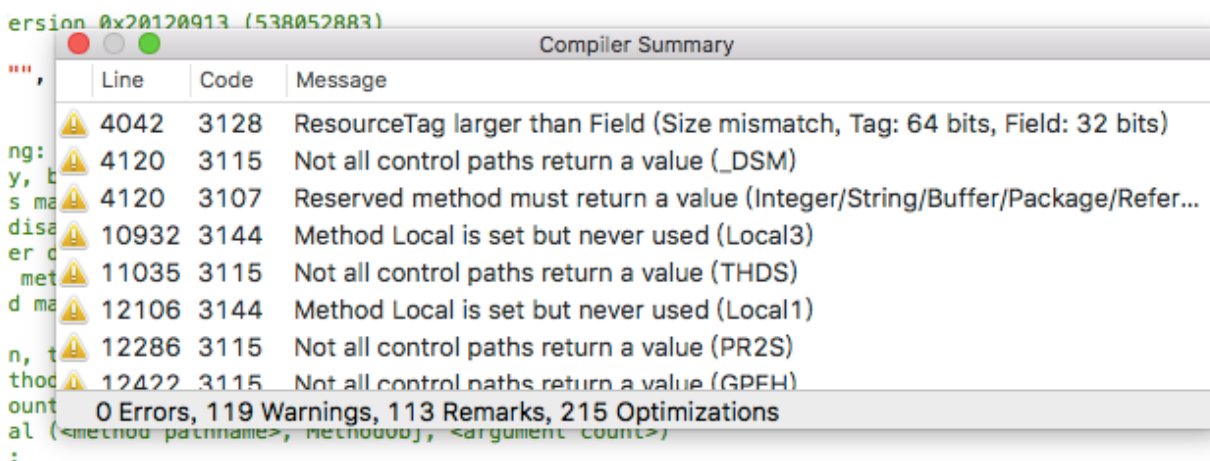
打开 DSDT.dsl, 点击 Compile, 在弹出提示框可见到3个错误



定位到相关位置, 确定是无效数据, 删之。

```
4  */
4  /*
4  * External declarations were imported from
4  * a reference file -- refs.txt
4  */
4
4  External (**F, UnknownObj) // Warning: Unknown object
4  External (**I, UnknownObj) // Warning: Unknown object
4  External (*X*F, UnknownObj) // Warning: Unknown object
5  External (_GPE.HLV, MethodObj) // 0 Arguments
5  External (_GPE.MMTB, MethodObj) // Imported: 0 Arguments
5  External (_GPE.P0L6, MethodObj) // 0 Arguments
5  External (_GPE.P1L6, MethodObj) // 0 Arguments
5  External (_GPE.P2L6, MethodObj) // 0 Arguments
5  External (_GPE.VHNV, MethodObj) // Imported: 3 Arguments
```

再次Compile 无错误 (使用 refs.txt 进行反编译的好处——减少很多常见的错误)



双显卡笔记本屏蔽独显：

(<https://www.tonymacx86.com/threads/guide-disabling-discrete-graphics-in-dual-gpu-laptops.163772/>)

原作者写道

Patching requirements

The goal is quite simple. These machines provide an `_OFF` method, usually in an SSDT, that can be called to power down the discrete device. The easiest fix is to call `_OFF` from the corresponding `_INI` method. Note that the `_OFF` method may be in DSDT or may have a different name (GPOF, OPOF, `_PS3`, etc.)

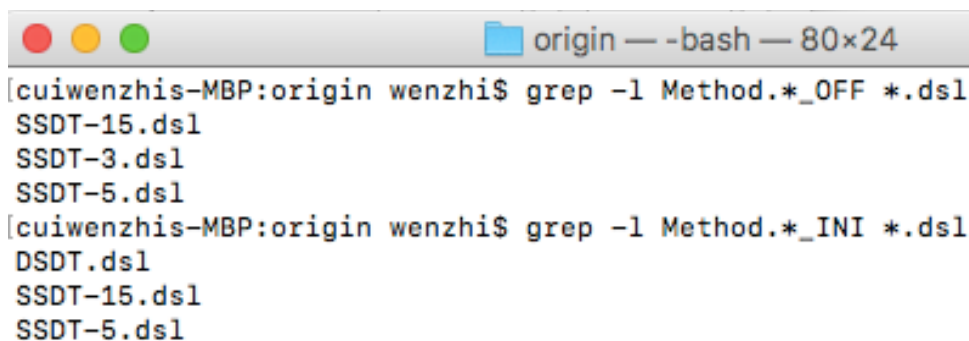
意思为，在机器提供的代码里，与独立显卡电源控制相关的代码是通常是在某个 SSDT 里的“`_OFF` method”。最简便的补丁方式是通过与之相关的“`_INI` method”来调用“`_OFF`”功能，即可实现关闭独立显卡的电源。

需要注意的是，“`_INI` method”有可能存放在 DSDT.aml 里，甚至以 GPOF, OPOF, `_PS3` 这些项目名称存在。

首先，确定相关入口包含在哪个文件

```
$ grep -l Method.*_OFF *.dsl
```

```
$ grep -l Method.*_INI *.dsl
```



```
origin — -bash — 80x24
[cuiwenzhis-MBP:origin wenzhi$ grep -l Method.*_OFF *.dsl
SSDT-15.dsl
SSDT-3.dsl
SSDT-5.dsl
[cuiwenzhis-MBP:origin wenzhi$ grep -l Method.*_INI *.dsl
DSDT.dsl
SSDT-15.dsl
SSDT-5.dsl
```

修改订单文件应该是 DSDT.dsl、SSDT-3.dsl、SSDT-5.dsl、SSDT-15.dsl

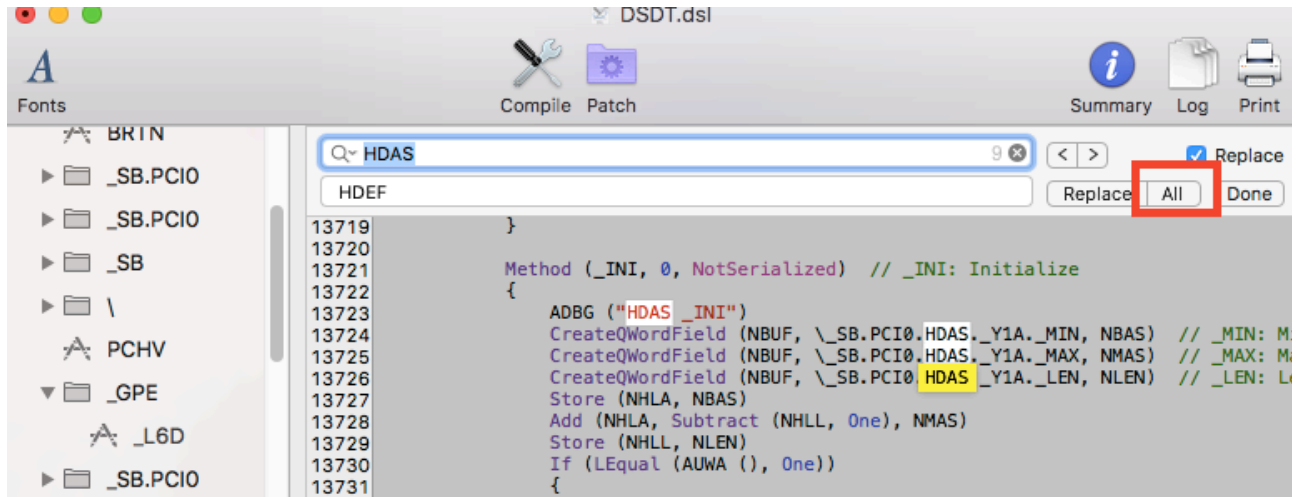
DSDT.dsl 文件的补丁及修改:

每完成一项操作进行一次 Compile 检测有无错误。

查找 HECI 替换为 IMEI

HDAS 替换为 HDEF

Done 结束操作



Dell 7559需要用到的 patch:

红标🚫为仅适用10.11.4-10.12.3版本系统, 10.12.4-10.12.6禁用

绿标✅为可选补丁, 对于10.12以后版本无特殊情况建议使用

- **[syn] Rename _DSM methods to XD SM**
- **[igpu] Rename GFX0 to IGPU**
- **[Audio] Audio Layout 3**
- **[bat] Dell Inspiron 15-7xxx**
- **[sys] Fix _WAK Arg0 v2** ✅
- **[sys] Fix Mutex with non-zero SyncLevel** ✅
- **[sys] HPET Fix** ✅
- **[sys] IRQ fix**
- **[sys] RTC Fix** ✅
- **[sys] Skylake LPC**
- **[sys] SMBUS Fix** ✅
- **[usb] USB3_PRW 0x6D Skylake**
- **[igpu] Brightness fix** 🚫

屏蔽独显需要用到的补丁:

```
External (_SB_.PCI0.PEG0.PEGP._PS3, MethodObj)
External (_SB_.PCI0.PEG0.PEGP._PS0, MethodObj)
External (_SB_.PCI0.PEG0.PEGP._OFF, MethodObj)
External (_SB_.PCI0.PEG0.PEGP._ON, MethodObj)
External (_SB_.PCI0.PEG0.PEGP.SGOF, MethodObj)
```

```
Method (M_ON, 0, NotSerialized)
{
    If (CondRefOf (\_SB_.PCI0.PEG0.PEGP._ON))
    {
        \_SB_.PCI0.PEG0.PEGP._ON()
    }
    If (CondRefOf (\_SB_.PCI0.PEG0.PEGP._PS0))
    {
        \_SB_.PCI0.PEG0.PEGP._PS0()
    }
    If (CondRefOf (\_SB_.PCI0.PEG0.PEGP.SGON))
    {
        \_SB_.PCI0.PEG0.PEGP.SGON()
    }
}
```

```
Method (M_OF, 0, NotSerialized)
{
    If (CondRefOf (\_SB_.PCI0.PEG0.PEGP._OFF))
    {
        \_SB_.PCI0.PEG0.PEGP._OFF()
    }
    If (CondRefOf (\_SB_.PCI0.PEG0.PEGP._PS3))
    {
        \_SB_.PCI0.PEG0.PEGP._PS3()
    }
    If (CondRefOf (\_SB_.PCI0.PEG0.PEGP.SGOF))
    {
        \_SB_.PCI0.PEG0.PEGP.SGOF()
    }
}
```

M_OF ()

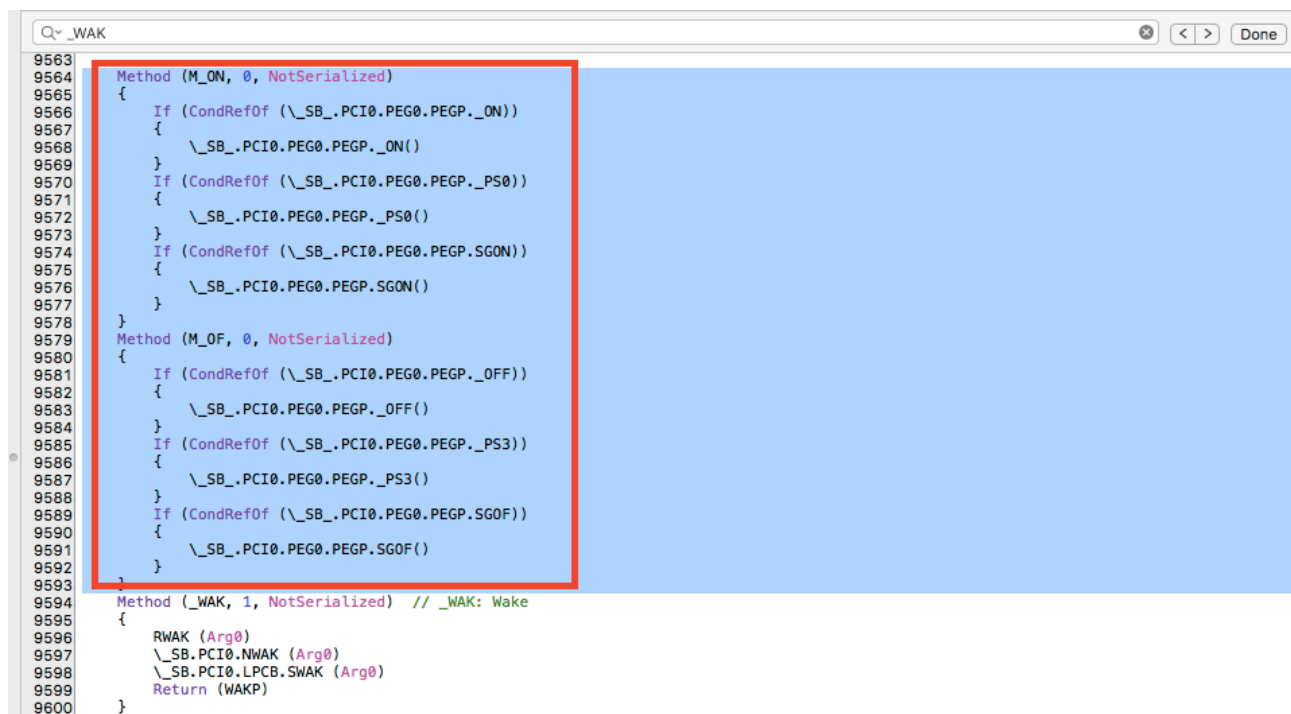
M_ON ()

在 DSDT.dsl 文件开头, 加入

```
/*
 * External declarations were imported from
 * a reference file -- refs.txt
 */

External (_SB_.PCI0.PEG0.PEGP._PS3, MethodObj)
External (_SB_.PCI0.PEG0.PEGP._PS0, MethodObj)
External (_SB_.PCI0.PEG0.PEGP._OFF, MethodObj)
External (_SB_.PCI0.PEG0.PEGP._ON, MethodObj)
External (_SB_.PCI0.PEG0.PEGP.SGOF, MethodObj)
External (_SB_.PCI0.PEG0.PEGP.SGON, MethodObj)
External (_GPE.WH0, MethodObj) // Imported: 0 Arguments
External (_GPE.P0L6, MethodObj) // 0 Arguments
External (_GPE.P1L6, MethodObj) // 0 Arguments
External (_GPE.P2L6, MethodObj) // 0 Arguments
External (_GPE.VH0V, MethodObj) // Imported: 3 Arguments
External (_PR_.BGIA, FieldUnitObj)
External (_PR_.BGMA, FieldUnitObj)
```

查找 _WAK, 在其之前添加



```
9563 Method (M_ON, 0, NotSerialized)
9564 {
9565     If (CondRefOf (\_SB_.PCI0.PEG0.PEGP._ON))
9566     {
9567         \_SB_.PCI0.PEG0.PEGP._ON()
9568     }
9569     If (CondRefOf (\_SB_.PCI0.PEG0.PEGP._PS0))
9570     {
9571         \_SB_.PCI0.PEG0.PEGP._PS0()
9572     }
9573     If (CondRefOf (\_SB_.PCI0.PEG0.PEGP.SGON))
9574     {
9575         \_SB_.PCI0.PEG0.PEGP.SGON()
9576     }
9577 }
9578 Method (M_OF, 0, NotSerialized)
9579 {
9580     If (CondRefOf (\_SB_.PCI0.PEG0.PEGP._OFF))
9581     {
9582         \_SB_.PCI0.PEG0.PEGP._OFF()
9583     }
9584     If (CondRefOf (\_SB_.PCI0.PEG0.PEGP._PS3))
9585     {
9586         \_SB_.PCI0.PEG0.PEGP._PS3()
9587     }
9588     If (CondRefOf (\_SB_.PCI0.PEG0.PEGP.SGOF))
9589     {
9590         \_SB_.PCI0.PEG0.PEGP.SGOF()
9591     }
9592 }
9593 Method (_WAK, 1, NotSerialized) // _WAK: Wake
9594 {
9595     RWAK (Arg0)
9596     \_SB_.PCI0.NWAK (Arg0)
9597     \_SB_.PCI0.LPCB.SWAK (Arg0)
9598     Return (WAKP)
9599 }
9600 }
```

在Method (_WAK内, 添加M_OF ()

```
9593 Method (_WAK, 1, NotSerialized) // _WAK: Wake
9594 {
9595     M_OF ()
9596     RWAK (Arg0)
9597     \_SB_.PCI0.NWAK (Arg0)
9598     \_SB_.PCI0.LPCB.SWAK (Arg0)
9599     Return (WAKP)
9600 }
9601 }
```


在Method (_PTS内添加M_ON ()

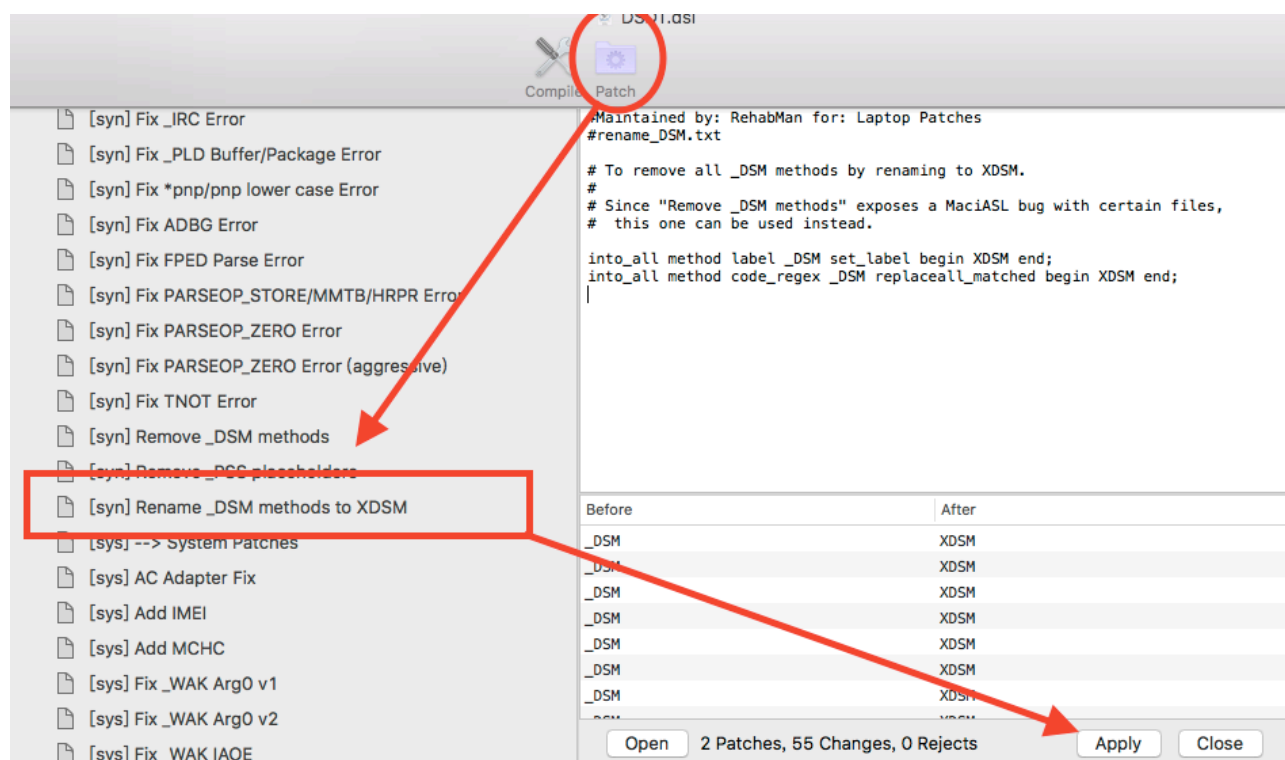
```
9553     })
9554     Method (_PTS, 1, NotSerialized) // _PTS: Prepare To Sleep
9555     {
9556         M_ON ()
9557         If (Arg0)
9558         {
9559             \_SB.PCI0.LPCB.SPTS (Arg0)
9560             \_SB.PCI0.NPTS (Arg0)
9561             RPTS (Arg0)
9562         }
9563     }
9564 }
```

查找Method (_INI, 定位到一段含Linux 及多个Windows的位置

在Store (0x07D9, OSYS)与If (CondRefOf (_OSI, Local0))之间添加M_OF ()

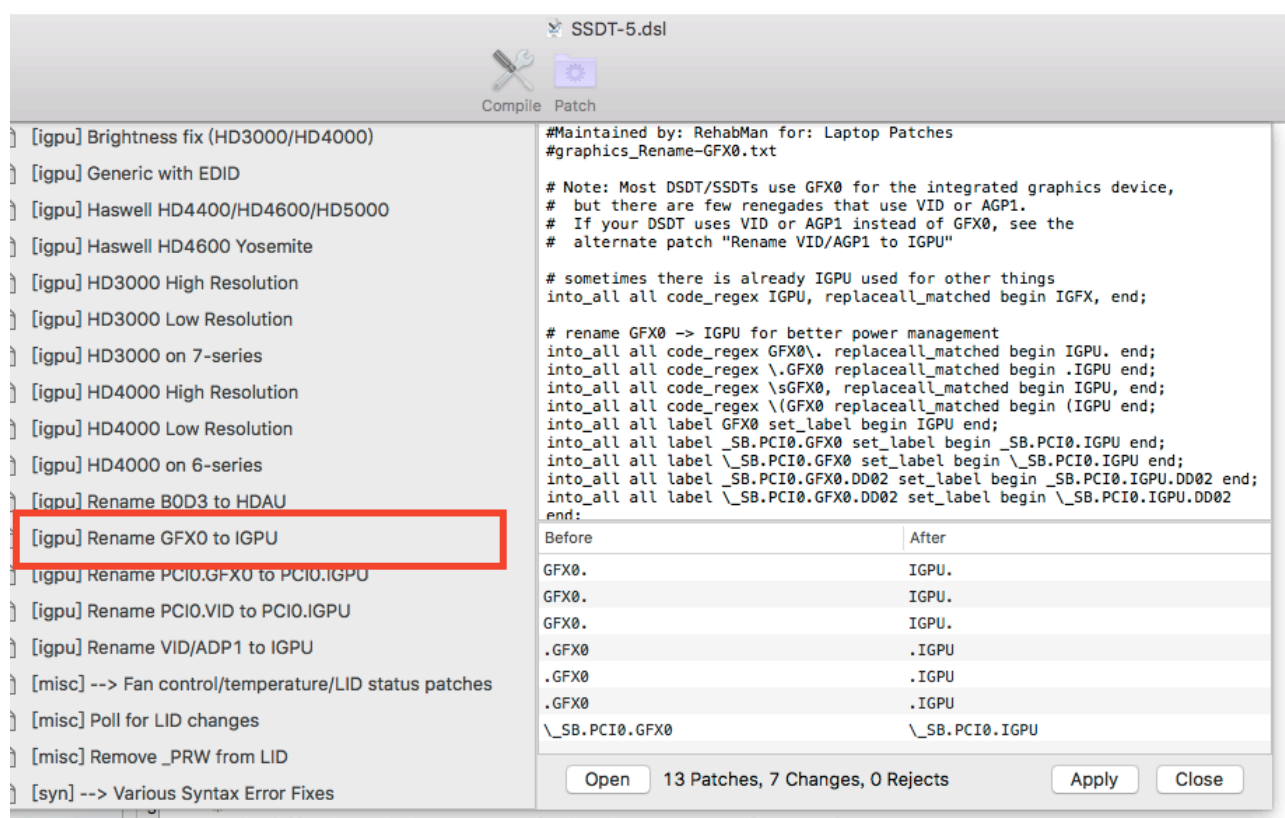
```
17921
17922     Method (_INI, 0, Serialized) // _INI: Initialize
17923     {
17924         Store (0x07D9, OSYS)
17925         M_OF ()
17926         If (CondRefOf (\_OSI, Local0))
17927         {
17928             If (_OSI ("Linux"))
17929             {
17930                 Store (0x03E8, OSYS)
17931             }
17932
17933             If (_OSI ("Windows 2001"))
17934             {
17935                 Store (0x07D1, OSYS)
17936             }
17937         }
17938     }
```


接下来打补丁，以第一个补丁为例：

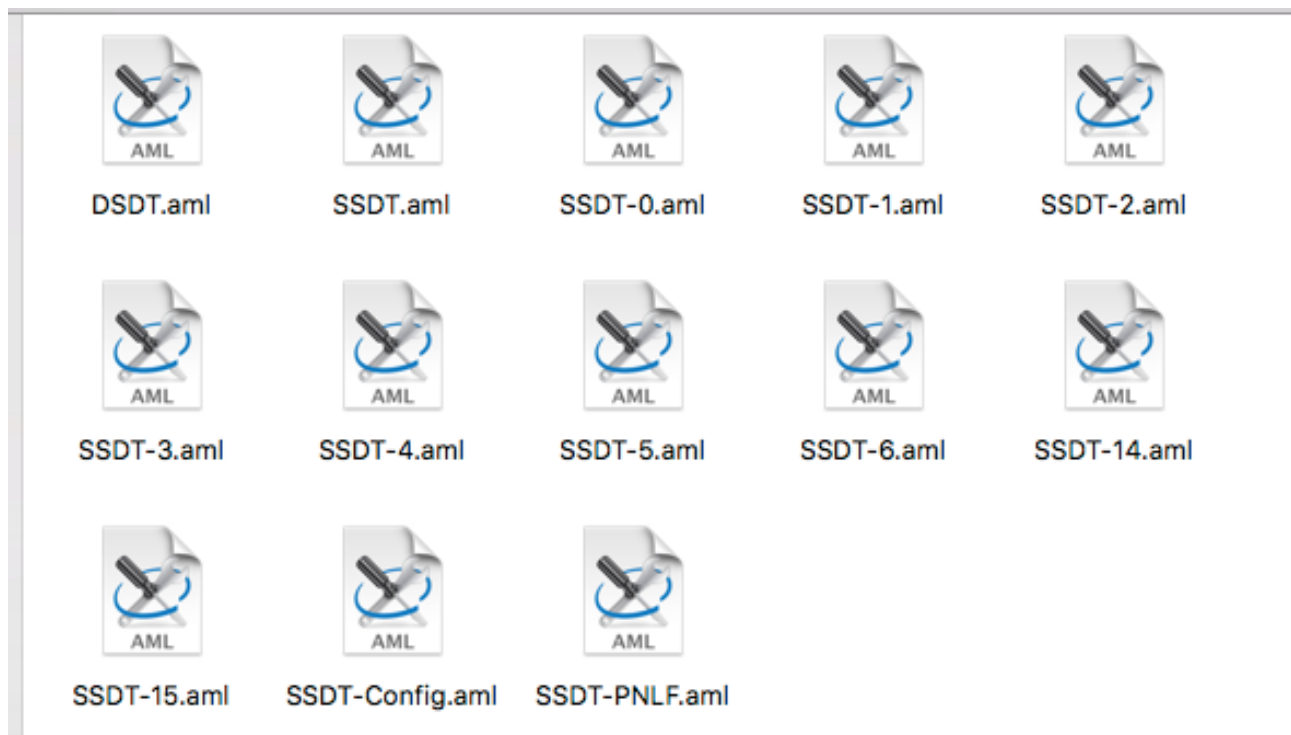


确认所有操作完成，且 Comlie没有错误即可保存 为DSDT.aml—笔记本专属的APCI文件，一机一份，同型号也并不能通用

经检查 SSDT-3.dsl 文件，内无 GFX0相关条目（PEGP），跳过不修改。
SSDT-5 及 SSDT-15需要修改 GFX0 为 IGPU



确保所有操作无误，讲SSDT-5 SSDT-15 另存为 SSDT-5.aml SSDT-15.aml,与 修改好的DSDT.aml一起放到 EFC/ACPI/patched 目录，其余提取到的 SSDT 文件，除外带 X 的SSDT 文件一放入。



其它文件

SSDT.aml—CPU变频,

SSDT-PNLF.aml—亮度调节,

SSDT-Config.aml—声卡、亮度快捷键调整相关,

SSDT-UIAC.aml—USB 插口注入（解决某些睡眠唤醒问题）

这些文件都需要多次实体测试才能正确制作，本篇不提且无计划翻译相关教程，原文如下：

SSDT—<https://www.tonymacx86.com/threads/guide-native-power-management-for-laptops.175801/>

<https://www.tonymacx86.com/threads/guide-usb-power-property-injection-for-sierra.222266/>

USB—<https://www.tonymacx86.com/threads/guide-usb-power-property-injection-for-sierra.222266/>

BackLight Control—<https://www.tonymacx86.com/threads/guide-laptop-backlight-control-using-applebacklightinjector-kext.218222/>