



Home / Browse / Clover EFI bootloader / Wiki



Clover EFI bootloader

Boot macOS, Windows, and Linux on Mac or PC with UEFI or BIOS firmware
Brought to you by: apianti, slice2009

This project can now be found here.

Fixing-DSDT



[Home](#) > [Fixing DSDT](#)

Table of Contents

- [Technical background](#)
- [DSDT mask \(Old Way\)](#)
 - [AddDTGP_0001 bit\(0\)](#)
 - [FixDarwin_0002 bit\(1\)](#)
 - [FixShutdown_0004 bit\(2\)](#)
 - [AddMCHC_0008 bit\(3\)](#)
 - [FixHPET_0010 bit\(4\)](#)
 - [FakeLPC_0020 bit\(5\)](#)
 - [FixIPIC_0040 bit\(6\)](#)
 - [FixSBUS_0080 bit\(7\)](#)
 - [FixDisplay_0100 bit\(8\)](#)
 - [FixIDE_0200 bit\(9\)](#)
 - [FixSATA_0400 bit\(10\)](#)
 - [FixFirewire_0800 bit\(11\)](#)
 - [FixUSB_1000 bit\(12\)](#)
 - [FixLAN_2000 bit\(13\)](#)
 - [FixAirport_4000 bit\(14\)](#)
 - [FixHDA_8000 bit\(15\)](#)
- [New Way patching](#)
 - [NewWay_80000000](#)
 - [FIX_RTC_20000](#)
 - [FIX_TMR_40000](#)
 - [AddIMEI_80000](#)
 - [FIX_INTELGFX_100000](#)
 - [FIX_WAK_200000](#)
 - [DeleteUnused_400000](#)
 - [FIX_ADP1_800000](#)
 - [AddPNLF_1000000](#)
 - [FIX_S3D_2000000](#)
 - [FIX_ACST_4000000](#)
 - [FixRegions_10000000](#)
- [Choosing the right mask](#)

Technical background

DSDT - Differentiated System Description Table - is the biggest and most complex ACPI table. The minimal length is 36 bytes, in reality it is about 20 kb or even more. This table describes devices and methods for accessing them. These methods can contain arithmetic and logical expressions representing a program written in a C-like programming language. Correcting this table means you need to have some sort of programming knowledge. Clover offers an option to automatically apply corrections, however it is important to understand that an artificial intelligence was not created yet and that the automatic method is far from being complete. It is better to do the corrections manually.

Why does it need to be fixed at all? DSDT patching was created with the intention to fix device *HPET* - High Precision Events Timer. The point is that OS X includes a kext named *AppleIntelCPUPowerManagement* for power management control (SpeedStep), which - by all means - needs interrupts *IRQ 0*

Get latest updates about
Open Source Projects,
Conferences and News.

Sign Up

No, Thank You

and `8`. Otherwise it will create a kernel panic. This next can either be removed or blocker, however you can alternatively correct the DSDT to ensure a normal behaviour of this next.

Nr. 1 : This is a necessity. Does Mac OS X really need HPET? Not really, but BIOS vendors tend to be slow and they just started writing the correct parameters. Usually a DSDT will still need to be corrected.

Nr. 2 : A DSDT contains certain dependencies on the operating system like `Windows 98`, `Windows 2001`, `Windows 2006` or `Linux`. Mac OS X uses identifier `Darwin`, which usually is missing. Even if it is not, it was created for FreeBSD. Mac OS X makes great use of the ACPI system and uses a DSDT to its maximum, as does Windows 2001; but not Linux, Windows 98 and not Windows 2006. It is always correct to mask the system as Windows 2001. Even if you find *Darwin* in your DSDT, mask it as Windows2001.

Many BIOS variants can use the variable OSYS = 0x07D2, but not 0x07D6, 0x07D9 or 0x2410 as written into a real Mac's DSDT.

Nr. 3 : The vendor of a motherboard's and thus the creator of a DSDT, cannot predict the devices you will be using (CPU, video card, etc.). They should be written into the DSDT however! Vice versa, devices like the internal speaker, floppy drive or parallel port should be excluded. Their drivers do not exist and are not even needed. Additionally it is often necessary to add or remove framebuffer/ports to devices like video cards or SATA controllers. The DSDT is written into the BIOS and is used by the system in AML binary code. It can be de-/compiled using IASL, which translates binary code into human readable DSL source code. A user will use this path to apply corrections: AML>DSL>edit>DSL>AML - this is the next point.

Nr. 4 : The last part is made impossible because of syntax and logical errors initially present in the OEM DSDT. You will need to correct them, too. Additionally, you can fix other mistakes, which prevent the PC from sleeping or waking up, or add new devices. (It is a bit strange, compiling/decompiling is not a strictly reversible operation and will change the table or even prevent further compiler operations. From my point of view the compiler is not bug-free. Non-conformance to the specification, however, should be considered as a warning, not as an error).

When you reached this step you can instruct Clover to use your modified DSDT by placing it into the directory EFI/CLOVER/OEM/xxx/ACPI/patched or - when the computer's name is not known yet - into EFI/CLOVER/ACPI/patched. Alternatively the OS can have its own DSDT in the root of the system partition.

Where can you obtain the initial DSDT that needs to be patched? There are different ways involving Windows, Linux or OS X. If you were able to start Clover somehow, you can enter its GUI and press `F4`. If Clover was installed on a FAT32 partition, then it will be able to save all ORM ACPI tables, including DSDT and FADT. The process can take a while, especially when saving many tables to a USB flash drive. This is especially useful when you have no access to other means of extracting the table set, for example with AIDA64. Also you can save the patched DSDT variant: enter Options in CloverGUI, change the DSDT mask, exit Options and press `F5`. A DSDT with the specified patches will be saved, which are represented in the file name, for example: *DSDT-F597.aml*. You can save multiple variants for comparison.

Now you can edit the DSDT and if you are not really comfortable doing this, Clover offers some automatic patches:

DSDT mask (Old Way)

```
! \ OBSOLETE SINCE CLOVER VERSION 4006
```

► Details

New Way patching

NewWay_8000000

Define that all bits will work new way.

FIX_RTC_20000

Exclude IRQ(0) from RTC device.

FIX_TMR_40000

Exclude IRQ(8) from TMR device. This is ancient DOS device and not needed in modern computers. Just wonder it pres

Get latest updates about
Open Source Projects,
Conferences and News.

AddIMEI_80000

Sign Up

This device is used for IntelHDxxx graphics. Adding them if very desirable operation. This bit also needed for use FakeID->IMEI. Don't forget for Core 2

systems.

FIX_INTELGFX_100000

New way IntelGFX device will not be patched without this bit.

FIX_WAK_200000

adding Return(Package(0)) into method _WAK if absent. This patch is for warning elimination. I don't know about working influence.

DeleteUnused_400000

There are not used devices like Floppy drive, LPT port and others that will be good to delete from DSDT.

FIX_ADP1_800000

Rename ACO device to ADP1 device.

AddPNLF_1000000

Adding device PNLF is very useful: only with it you may have brightness control. This patch is also influence on good Sleep/Wake of the system.

FIX_S3D_2000000

Also resolving some Sleep/Wake problems by correcting _S3D methods.

FIX_ACST_4000000

Name ACST have different use for Apple and for ASUS. For ASUS it is AC adapter state. For Apple it is a replacement for _CST, c-states table. To not conflict it is needed to rename such names to something else.

FixRegions_10000000

Address of some regions in DSDT depends on many factors and may change time to time

```
OperationRegion (GNVS, SystemMemory, 0xDE6A4E18, 0x01CD)
```

The presence of floating regions make impossible to use custom DSDT because this region may be shifted and will not correspond to current state. This patch is intended to find all such regions in BIOS and correct them in custom DSDT. So now you can produce your custom DSDT with wrong regions and set this patch.

Choosing the right mask

How can you choose the necessary patches and how do you know which ones are harmless or dangerous? The computer will not be harmed either way. All the changes are stored memory only and will be removed after rebooting.

You can try setting different combinations in CloverGUI and save them by pressing **F5** in the *Options* menu.

To make sure the currently patched DSDT is not creating a conflict, you can change the DSDT name in the menu - *DSDT name: NO.aml*. This file will not be found, Clover will extract the original OEM DSDT from BIOS and apply fixes set in the DSDT mask section. In case of OS booting successfully, your previously set (working) values will be used.

0xFFFF enables all fixes and if the OS loads successfully this way, you will know that our efforts were not for nothing. If you already realised that some fixes are not needed for your system (for example WiFi), they can even make things worse.

Most systems, which are neither too old, nor too new, a mask of **0xA7D7** has been proven to produce a well working OS.

[Sign Up](#)

New way you may make patched DSDT once with full mask. Then correct patched DSDT manually. Then use this manually patched DSDT,aml loading

No, Thank you

but set FixRegions_10000000 The mask will be 0x90000000 = NewWay_80000000 + FixRegions_10000000

[< Design](#)

[Back to top ^](#)

[Native Speedstep >](#)

**Get latest updates about
Open Source Projects,
Conferences and News.**

[Sign Up](#)

[No, Thank you](#)