

Erledigt

Ozmosis installiert => ungepatchte DSDT extrahieren

Beitrag von „Stellaris“ vom 20. November 2016, 20:06

Hallo zusammen,

mal wieder eine Anfängerfrage zu der ich nichts Passendes gefunden habe:

Ich habe mir mittlerweile zusätzlich zu Ozmosis Clover installiert und damit per F4 meine DSDT etc. extrahiert.

Meine Frage ist jetzt, ob Clover bei installiertem Ozmosis die von Ozmosis gepatchten DSDTs extrahiert oder tatsächlich die OEM-DSDTs. Ich habe extra Ozmosis mit Escape deaktiviert, bin mir aber immer noch nicht 100% sicher.

Dank und Grüße!

Beitrag von „al6042“ vom 20. November 2016, 20:11

Wenn in dein Ozmosis ROM bereits eine gepatchte DSDT eingebaut wurde, kann Clover auch nur diese wieder extrahieren.

Wenn du das absolute Original haben möchtest, kannst du das aus einem heruntergeladenen Original ROM entweder mit dem Terminal-Tool "OZMTool" oder der App "DSDT2Bios" direkt extrahieren.

Beitrag von „griven“ vom 20. November 2016, 20:16

Wenn Du das OZ von uns hast dann ist die DSDT nur Fehlerbereinigt sprich die kannst Du mit F4 über Clover extrahieren und verwenden. Da sich die Konfigurationen individuell sehr stark

unterscheiden verzichten wir bei den OZ Mods darauf die DSDT umfangreich zu bearbeiten. In der Regel trägt sie nur den CleanCompile Patch so das gleich damit weiter gearbeitet werden kann.

Beitrag von „Stellaris“ vom 20. November 2016, 20:24

Super, danke für die Info!

[@griven](#), könntest du mir kurz erklären was der clean-compiler-Patch macht?

Und warum kommt dann Windows mit einer für OSX bearbeiteten DSDT klar?

edit: Anders rum gefragt: Was könnte man den über "umfangreichere" Patches, die in Ozmosis nicht gemacht werden, denn erreichen?

Grüße

Beitrag von „griven“ vom 20. November 2016, 20:32

Der CleanCompile Patch beseitigt typische Fehler die verhindern das eine DSDT kompiliert werden kann. Eine DSDT ist ein notwendiger und integraler Bestandteil der Firmware und ist daher in jeder Firmware enthalten. Die Hersteller von PC Mainboards arbeiten hier aber leider meist nicht sehr sauber sprich halten sich nicht sonderlich gut an den ACPI Spezifikationen die regeln wie eine solche DSDT auszusehen hat um den Standard gerecht zu werden. Es gibt neben einer gewissen Syntax die einzuhalten ist auch einige andere Konventionen die spezifiziert sind und die der DSL Compiler anmeckert wenn sie nicht eingehalten werden. Der CleanCompile Patch kümmert sich um genau diese Konventionen sprich er verbessert gern gemachte Syntax Fehler und trägt dafür Sorge das die Namen der Devices und Methoden den Konventionen entsprechen. Wenn man so will ist es ein Lektor der dafür Sorge trägt all zu offensichtliche Fehler zu beseitigen.

Das Windows auch mit einer ziemlich umfangreich an die Bedürfnisse von OS-X angepassten DSDT zu recht kommt liegt in der Hauptsache daran das Windows die DSDT weitestgehend ignoriert und sich lieber darauf verlässt die Geräte über den PCI bzw. LPC Bus zu identifizieren

und dann die Methoden und Treiber zur Ansteuerung der Komponenten lieber aus seinen eigenen Quellen hinzu lädt als sich auf den ACPI Standard zu verlassen.

Beitrag von „al6042“ vom 20. November 2016, 20:34

Die DSDT.aml ist eine Konfigurations-Datei die in jedem BIOS eines Mainboards vorhanden ist und die Geräte beschreibt, die auf dem Mainboard verbaut sind.

Das Format .aml bezieht sich auf die tatsächliche Datei in Maschinensprache kompiliert, während die .dsl-Variante eine Text-basierende Version der gleichen Datei ist, aber nicht direkt genutzt werden kann.

Da Windows und Linux eine eigene Hardwareerkennung mitbringen, ist die DSDT der PC-Mainboards sehr rudimentär gehalten, während in Apple-Geräten alles haarklein beschrieben wird.

Eine fertige DSDT muss in .aml-Form genutzt werden und ist nicht mit einem Kext vergleichbar, da sie bereits vor dem Starten des Betriebssystems zum Einsatz kommt.

Die SSDT.aml ist gleich aufgebaut, hält aber meistens nur CPU-spezifische Informationen oder zusätzliche "Scope"-Einträge für bereits definierte Geräte.

Die Dateien werden nicht ausgeführt, sondern aus-, bzw. eingelesen.

Aber nicht von einem Programm unter Windows, Linux oder OSX, sondern vom jeweiligen Kernel des Betriebssystems.

Dieser holt sich die Daten aus dem BIOS.

Beitrag von „Stellaris“ vom 20. November 2016, 21:02

Ahja, das bringt schon etwas Licht in's Dunkel 👍 .

Dann wäre noch die Frage wo man den cleanCompile-Patch herbekommt, da hat Google nichts Brauchbares ausgespuckt.

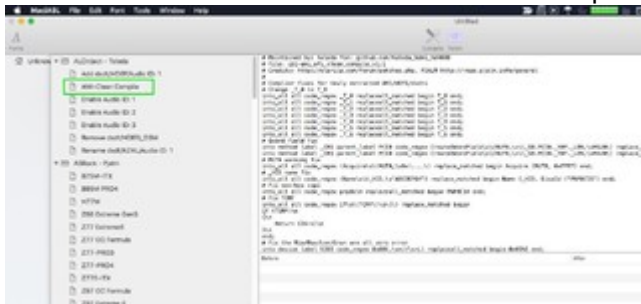
und noch eine ;):

Wenn das System ohne Probleme läuft, was ja bei Ozmosis der Fall ist, obwohl die DSDT quasi nicht gepatcht wurde, was können dann umfangreichere Patches noch bringen?

Grüße

Beitrag von „al6042“ vom 20. November 2016, 21:06

Im [Macias](#), bzw. der dort befindlichen .plist sind die Quellen der DSDT-Patche hinterlegt. Dort findet man dann auch den "Clean Compile" Patch:



Zusätzlich sollte man sich aber auch folgenden FAQ-Beitrag zu Gemüte führen:

<https://www.hackintosh-forum.d...gepatchten-DSDT-Desktops/>

Der arbeitet nicht mal mit dem "Clean Compile", da dort direkt auf die Bedürfnisse der einzelnen Board-Typen eingegangen wird.

Beitrag von „Stellaris“ vom 20. November 2016, 21:34

Achso, in MaciASL hab ich noch nicht geschaut.

Die Anleitung habe ich mir schon durchgelesen :). Allerdings beantwortet er immer noch nicht die Frage, warum Ozmosis mit einer ungepatchten DSDT auskommt. Ok, man braucht noch

einen Kext für LAN und Sound (vermutlich weil diese Chips in keinem Mac jemals verbaut wurden und man daher keinem original Kext beibringen kann damit umzugehen?). Aber so Sachen wie RTC-Fix oder Shutdown-Fix muss doch auch Ozmosis irgendwie handeln, oder?

Grüße

Beitrag von „al6042“ vom 20. November 2016, 21:39

Ozmosis macht da gar nichts, da dies nur der Bootloader ist...

Entweder hast du Glück und der Patch wird nicht vermisst oder auch nicht und er muss in eine DSDT eingebaut werden.

Das entscheidet sich aber von Fall zu Fall.

Beitrag von „griven“ vom 23. November 2016, 23:43

[@Stellaris](#) und das tut Ozmosis deshalb weil die fleissigen Elfen die hier in der OZ Küche die ROMS backen genau diese Patches schon in der DSDT platzieren 😄

Mal ein kleiner Einblick wie so ein ROM entsteht so von Anfang bis Ende...

1. Original Bios von der Herstellerseite laden
2. DSDT aus dem original Bios mittels ozmtool extrahieren (ozmtool --dsdtextextract)
3. DSDT mittels MacIASL von gängigen Fehlern befreien und je nach Board noch die LPC, SMBUS, SATA Fixes einpatchen
4. DSDT kompilieren und speichern
5. Alles mit der Hilfe des ozmtools wieder in einen ROM packen (ozmtool --ozmcreate -d dsdt/DSDT.aml)
6. Rom hochladen und dann über die viele Arbeit jammern 😄

Alles kein Hexenwerk sondern einfach ein zusätzlicher Service der dem Enduser das Leben erleichtert 😄