

Erledigt

Hilfe bei Erstellung DSDT/SSDT gewünscht

Beitrag von „Amiga“ vom 12. August 2018, 17:38

Hallo

Ich habe Heute erfolglos versucht eine gepatchte DSDT und SSDT für mein System zu erstellen:

Probleme:

1. Ich kann von den 3 angezeigten DSDT Compile Fehlern im MaciASL nur den ersten Fehler lösen. Die zwei weiteren bekomme ich nicht in den Griff.
2. Die für das Patchen erforderliche Repo in MaciASL bekomme ich nicht angezeigt. Die letzte plist zur Rehab Repo steht in Preferences und der Link ist im Programm hinterlegt. Geht trotzdem nicht?



Kann mir jemand bei der Erstellung helfen?

Habe die Dateien aus Clover angehängt.

Lg

Rolf

Beitrag von „al6042“ vom 12. August 2018, 17:44

Erstaunlicherweise habe ich noch eine DSDT für das Z97X-UD5H BIOS Version F10 hier herumliegen.

Aber wofür willst du die ganzen SSDTs ändern?

Es reicht die gepatchte DSDT und ggf. eine speziell erstellte SSDT für deine CPU.

Die sollte dann aber über Piker Alpha ssdtprgen.sh erstellt werden.

Beitrag von „Amiga“ vom 12. August 2018, 18:02

Hallo Al6042

Vielen Dank für die DSDT.

Bin erstaunt was bei Dir alles so auf dem Schreibtisch zu finden ist.

Geht bei dir die Repo von MaciASL?

Das Script von Pikeralpha kenne ich durch Zufall. Würde eine SSDT einen Vorteil für mein System bringen? Ich kann nicht abschätzen ob der Aufwand lohnt.

Gruß von Quedlinburg nach Mannheim. Hab ja auch mal in Mainz gewohnt.

Rolf

Beitrag von „rubenszy“ vom 12. August 2018, 18:07

Würde es aber nur in Verbindung mit acpi von CPUFriend.

SSDT erstellen und hier hochladen, dazu noch SMBios angeben.

Beitrag von „Amiga“ vom 12. August 2018, 18:31

Hallo rubenszy

Da ich davon noch nie etwas gelesen habe folgende Fragen:

Also CPUfriend ist ein kext und kommt in den Clover kext Ordner. Richtig?

Aber was soll dann nach dem Systemneustart passieren?

Erhelle doch etwas Dunkel Deutschland. Wohnst ja nicht weit von mir.

Lg

Rolf

Beitrag von „al6042“ vom 12. August 2018, 18:33

Bei mir funzt Maciasl 1.31 aus folgendem Link -> [MaciASL](#)
Aber mit folgender .plist -> [MaciASL](#)

Falls du Maciasl schon geöffnet hattest, starte erst den Rechner neu und platziere die .plist in
~/Library/Preferences, bevor du Maciasl wieder startest.

Beitrag von „rubenszy“ vom 12. August 2018, 18:52

erstelle erst mal eine SSDT und lade die mal hoch.
[https://www.hackintosh-forum.d...p/FAQ/197-SSDT-Speedstep/](https://www.hackintosh-forum.de...p/FAQ/197-SSDT-Speedstep/)
am besten deine config auch gleich.

Beitrag von „kuckkuck“ vom 12. August 2018, 20:34

[Zitat von Amiga](#)

Aber was soll dann nach dem Systemneustart passieren?

Die SSDT + Kext Kombi sorgt für korrekt laufendes CPU PowerManagement und das kann wiederum einen schönen Temposchub mit sich bringen und sich ebenfalls positiv auf PM abhängige Dienste wie Sleep auswirken. Ist also definitiv lohnend 😊

Beitrag von „Amiga“ vom 13. August 2018, 10:09

Hallo rubenszy

Ich habe wie gewünscht die ssdt.aml erstellt und zusammen mit der clover .plist angehängt.

Der CPUFriend.kext liegt auch schon im Downloadordner.

LG Rolf

Beitrag von „rubenszy“ vom 13. August 2018, 11:15

Viel Spaß mit der SSDT.
Du brauchst kein CPUFriend.kext, die SSDT vereint beides in einem.

Beitrag von „kuckkuck“ vom 13. August 2018, 13:48

[Zitat von rubenszy](#)

Du brauchst kein CPUFriend.kext

Ich bin gerade ein wenig verwirrt, du hast doch cf-frequency-data eingebaut und das braucht doch die CF Kext? 😞

Oder steh ich gerade auf dem Schlauch? 😄

Beitrag von „rubenszy“ vom 13. August 2018, 14:11

CPUFriend.kext wird dadurch hinfällig, den braucht man nur wenn du einen CPUFriendDataProvider.kext erstellst.
Daher macht es mehr Sinn eine Kombination aus beiden ssdtPRGen und CPUFriendDataProvider in der SSDT.

Beitrag von „kuckkuck“ vom 13. August 2018, 14:31

Ich bin verwirrt, cf-frequency-data ist doch kein macOS bekanntes Property für CPU0, dass von Apples Treibern verarbeitet wird? Das property (durch die SSDT injected) kann doch allein garnichts machen... 😞

cf-frequency-data ist doch lediglich ein Property, dass CPUFriend erkennt und einließt und daraufhin passend injected. Innerhalb der Daten dieses (cf-frequency-data)Strings lässt sich ja eine XML Datei finden. Das heißt wir haben hier properties innerhalb eines property, macOS kennt sowas soweit ich weiß garnicht...

Ich denke mal cf-frequency-data steht eigentlich für CPUFriend-X86Vektor-Daten und setzt die Nutzung der CPUFriend.kext voraus um etwas zu bewirken.

Beitrag von „rubenszy“ vom 13. August 2018, 15:14

Hatte ich mal so gelesen, das man durch das einfügen der Frequency Vektoren (von der erstellten ssdt_data) in die SSDT, CPUFriend.kext hinfällig wird, da macOS mit den eingetragenen Daten was anfangen kann.

Ich lasse mich auch gern eines besseren belehren, da man nur wenig über CPUFriend findet was einem wirklich weiter hilft.

Beitrag von „kuckkuck“ vom 13. August 2018, 15:40

[Zitat von rubenszy](#)

Ich lasse mich auch gern eines besseren belehren, da man nur wenig über CPUFriend findet was einem wirklich weiter hilft.

Da gehts mir genauso, deshalb frag ich nach...

Also sowohl eine CF-SSDT als auch die DataProvider Kext beinhaltet ein cf-frequency-data Property. Wenn man dieses mal in ASCII übersetzt sieht es in etwa so aus:

cf-data als XML

Entspricht also größtenteils der Plist aus dem X86PlatformPlugin.

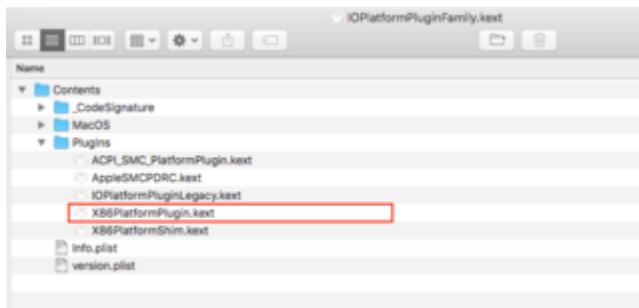
Im CPUFriend.cpp (Xcode) findet man folgendes Snippet:

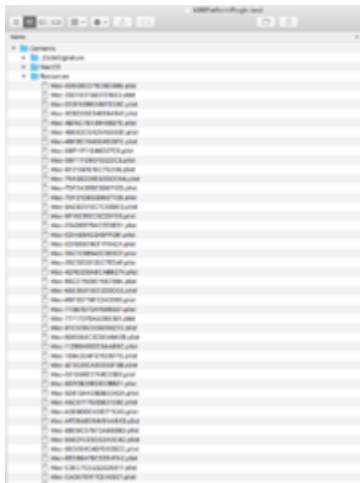
Kext-Code

Auch findet man in originalen Mac Dumps kein cf-frequency-data Property für CPU0 im IOReg. Deswegen hätte ich jetzt gedacht, dass egal ob CPUFriendProvider.kext oder SSDT, CPUFriend.kext an sich notwendig ist...

Beitrag von „rubenszy“ vom 13. August 2018, 16:21

Kurz gesagt ist das nur eine Plist, das die Plist von hier





mit werten füttert, die in der Originalen nicht vorhanden sind, also andere FrequencyVectors, ähnlich das selbe was <https://github.com/Piker-Alpha/freqVectorsEdit.sh> macht.

Beitrag von „kuckkuck“ vom 13. August 2018, 16:36

Der einzige Sinn von CPUFriend ist dementsprechend andere FrequencyVectors als die im SMBios hinterlegten zu injecten.

Dementsprechend macht die Nutzung der Kext auch nur Sinn, wenn die Daten aus der genutzten X86-Plist auch verändert werden. FreqVectorsEdit ändert soweit ich weiß lediglich die Zuordnung, sprich lassen sich für zB ein iMac17.1 auch die FrequencyVectors eines anderen Macs (andere Board-ID) injecten. Die Vectors an sich werden dabei nicht verändert (manuelle Anpassung der Plist ist mit dem Tool jedoch möglich).

Wenn du eine SSDT mit CPUFriend baust, wie veränderst du dann die FrequencyVectors?

Beitrag von „rubenszy“ vom 13. August 2018, 16:59

Gar nicht, was ich nicht verstehe ist, was soll das bringen die 1 zu1 Kopie plist mit cpufriend

noch mal zu laden wenn sie schon vom SMBios aus geladen wird.

Beitrag von „Noir0SX“ vom 13. August 2018, 17:10

[Zitat von rubenszy](#)

.....da man nur wenig über CPUFriend findet was einem wirklich weiter hilft.

Passt zwar nicht ganz zum Thema... oder doch. Hat aber nichts mit Euch zweien zutun.

Ich verstehe dann auch User nicht, wenn man die Infos CPUFriend oder Lilu & Plugins zusammenträgt, die dann per PN/Mail noch meckern.

So weiter im Text, musste ich mal los werden 😊

Beitrag von „rubenszy“ vom 13. August 2018, 17:19

Das was du machst ist schon OK, nur stellen wir gerade mal fest, das dieser kext eigentlich nicht wirklich was bringt, wenn man nicht die plist aus dem CPUFriendDataProvider.kext mit anderen FrequencyVectors ändert.

Das einzige was Sinn macht mit dem kext ist, wenn du eine CPU hast die einem iMac18.3 gleicht, du aber das SMBios von einem iMac 14.2 eingegeben hast aber die FrequencyVectors von den iMac18.3 nutzen möchtest.

Beitrag von „kuckkuck“ vom 13. August 2018, 17:23

[Zitat von rubenszy](#)

was soll das bringen die 1 zu1 Kopie plist mit cpufriend noch mal zu laden wenn sie schon vom SMBios aus geladen wird.

Ja nichts 😊

Zitat von Github

file should be a complete plist from
/System/Library/Extensions/IOPlatformPluginFamily.kext/Contents/PlugIns/X86PlatformPlugin.kext/Contents
with certain modifications (Otherwise why do you need it?) instead of
something like a raw FrequencyVectorsentry.

[@NoirOSX](#) Bin ein großer Fan von deinem Thread, was ich ja auch schonmal per PM zu verstehen gegeben habe, aber hier geht es um noch mehr als die allgemeine Nutzung der Kext. Das ganze Thema ist ein wenig kompliziert, ich erinnere mich auch an einen 6 seitigen Thread zum Thema PM, den es letztens erst gab...

Zitat von rubenszy

Das einzige was Sinn macht mit dem kext ist, wenn du eine CPU hast die einem iMac18.3 gleicht, du aber das SMBios von einem iMac 14.2 eingegeben hast aber die FrequencyVectors von den iMac18.3 nutzen möchtest.

Wann kommt das schon vor. Die Hardware wird in diesem Fall ziemlich sicher eher dem 18.3 gleichen (Platform) und die paar Ähnlichkeiten, die das Gerät dann noch zum 14.2 hat, sollte man dann manuell anpassen und nicht den 14.2 an den 18.1 anpassen...

Das einzige was IMO Sinn machen würde, wäre alle FrequencyVectors, manuell an die eigene CPU anzupassen. Ich habe das noch nie gemacht, da müsste ich mir mal anschauen wie die FreqVectors im X86 Plugin überhaupt definiert werden und das dann durch meine Werte (die man auch erstmal rausfinden muss) ersetzen.

Beitrag von „Amiga“ vom 14. August 2018, 10:23

Hallo an Alle

Stand der Dinge:

[@al6042](#)

DSDT in Clover eingefügt und läuft perfekt.

MaciASL und Link zu Repo laufen jetzt auch.

[@rubenszy](#)

SSDT ist ohne cpufriend Kext in Clover eingefügt und scheint auch zu laufen.

[@rubenszy](#), [@kuckkuck](#), [@NoirOSX](#)

Zum Exkurs kann ich leider noch nichts beitragen.....

Das System läuft gefühlt schneller als vorher.

Das Update auf Mojave Public Beta 6 ist gerade ohne Probleme durchgelaufen.

Jetzt noch ein Backup und die Beta 6 erkunden.

Ich bin sehr zufrieden und meine Forumshelfer können es auch sein!

LG Rolf