

Erledigt DSDT Sammelthread (Hilfe und Diskussion)

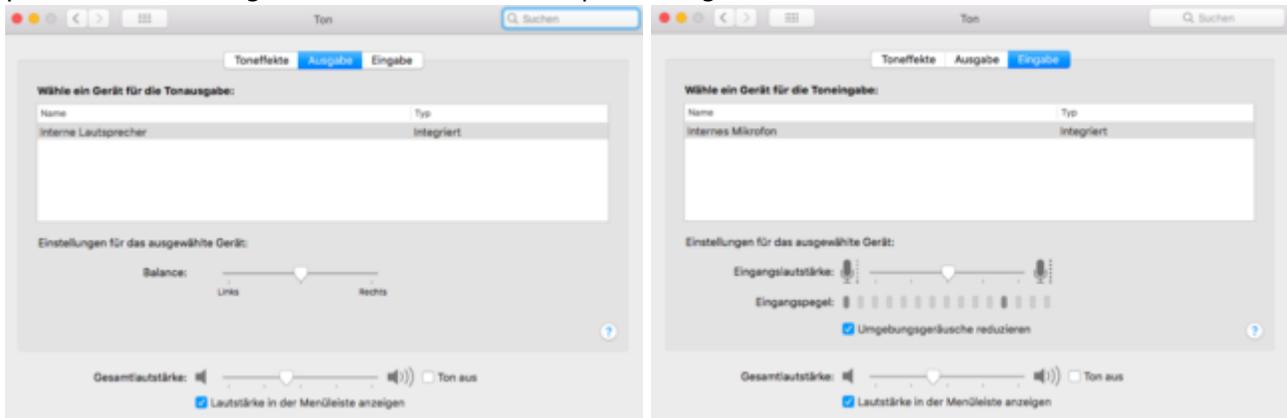
Beitrag von „derHackfan“ vom 3. Juli 2017, 17:26

Hallo Community,

früher oder später stolpert jeder Hackintosher ein mal über das Thema DSDT und die damit verbundenen Möglichkeiten bezogen auf seine Hardware und sein OS X.

In meinem Fall um mal ganz konkret zu werden, ist der Clover Bootloader nicht in der Lage einen Ton aus meinem Qosmio Hackbook zu entlocken, eigentlich sollte das (doch) mit einem FixHDA und Audio/Inject Eintrag kein Problem sein.

Nehme ich aber nun die Mühe auf mich und erstelle eine gepatchte DSDT (clean ohne Fehler, patches mit warnings) dann sieht die Sache plötzlich ganz anders aus.



Ich möchte einfach jeden aus der Community auffordern seine Erfahrungen und Problem hier zu posten um gemeinsam daraus zu lernen, **jeder Anfänger, Fortgeschrittener, Experte oder Halbgott** darf hier simples und spezielles einwerfen und diskutieren.

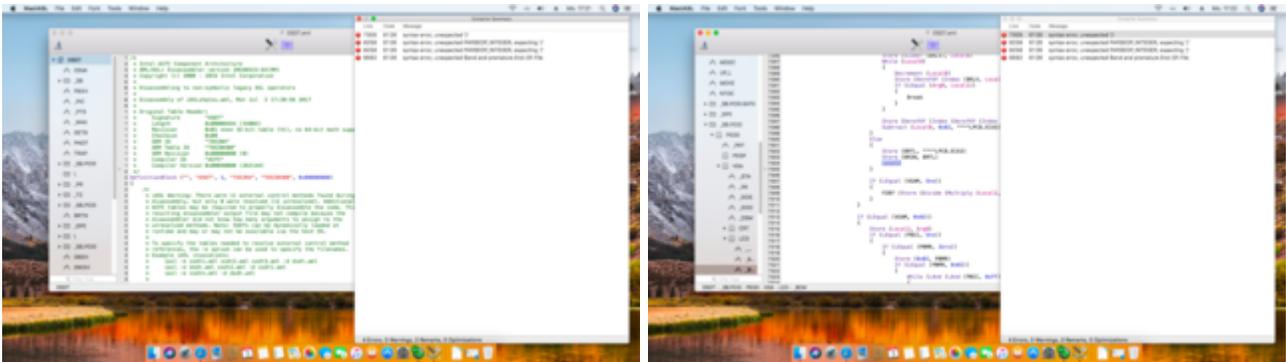


Ich wünsche viel Erfolg mit dem Thema DSDT und weiter Spaß im Forum.

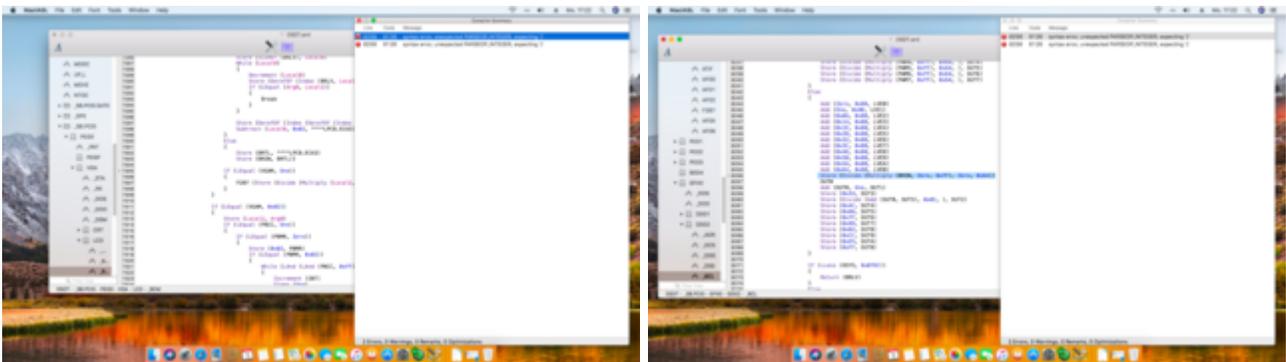
Gruß derHackfan

Beitrag von „derHackfan“ vom 3. Juli 2017, 17:28

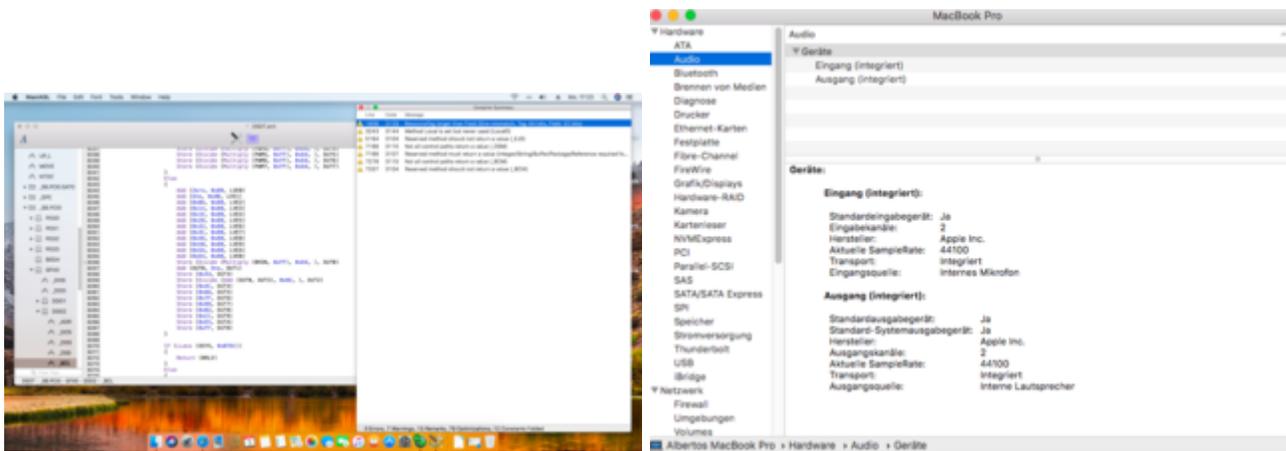
Hier mal ein Beispiel dafür welche Stolpertseine sich bei mir aufgetan haben bis ich eine fehlerfreie DSDT hast.



Problem:



Problem:



Lösung:

An diesem Beispiel möchte ich möglichst viele Leute dazu auffordern sich dem Thema DSDT zu widmen, postet eure Probleme und Hindernisse auf dem Weg zu fehlerfreien und fertig gepatchten DSDT in diesem Thread.



Ich wünsche viel Erfolg mit dem Thema DSDT und weiter Spaß im Forum.

Gruß derHackfan

Beitrag von „derHackfan“ vom 3. Juli 2017, 17:53

Da ich mich zu den **Anfängern** zähle ist meine Vorgehensweise auch eher simple und nach Anleitung.

Ich wünsche viel Erfolg mit dem Thema DSDT und weiter Spaß im Forum.

Gruß derHackfan

Beitrag von „derHackfan“ vom 3. Juli 2017, 18:28

- Platzhalter -

Code

1. // Eine Zeile ausgrünen

Code

1. /* Einen Bereich ausgrünen */
-

Beitrag von „Harper Lewis“ vom 3. Juli 2017, 19:42

Spannendes Thema! Die Zeilen, die einen Syntaxerror verursacht haben, hast du entfernt, oder?

Beitrag von „Noir0SX“ vom 3. Juli 2017, 21:33

[@Harper Lewis](#) nicht immer komplett. Bin da auch erstmal Bild vor zurück und wieder vor 😊

Beitrag von „Doctor Plagiat“ vom 3. Juli 2017, 21:56

Über die anfängliche Fehler in der rohen DSDT kann ich keine neuen Erkenntnisse beibringen, da bei meinen DSDTs (Desktop und Hackbook) nur die üblichen Fehler auftraten, die schnell zu beseitigen waren (siehe Wiki "ACPI -> [Grundlegende Schritte zum Erstellen einer gepatchten DSDT - Desktops](#)).

Bei meinem Hackbook hatte ich aber ganz andere Probleme. Für ein Notebook fand ich den Patch "Rename GFX0 to IGPU" ganz wichtig. Immer wenn ich diese DSDT dann benutzte, wurde der SATA-Controller und damit die SSD nicht mehr erkannt. Klinkt total zusammenhanglos, aber es war nun mal so. Wenn man keine Ahnung hat versteht man auch manche Zusammenhänge nicht. Logisch oder?

Durch mehrere Vergleiche unterschiedlicher DSDTs änderte ich dann nochmal das Device MCHC, welches vorher so aussah.

Code

1. Device (MCHC)
2. {
3. Name (_ADR, Zero) // _ADR: Address
4. }

Nachher:

Code

1. Device (MCHC)
2. {
3. Name (_ADR, Zero) // _ADR: Address
4. }
- 5.
- 6.
7. Device (GFX0)
8. {
9. Name (_ADR, 0x00020001) // _ADR: Address
10. }
- 11.
- 12.

13. Device (B0D3)
14. {
15. Name (_ADR, 0x00030001) // _ADR: Address
16. }

Alles anzeigen

Seit dem klappt es auch mit dem SATA-Controller, was vielleicht daran liegt, dass wieder ein Device (GFX0) vorhanden ist.

Falls jetzt ein DSDT-Versteher hier fragt, ob ich was falsches geraucht habe. Nein, ich bin Nichtraucher und verstehe weder warum es vorher nicht, noch warum es danach funktioniert. Aber es freut mich schon, dass es funktioniert.

Beitrag von „derHackfan“ vom 3. Juli 2017, 22:58

[@Harper Lewis @BlackOSX](#) das ganze bezieht sich natürlich auf meine rohe DSDT und mein Toshiba X870-11D.

Den [Local2](#) aus dem zweiten Bild habe ich einfach gelöscht.

Die beiden [ZERO](#) in Line 8256 habe ich auch gelöscht und den einzelnen [DUTO](#) in die Klammer gerückt.

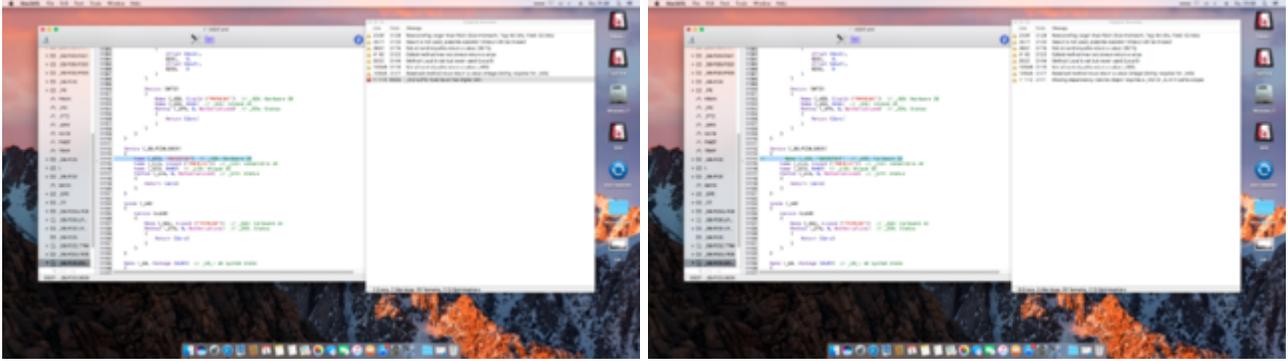
Das Ergebnis ist dann 0 Errors, 7 Warnings, 13 Remarks, 79 Optimizations und ich konnte mit dem eigentlichen patchen beginnen.

Mit diesem Zwischenergebnis sollte jeder unbedingt seine eigenen Erfahrungen beim patchen der DSDT sammeln, man kann sich dumm und duseelig lesen oder einfach mal anfangen, Learning by Doing ist meine Strategie.

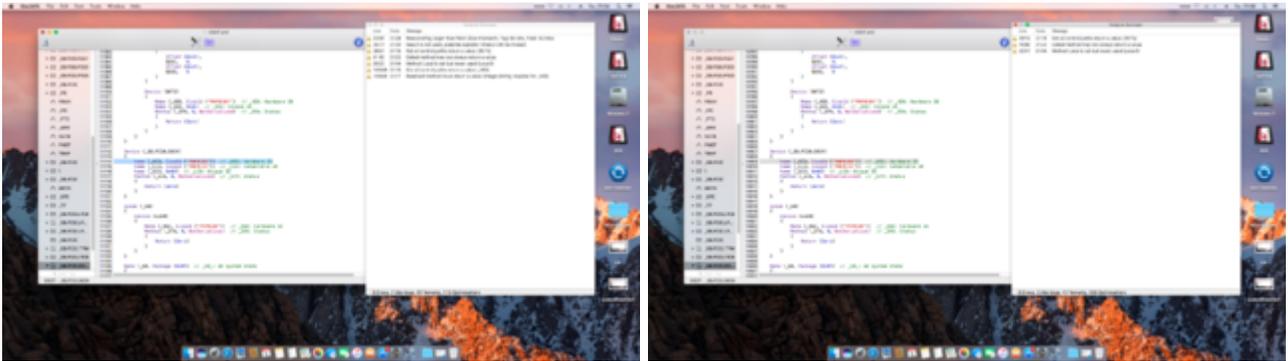
Beitrag von „derHackfan“ vom 6. Juli 2017, 22:31

Hier noch ein weiteres Beispiel aus der Praxis, der [@cinemascope](#) hat heute im Thread eine

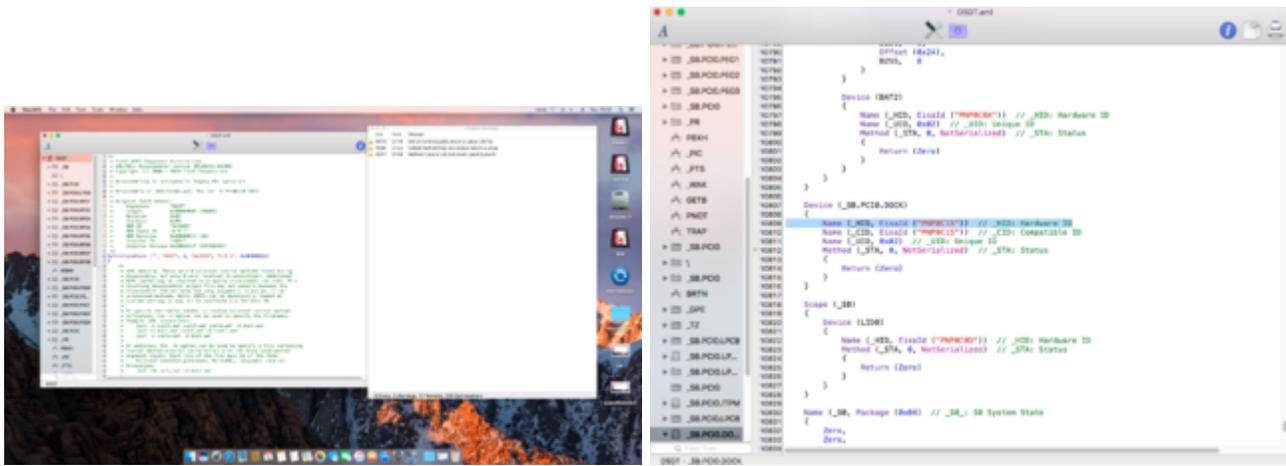
rohe DSDT von seinem Gigabyte Z77 DS3H hochgeladen.



Problem: 1 Errors in Line 11114 und anschließend auskommentiert.



Problem: Bereinigt auf meine Art und anschließend direkt den Gigabyte - Pjalm Z77 DS3H Patch angewendet.



Problem: Man kann auch ohne Fehlerbereinigung den Pjalm Patch anwenden und findet anschließend das Ergebnis in Line 10809.

An diesem Beispiel kann man sehr schön erkennen dass das Patchen einer rohen DSDT gar nicht so schwer ist, richtig interessant wird es erst wenn man keinen fertigen Pjalm Patch für sein Mainboard findet.



Ich wünsche viel Erfolg mit dem Thema DSDT und weiter Spaß im Forum.

Gruß derHackfan

Beitrag von „Doctor Plagiat“ vom 7. Juli 2017, 09:50

[Zitat von derHackfan](#)

dass das Patchen einer rohen DSDT gar nicht so schwer ist

Insofern gebe ich dir recht, solange man weiß welche Patches denn überhaupt angewendet werden müssen. Was auch voraussetzt, dass man weiß für welches Wehwehchen der jeweilige Patch Linderung verspricht. Ich glaube, dass das die am höchsten liegende Latte beim DSDT-Patching ist, die übersprungen werden muss. 😞

Und wenn, wie du erwähnt hast, gar kein fertiger Patch vorhanden ist, guck ich ich ganz doof in die Röhre oder was auch immer. 😭

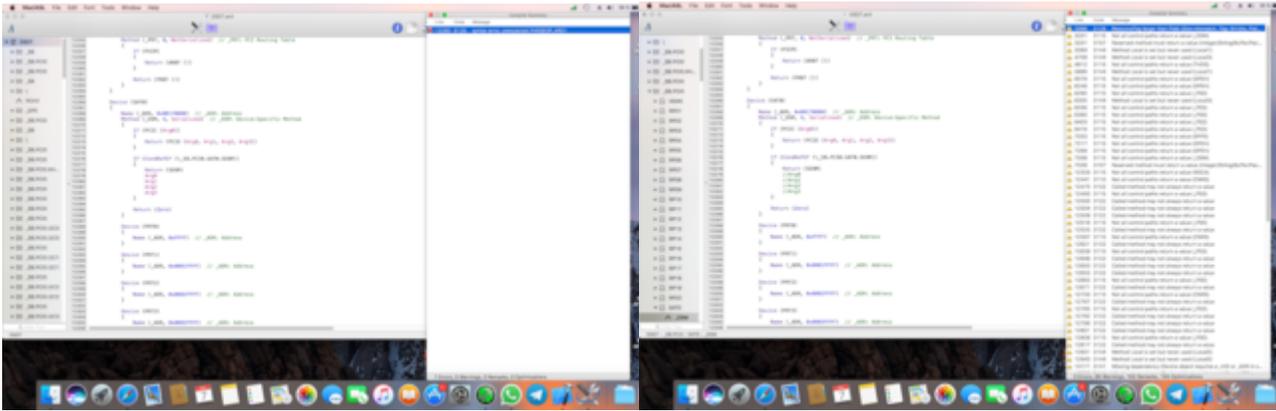
PS. Meine eigene DSDTs kriege ich schon ganz gut hin (so zum lernen, denn ich habe ja eine fertige), aber mit einer fremden, die vielleicht noch viel anders aufgebaut ist, würde ich wahrscheinlich kläglich versagen. Vor allem scheitere ich immer noch am Auslesen eines IOReg-Dumps um das Ergebnis in die DSDT zu übertragen.

Beitrag von „derHackfan“ vom 7. Juli 2017, 10:25

Da brauche ich kein Zitat für und bin (auch) in jedem Punkt ganz bei dir, aber das ist ja auch der Sinn von diesem Thread dass man seine Erfahrungen teilt und gemeinsam Schritt für Schritt vorankommt. 👍

Beitrag von „Noir0SX“ vom 7. Juli 2017, 10:25

Beispiel um eine bereinigte DSDT zu bekommen, welche oft vorkommt



Lösung

"Arg"-Einträge entfernen bzw. Auskommentieren

Beitrag von „Doctor Plagiat“ vom 7. Juli 2017, 11:24

Das ist, glaube ich, der Erstfehler bei fast allen DSDTs. Du hast sie auskommentiert 👍 ich lösche sie immer. Ergebnis ist in jedem Fall dasselbe.

Beitrag von „Thogg Niatiz“ vom 7. Juli 2017, 11:31

Vielleicht ist es hilfreich immer mal dazu zu schreiben, warum die jeweilige Änderung notwendig ist.

Zu den ArgN Variablen nach dem Return:

Die `_DSM` Methode (im Prinzip eine Funktion) nimmt 4 Parameter entgegen ("`Method(_DSM, 4, Serialized)`"). Die Werte dieser Parameter lassen sich innerhalb der Methode der Reihenfolge nach als `Arg0`, `Arg1`, ... ansprechen und etwa an andere Methoden wie in Zeile 12273 `PCID(Arg0, Arg1, Arg2, Arg3)` ebenfalls als Parameter übergeben. Die Methode `PCID` wird

in dieser Zeile aufgerufen, macht etwas mit diesen 4 Werten und liefert dann einen Wert zurück, welcher dann dank `Return([Rückgabewert von PCID])` jenen Wert als Ergebnis der `_DSM` Methode liefert. Somit kommt dann irgendwo dort, wo ursprünglich die `_DSM` Methode aufgerufen wurde, ein `[Rückgabewert von _DSM]` an. Zurück zum eigentlichen Problem mit den `ArgN` nach der `Return()` Funktion: Sobald die `Return()` Funktion aufgerufen wurde, wird in dieser Methode kein nachfolgender Code mehr verarbeitet, das heißt, dass die 4 herrenlosen `ArgN` Variablen nach dem `Return` vollkommen unnütz sind.

Wie kommen die 4 `ArgN` überhaupt dort hin? Wahrscheinlich wurde die original DSDT mit einer anderen ACPI Version erstellt und wird daher von MaciASL etwas missverstanden.

Ich vermute, dass die Zeile 12278 eigentlich so aussehen müsste: `Return(SDSM(Arg0, Arg1, Arg2, Arg3))`, statt einfach die `ArgN` zu löschen. Euch fällt sicher die Ähnlichkeit zur Zeile 12273 auf. Wenn ich mit meiner Vermutung richtig liege, gibt es irgendwo in der Nähe eine Zeile `Method(SDSM, 4, ...` - die 4 ist ein Indikator dafür, dass `Arg0`, `Arg1`, `Arg2`, `Arg3` nicht einfach auskommentiert gehören, sondern eine mehr oder weniger wichtige Funktion haben, weil die Methode bis zu 4 Parameter erwartet.

Beitrag von „Doctor Plagiat“ vom 7. Juli 2017, 11:46

Vielen Dank [@Thogg Niatiz](#) für die ausführliche Erklärung. Ich als DSDT-Nichtversther bin immer auf so etwas angewiesen. Ich übernehme etwas in die DSDT, was ein anderer an anderer Stelle erklärt hat und hoffe dass es funktioniert, was es meistens auch macht. Um das richtig zu verstehen, benötigt man wahrscheinlich Programmierkenntnisse, die ich leider nicht habe.

Beitrag von „Noir0SX“ vom 7. Juli 2017, 12:15

[@Thogg Niatiz](#) Danke für deine Erklärung.

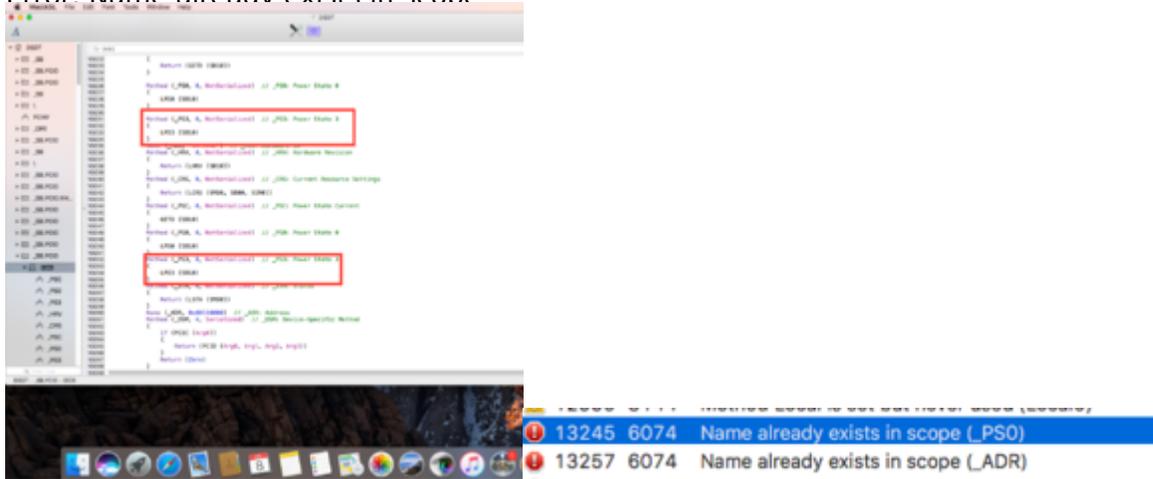
In dem Fall von oben, erledigt sich das ganze ehe von allein `Device (SAT0)` und dem Patch SATA

Beitrag von „Doctor Plagiat“ vom 7. Juli 2017, 12:26

Das wusste ich auch noch nicht. Eben ausprobiert und es funzt. Da haben wir also noch eine weitere Erkenntnis, man muss wissen in welcher Reihenfolge die Patches angewendet werden müssen, dann verschwinden manche Fehler von alleine.

Beitrag von „Noir0SX“ vom 8. Juli 2017, 10:34

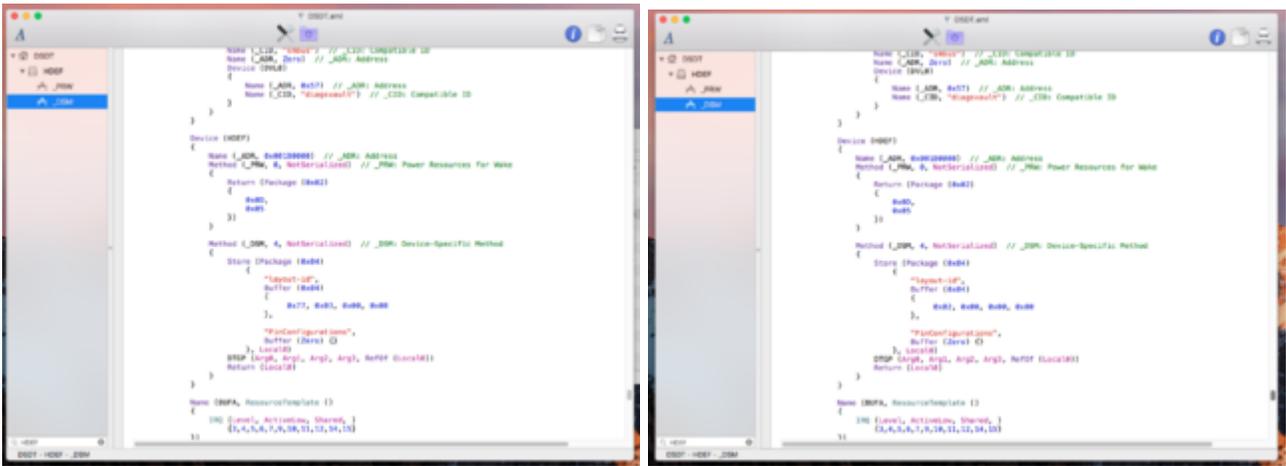
Frror: Name already exists in scope



Methode existiert doppelt, innerhalb des Scope

Beitrag von „derHackfan“ vom 11. Juli 2017, 00:39

Hier noch ein weiteres Beispiel aus der Praxis, der [@Sharky04](#) hat heute im Thread eine DSDT von seinem Gigabyte P55-UD2 Rev 1.0 hochgeladen.



Problem und Lösung: Durch ändern der Layout ID in der DSDT und einsetzen des original AppleHDA.kext läuft jetzt der Sound.



An diesem Beispiel kann man sehr schön erkennen dass das Ändern einer DSDT/HDEF/Layout ID gar nicht so schwer ist.



Ich wünsche viel Erfolg mit dem Thema DSDT und weiter Spaß im Forum.

Gruß derHackfan

Beitrag von „hitman20“ vom 13. Juli 2017, 21:52

Ich schreibe euch hier mal die Schritte wie eine DSDT richtig decompiliert wird, nachdem diese mit Clover extrahiert worden ist. Die extrahierten *.aml Dateien findet Ihr auf der EFI Partition unter EFI/CLOVER/ACPI/origin. Aus diesem Ordner werden eigentlich nur die DSDT Datei und die SSDT Daten benötigt. Diese nehme ich sicherheitshalber auch noch mit falls in einer Datei mal auf eine Externe Referenz verwiesen wird. Die SSDT Daten die ein "x" im Dateinamen haben z. B. "SSDT-4x.aml" werden nicht benötigt. Bitte die DSDT und die SSDT Dateien in einen eigenen Ordner kopieren.

Damit mein vorgehen funktioniert, benötigt Ihr das Programm MaciASL, das es hier im Forum auch zum Download gibt. Bitte dies herunterladen und am besten nach Programme kopieren. Aus diesem Programm benötigt ihr noch die Datei "iasl61". Diese könnt Ihr euch heraus kopieren, in dem Ihr einen Rechtsklick auf MaciASL macht und dort auf "Paketinhalt anzeigen" geht und dann in den Ordner Contents -> MacOS und dort dann die Datei "iasl61" nach /usr/bin kopiert. Wenn die Datei in dem Pfad /usr/bin ist bitte diese nach "iasl" umbenennen.

Wenn Ihr alles richtig gemacht habt, und Ihr ein Terminal öffnet und den Befehl "iasl" eingibt, sollte folgender Auszug kommen mit noch ein paar Optionen mehr, die ich aber jetzt nicht in den Spoiler gepackt habe.

Spoiler anzeigen

Wenn der Befehl funktioniert hat, könnt Ihr dann im Terminal in euren Ordner wechseln in dem die DSDT und SSDT Dateien liegen.

Wenn Ihr dann in eurem Ordner seid, bitte den folgenden Text aus dem Spoiler kopieren:

Spoiler anzeigen

Wenn Ihr diesen Text kopiert habt, dann bitte folgenden Befehl im Terminal ausführen:
pbpaste>refs.txt . Dieser erstellt euch dann die Datei refs.txt mit dem kopierten Inhalt aus der Zwischenablage.

Wenn Ihr das gemacht habt, dann kann man jetzt mit folgendem Befehl die *.aml Dateien korrekt decompilieren. Der Befehl lautet "iasl -da -dl -fe refs.txt DSDT.aml SSDT*.aml". Wenn alles ohne Fehler funktioniert hat, solltet Ihr dann Dateien mit der Endung .dsl erhalten. Danach kann mit maciASL die DSDT.dsl geöffnet werden und Ihr könnt probieren ob sich die Datei bereits ohne Fehler kompilieren lässt. Sollte dies der Fall sein, könnt Ihr euch mit den weiteren Patches beschäftigen die eure DSDT benötigt, damit alles korrekt läuft auf eurem

Hackintosh.

Es kann natürlich auch zu Fehlern kommen wenn Ihr den Befehl "iasl -da -dl -fe refs.txt DSDT.aml SSDT*.aml" ausgeführt habt. wie z.B der im folgenden Spoiler. Dann werden noch keine .dsl Dateien erstellt.

Spoiler anzeigen

Dort kann natürlich auch ein anderer Wert als "MSID" stehen aber das vorgehen ist dann das selbe. Das liegt dann daran das eine oder mehrere SSDT Dateien doppelt sind, falls dieser Fehler mehrfach vorkommt. Die doppelten SSDT Dateien könnt Ihr mit dem Befehl "grep -l MSID *.aml." herausfinden. Ihr bekommt dann die SSDT Dateien angezeigt die diesen Namen alle enthalten. Eine Datei davon könnt Ihr dann löschen und den Befehl "iasl -da -dl -fe refs.txt DSDT.aml SSDT*.aml" nochmal ausführen und dieser sollte dann erfolgreich sein und dann die .dsl Dateien erzeugen. Es kann natürlich sein das Ihr vielleicht mehrere Dateien habt bei denen ein ACPI doppelt vorkommt. Dann müsst Ihr den "grep" Befehl natürlich öfter ausführen und nach den ACPI Namen suchen die der Compiler anmeckert.

Der Output aus dem grep Befehl sieht dann z. B. so aus

Spoiler anzeigen

Meine DSDT konnte ich nach diesem vorgehen ohne Probleme kompilieren. Wenn ich meine DSDT ohne diese Schritte kompiliere bekomme ich folgende Fehler:

Spoiler anzeigen

Ich habe die Fehler auch mal als Screenshot angehängt und auch das Ergebnis nachdem die Aktionen ausgeführt wurden. Manuell musste ich dann nichts mehr bereinigen.

Ich hoffe das war so weit verständlich und hilft euch weiter.

Beitrag von „Doctor Plagiat“ vom 14. Juli 2017, 11:43

Vielen Dank für diese Anleitung. Ich habe das soeben getestet und es hat fehlerfrei geklappt.

Bei dieser Methode besteht nicht die Gefahr, dass man einen Fehler einfach händisch löscht oder auskommentiert und damit vielleicht Informationen eliminiert, die eigentlich noch gebraucht werden. 👍 Sehr schön.

Beitrag von „derHackfan“ vom 14. Juli 2017, 16:04

Irgendwie bin ich wohl zu blöd für die neue Methode. 🤔

Vielleicht kann mir jemand das mal anhand meines origin Ordner erklären, bei mir schmeisst er im Terminal bei der SSDT-3.aml einen Fehler und ich bekomme auch keinen fehlerfreien Compile.

Zum Vergleich habe ich die händische Lösung im Anhang dabei.



Beitrag von „Doctor Plagiat“ vom 14. Juli 2017, 17:25

Vielleicht ist dir nur ein kleiner Fehler passiert. Ich habe die Anleitung mit deinem Ordner nochmal durchgespielt, hat fehlerfrei funktioniert.

derHackfan.zip

EDIT: Hier die Schritte die ich ausgeführt habe.

1. Mit "sudo cp /Applications/MaciASL.app/Contents/MacOS/iasl61 /usr/bin/iasl" iasl nach /usr/bin/ kopiert.
2. Im Terminal in den Ordner gewechselt, in dem die DSDTs und SSDTs liegen.
3. Diesen Text in die Zwischenablage kopiert.
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)

4. "pbpaste>refs.txt" im Terminal ausführen.

5. "iasl -da -dl -fe refs.txt DSDT.aml SSDT*.aml" im Terminal ausführen.

Beitrag von „derHackfan“ vom 14. Juli 2017, 17:54

Zitat von Doctor Plagiat

5. "iasl -da -dl -fe refs.txt DSDT.aml SSDT*.aml" im Terminal ausführen.

Jup, jetzt gehts auch bei mir, der Fehler war dass ich die SSDT einzeln nacheinander abgearbeitet habe. 🤔

```
iasl -da -dl -fe refs.txt DSDT.aml SSDT-0.aml  
iasl -da -dl -fe refs.txt DSDT.aml SSDT-1.aml  
iasl -da -dl -fe refs.txt DSDT.aml SSDT-2.aml  
iasl -da -dl -fe refs.txt DSDT.aml SSDT-3.aml  
iasl -da -dl -fe refs.txt DSDT.aml SSDT-7.aml
```

Die Idee gefiel mir aber ganz gut. 😄

Beitrag von „andreas_55“ vom 16. Juli 2017, 13:09

[ACPI-Introduction copy.pdf](#)Für den groben Überblick (habe ich nur auf Englisch gefunden).

Ich suche immer noch ein Dokument über den strukturellen Aufbau einer DSDT. Bin bisher nur auf das große ACPI Manual (http://www.uefi.org/sites/default/files/resources/ACPI_6_2.pdf) gestoßen, wo man alle möglichen Definitionen und Methoden usw. nachschauen kann. Ist aber eigentlich was für Developer.

Vielleicht hat ja jemand eine Empfehlung?

Edit: Im verlinkten Dokument steht wohl schon alles drin, leider auf ganz ganz vielen Seiten. Grundstruktur findet sich auf S. 253, 254.

Edit 2: Baumstruktur

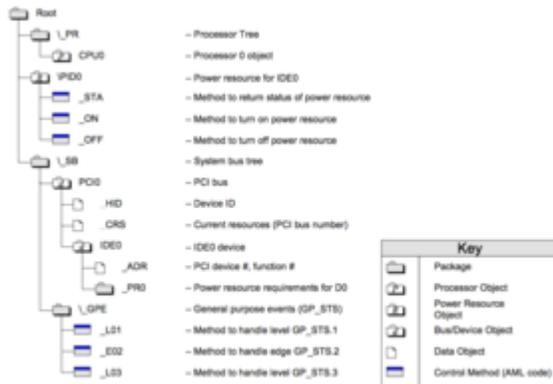


Figure 5-32 Example ACPI Namespace

Beitrag von „derHackfan“ vom 16. Juli 2017, 14:23

Nur das PDF hier welches noch ausführlicher ist aber vielleicht hilft es ja jemandem.

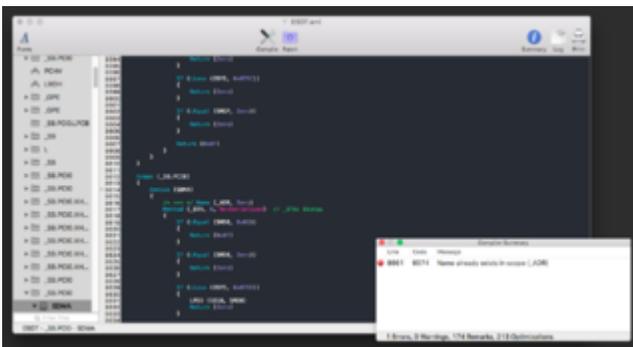
Beitrag von „Thogg Niatiz“ vom 28. November 2017, 19:54

Für alle DSDT Fetischisten hier etwas Abhilfe gegen die recht häufige Compilerwarnung

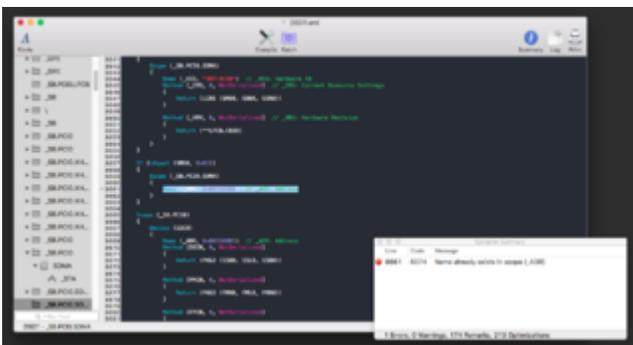
"Missing dependency (Device object requires a _HID or _ADR in same scope)", wobei zumeist die Geräteadresse statt direkt im Device in einem zugehörigen Scope steht:



Durch Anklicken der Warnung ein betroffenes Device ausfindig machen. Normal sollte darin als erstes die Adresse deklariert werden, was hier offensichtlich nicht der Fall ist. Nun könnte man den Scope mit der Adresse suchen, was aber für viele nicht intuitiv sein dürfte...



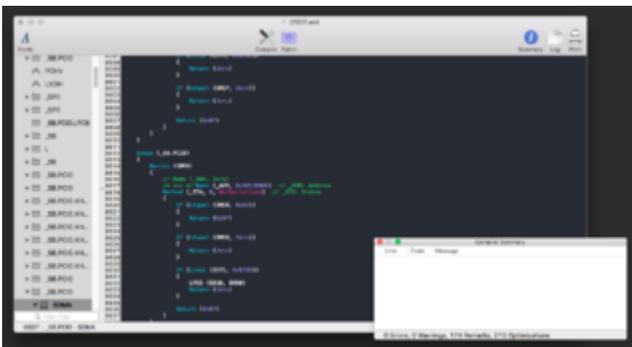
Wenn man einen "Dummy" Adresseintrag macht, wirft der Compiler einen Fehler, da zwei Adressen beim gleichen Device unweigerlich zu Problemen führen würden. Dadurch wird man aber leicht zum gesuchten Scope geführt.



Hier ist die Adresse, welche man gleich ausschneiden und später im Device einfügen kann.



Durch das Ausschneiden ist im Scope nichts mehr enthalten, man kann ihn also komplett entfernen.



Zurück im Device ersetzt man die Adresse 0x0 durch die korrekte Adresse.

Beitrag von „grt“ vom 28. November 2017, 22:02

[Zitat von Thogg Niaticz](#)

eigentlich so aussehen müsste: `Return(SDSM(Arg0, Arg1, Arg2, Arg3)),`

hatte ich mir auch gedacht, und probiert, gab mächtig errors, destawegen hab ich die 4 args dann doch gelöscht. hats bei dir so geklappt?

Beitrag von „Thogg Niatiz“ vom 28. November 2017, 22:13

Nein, das war reinste Theorie. Wahrscheinlich ist die Lösung bei dir ähnlich, aber da kann ich so ohne Weiteres leider nicht mehr zu schreiben. Vielleicht gibt es bei dir auch Probleme, weil ein relevanter Teil in eine SSDT ausgelagert wurde, aber das ist unwahrscheinlich.

Beitrag von „snookerap“ vom 25. Februar 2018, 16:43

Hi Leute,

so richtig hab ich es noch nicht gecheckt... 😞

Hab mir jetzt schon diverse Anleitungen durchgelesen, leider verwirrt es mich dadurch immer mehr..

Hier aber vielleicht eins nach dem anderen, hoffe das ich auch an der richtigen Stelle poste.

Wie extrahiere ich mit Clover oder einem anderen Tool ein cleane DSDT auf meinem BIOS ?

Beitrag von „al6042“ vom 25. Februar 2018, 16:50

Mit Clover einfach die F4-Taste im Clover Bootmenü drücken.

Damit werden die DSDT und andere ACPI-Tabellen-Dateien in den Ordner /EFI/CLOVER/ACPI/origin extrahiert.

Beitrag von „snookerap“ vom 25. Februar 2018, 16:57

hab diese nun extrahiert, siehe Anhang.

Könnte die sich jemand mal bitte anschauen?
Hab einen Skylake (i3-6100 mit MSI Z170A PC Mate)

Vielen Dank...

Beitrag von „Noir0SX“ vom 25. Februar 2018, 17:07

Deinen ersten Fehler bekommst Du so `BNUM, 8, // BNUM, 8,` weg, damit kannst Du dann schonmal arbeiten.

Was zum lesen gibt es im Wiki <https://www.hackintosh-forum.de/threads/gepatchten-DSDT-Desktops/>

Immer schön nach jedem Patch schauen ob Fehler kommen und speichern und dann zum nächsten....

Beitrag von „snookerap“ vom 25. Februar 2018, 17:32

Da gehts schon los...
wo muss ich diese 2 Einträge einsetzen?

Beitrag von „derHackfan“ vom 25. Februar 2018, 17:39

Du machst einfach `//` oder zwei Schräger vor `BNUM` und klickst auf Complie. 😊

Beitrag von „snookerap“ vom 25. Februar 2018, 17:57

die Fehler mit dem gelben Ausrufezeichen kann ich erstmal ignorieren?

Beitrag von „derHackfan“ vom 25. Februar 2018, 18:10

Das sind keine Fehler, das sind Warnings, bitte genau lesen! 😄

Ja, jetzt hast du einen clean Compile und kannst mit dem eigentlichen patchen beginnen.

Beitrag von „derHackfan“ vom 25. Februar 2018, 19:20

Btw: Kann mir jemand helfen bei meinem SSDT Gefummel?

Spoiler anzeigen



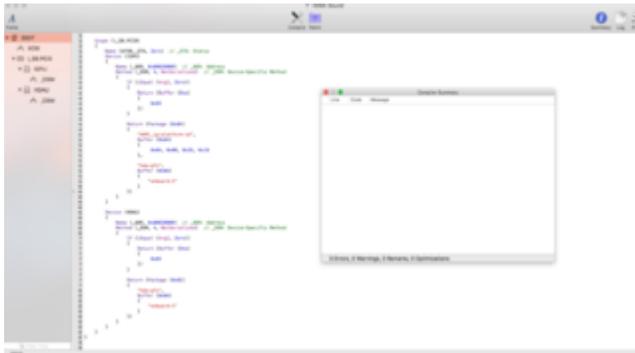
Ich finde den Fehler nicht ...

Beitrag von „al6042“ vom 25. Februar 2018, 19:24

Und ich finde die Zeile, in der der Fehler sein soll, nicht... 😊

Beitrag von „derHackfan“ vom 25. Februar 2018, 19:38

Hm, jetzt geht es plötzlich ... 🤔



Beitrag von „snookerap“ vom 25. Februar 2018, 19:43

kann mir jemand noch sagen, wie ich die Patches für Skylake`s einfüge?

Beitrag von „McRudolfo“ vom 25. Februar 2018, 19:46

[@derHackfan](#)
Das kenne ich 😊

Beitrag von „snookerap“ vom 25. Februar 2018, 20:09

mit dem Patchen soweit klar...
Leider habe ich keine passende Repo... für Skylake.

Hat jemand eine Repo für mich, wo möglichst viele Patches für Skylake verfügbar sind???

Beitrag von „al6042“ vom 25. Februar 2018, 20:15

Für Skylakes gibt es keine Repo...
Da muss man etwas tricksen und entsprechend der Anleitung unter <https://www.hackintosh-forum.de...gepatchten-DSDT-Desktops/> rangehen.

Beitrag von „snookerap“ vom 25. Februar 2018, 20:22

ja okay,
aber welche Repo muss ich nutzen?
Aktuell wird keine angezeigt.

Beitrag von „al6042“ vom 25. Februar 2018, 20:32

Das liegt daran, dass die Links aus dem ursprünglichen .plist-File nicht mehr vollständig erreichbar sind.
Abhilfe schafft folgender Link -> [MaciASL](#)

Beitrag von „snookerap“ vom 25. Februar 2018, 21:08

was mache ich falsch?

die plist habe ich in ~/Library/Preferences gepackt, leider zeigt mir trotzdem keine Repos an.
hat jemand die Repo als URL?

Beitrag von „derHackfan“ vom 25. Februar 2018, 21:10

[Zitat von snookerap](#)

die plist habe ich in ~/Library/Preferences gepackt, leider zeigt mir trotzdem keine Repos an.

Hast du mal ein NVRAM Reset im Terminal gemacht?

MaciASL beenden
sudo nvram -c
Neustart ...

Beitrag von „snookerap“ vom 25. Februar 2018, 21:21

jetzt eben mal gemacht, leider ohne Erfolg...

als URL möglich einzufügen?

Beitrag von „snookerap“ vom 27. Februar 2018, 06:41

Ich komm hier nicht so richtig weiter...

Ist eine gepatchte DSDT nötig wenn alles soweit funktioniert (manueller Ruhezustand, SpeedStep, USB etc.) ???

Beitrag von „derHackfan“ vom 27. Februar 2018, 07:36

Es geht auch ohne, wenn dein System ohne DSDT zufriedenstellend funktioniert dann lass es so, zwingend notwendig ist sie nicht. 😊

Beitrag von „Dr.Stein“ vom 27. Februar 2018, 13:25

SpeedStep macht der ssdt Generator
usb geht such ohne DSDT.

Beitrag von „snookerap“ vom 28. Februar 2018, 06:44

Was heißt denn SSDT Generator?
Mein i3 taktet ordentlich auf 800Mhz runter, heißt doch, dass SpeedStep geht oder?

Beitrag von „BS9“ vom 21. August 2018, 09:54

Guten Morgen 😊

ich habe einige Verständnisfragen zum Thema DSDT:

1. Wo liegt der Vorteil, Patches direkt in der DSDT zu machen anstatt diese im ACPI Tab in der config.plist zu setzen?
2. Wenn ich die DSDT mit F4 extrahiere bekomme ich neben der "DSDT.aml" einige andere Files. Sind diese zur Bearbeitung relevant oder muss ich mich nur um die "DSDT.aml" kümmern?

Danke und Gruß

Beitrag von „kuckkuck“ vom 21. August 2018, 13:28

[Zitat von BS9](#)

Wo liegt der Vorteil, Patches direkt in der DSDT zu machen anstatt diese im ACPI Tab in der config.plist zu setzen?

Nicht alle Patches lassen sich über diese Sektion verwirklichen. Der ACPI Tab bietet häufig nicht alles, was man für ein möglichst vollständig gepatchtes ACPI braucht. Komplexere Arbeiten lassen sich nur durch das manuelle editieren der DSDT verwirklichen. Auch funktionieren die Features von Clovers ACPIPatcher nicht immer in allen Fällen. Manchmal setzt man einen Haken, der passende Patch ist dann aber zB im IOREG nicht zu erkennen. Sollte man jedoch keine komplexen Patches in der DSDT benötigen, ist der ACPIPatcher hervorragend, vorallem weil es Patches in allen ACPI Tabellen vornimmt, womit wir zu Frage 2 kommen:

[Zitat von BS9](#)

Wenn ich die DSDT mit F4 extrahiere bekomme ich neben der "DSDT.aml" einige andere Files. Sind diese zur Bearbeitung relevant oder muss ich mich nur um die "DSDT.aml" kümmern?

Bei den dabei extrahierten Dateien, handelt es sich um alle OEM ACPI Tabellen deines

Systems. Dabei sind auch einige SSDTs. Man kann diese für verschiedenes brauchen.

Beispiele:

- Für das korrekte dekompileieren der DSDT ist es manchmal notwendig, Einträge von SSDTs in die DSDT mit einzubeziehen
- Manchmal sind auch zB SSDTs fehlerhaft. Bei der MATS.aml gab es in der Vergangenheit häufig Probleme mit fehlerhaften Headern. Dann konnte man die MATS.aml bearbeiten und gepatcht wieder injecten
- Nimmt man Renames in den Tabellen vor, sollten diese nicht nur in der DSDT gemacht werden, sondern in allen ACPI Tabellen, damit Abhängigkeiten nicht verloren gehen. Auch hier muss man also teilweise die weiteren ACPI Tabellen bearbeiten und injecten.
- Manchmal will man Änderungen an bestimmten Devices vornehmen, diese befinden sich aber nicht in der DSDT, sondern zb in einer der OEM SSDTs. Daraufhin bearbeitet man dann diese

Beitrag von „BS9“ vom 21. August 2018, 13:37

Danke für die ausführliche Antwort 😊

Beitrag von „kaneske“ vom 21. August 2018, 18:58

Ist einer der Spezialisten so nett mir eine DSDT bzw passende SSDT für mein System zu basteln? Ich musste meine löschen da mein System nicht mehr hoch fuhr mit der alten, warum auch immer.

Danke für Hilfe, ich kann das garnicht.

Beitrag von „al6042“ vom 21. August 2018, 19:25

Die SSDT wirst du selbst mit dem Tool [ssdtprgen.sh](https://github.com/ptitSeb/ssdtprgen) erstellen müssen, damit auch tatsächlich

deine CPU dort eingetragen wird.

Unter Umständen kennt das Tool aber deine Coffeelake CPU noch nicht.

Dann solltest du dich an [@rubenszy](#) halten, der dich mit dem [CPUFriend](#) bekannt machen kann.

Was die DSDT angeht, kannst du die Original-Datei per F4-Taste im Clover-Bootmenü extrahieren.

ACHTUNG: Es gibt weder akustische noch optische Rückmeldung, dass das Drücken der F4-Taste etwas getan hat, deswegen macht es Sinn die Taste mehrfach zu drücken.

Du findest die extrahierte DSDT.aml dann im Ordner /EFI/CLOVER/ACPI/origin, auf der versteckten EFI-Partition.

Poste die Datei, zusammen mit deiner config.plist und ich schaue mir das mal an.

Die config.plist muss ggf. auch angepasst werden, damit dort befindliche DSDT-Patches/-Fixes sich nicht im Nachgang mit der gepatchten DSDT beißen.

Beitrag von „kaneske“ vom 21. August 2018, 19:43

[@al6042](#) vielen Dank für die schnelle Antwort. Ich sende die Dateien gleich rüber.

Hier die Dateien aus meiner EFI:

Beitrag von „al6042“ vom 21. August 2018, 20:09

Na dann...

Benenne deine Original config.plist zu "config-alt.plist um und platziere beiliegende Datei an deren Stelle.

Die DSDT.aml muss in den Ordner /EFI/CLOVER/ACPI/patched.

Viel Erfolg...

Beitrag von „kaneske“ vom 21. August 2018, 20:37

Mit der Kombi kommt bei kaltstart das im Verbose und es ist genau so wie bei der anderen dsdt

Beitrag von „al6042“ vom 21. August 2018, 20:39

Kannst du im Safe Mode booten?
Bitte teste auch den NVRAM-Reset per F11-Taste.

Beitrag von „kaneske“ vom 21. August 2018, 20:56

Beides leider nicht nur ohne die DSDT kommt er hoch...

Beitrag von „al6042“ vom 21. August 2018, 21:02

Dann poste doch bitte mal deinen kompletten EFI-Ordner.

Beitrag von „kaneske“ vom 21. August 2018, 21:09

Ja gerne, nochmal DANKE für die tolle Hilfe bereits.

Anliegend der Ordner.

Ich weiß auch nicht ob alles darin so passt.



Beitrag von „al6042“ vom 21. August 2018, 21:20

Dann teste bitte die beiliegende EFI...

Benenne deinen alten Ordner vorher bitte zu "EFI-alt" um...

Änderungen:

Kexte aktualisiert und dabei auch den neuen Whatevergreen, anstatt den alten Shiki und IntelGraphicsFixup eingesetzt.

Zudem das SMBIOS auf einen iMac18,3 geändert, der näher an deiner Hardware ist, als der iMacPro1,1...

Beitrag von „RichardIstSauer“ vom 21. August 2018, 22:46

Hallo, brauche ich überhaupt eine DSDT ? Den Sinn hinter einer SSDT verstehe ich, aber den einer DSDT noch nicht so ganz. Wäre nett, wenn es mir jemand erklären könnte. Braucht man durch sie weniger Treiber ?

Beitrag von „al6042“ vom 21. August 2018, 23:23

Zum Teil schon, vor allem, wenn die Kexte (Treiber) tatsächlich nur die spezielle Kennnummern

eines Geräts injecten.

Jedes Gerät (Audio-Codec, Netzwerk- oder Grafikkarte, USB-Controller, usw.) hat eine Kennnummer, die sich aus der Hersteller-ID (Vendor-ID) und der Geräte-ID (Device-ID) zusammensetzt.

Bei einigen Geräten kann man sich zusätzliche Kexte ersparen, in dem in der DSDT an der entsprechenden Stelle Hinweise auf die Kompatibilität zu Original-Apple-Geräten einbaut. Dazu muss aber erklärt werden, was die DSDT eigentlich macht.

Die DSDT.aml ist eine ACPI-Tabelle, die im BIOS eines Rechners integriert ist und eine Art Lageplan der auf dem jeweiligen Mainboard verbauten Geräte darstellt.

An dieser Stelle sei gesagt, dass es hier nur um fest integrierte Geräte geht, nicht um Zusatzkarte, wie z.B. eine AMD-/Nvidia-Grafikkarte.

Innerhalb der DSDT haben diese Geräte bestimmte "Device"-Bezeichnungen, die man auch über den IORegistryExplorer einsehen kann.

Da gibt es z.B. folgende "Devices":

erwarteter macOS Device-Name	Windows Device-Name der B/H/Z-Boards und Laptops der Serien 77/81/87/97/110/170/270/370	tatsächliches Gerät
IGPU	GFX0	interne Grafik der CPU
GFX0	PEGP	externe AMD-/Nvidia Grafik
SATA	SAT0	interner SATA-Controller
HDEF	HDAS (ab Skylake)	interner Audio Codec
GLAN	GLAN, IGBE	interner LAN Controller
RPxx (xx = fortlaufende Nummer)	RPxx, EXPx (x = fortlaufende Nummer)	Zum Teil sind das die PCIe-Slots oder interne PCIe-Anschlüsse

Die Liste lässt sich weiterführen.

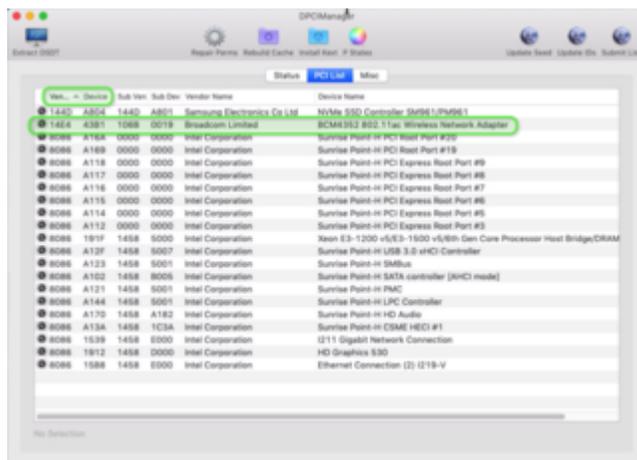
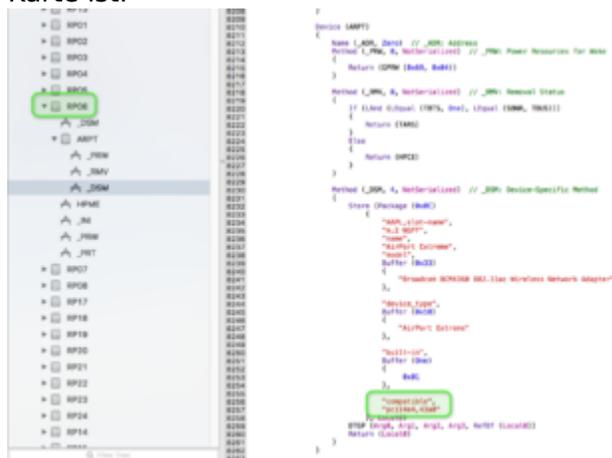
Bei X58/X79/X99/X299er Boards sind die Windows Device-Namen nochmal komplett anders...

Um das Beispiel mit dem Kexten nochmal aufzugreifen:

Es gibt WLAN-Karte von Atheros und Broadcom, die nach Einbau nicht unbedingt automatisch funktionieren.

Zur Unterstützung kann dann entweder ein entsprechender Inject-Kext eingebunden werden oder man identifiziert per IORegistryExplorer das entsprechende RPxx-Device und injected in

der DSDT, bei diesem "Device (RPxx)" einen Code-Schnipsel, in dem das System davon überzeugt wird, dass das verbaute Gerät kompatibel zu einer Original Apple-tauglichen WLAN-Karte ist:



Hier wurde eine WLAN Karte von Broadcom, die im DPCI-Manager eine Apple-nicht-bekannte Kennnummer (Vendor-/Device-ID) besitzt, durch den Codeschnipsel

Code

1. "compatible",
2. "pci14e4,43a0"

so "umgebogen", dass der entsprechende Treiber "AirPortBrcm4331.kext" ohne zusätzlichen Injector-Kext geladen und genutzt wird

Beitrag von „RichardIstSauer“ vom 21. August 2018, 23:31

Danke für die ausführliche Erklärung. Kann ich mit der DSDT auch mein Touchpad und Tastatur Problem lösen ? Da ich keinen Treiber finde, der vernünftig funktioniert.

Beitrag von „al6042“ vom 21. August 2018, 23:36

Nope, das wird wahrscheinlich nicht funktionieren, da die verbauten Geräte nicht annähernd mit den macOS-Kexten funktionieren können, so wie es in dem Beispiel mit der Broadcom WLAN-Karte der Fall ist.

Beitrag von „kuckkuck“ vom 21. August 2018, 23:51

[Zitat von RichardSauer](#)

Tastatur Problem

Wenn es sich um ein Tasten-Zuordnungs Problem handelt, kann das sogar möglich sein, ist aber ziemlich kompliziert...

Man muss wissen, dass eine DSDT nicht wirklich ausführbaren Code, wie zB eine Treiber-Kext, beinhaltet. Man kann also keine Geräte zum Leben erwecken, für die es gar keine Ressourcen bzw kompatiblen ausführbaren Code in macOS gibt... Man kann eher noch ausführbaren Code durch das umstrukturieren/bearbeiten des Geräte-"Lageplans" beeinflussen 😊

PS: [@al6042](#) Tolle Zusammenfassung!

Beitrag von „kaneske“ vom 22. August 2018, 09:06

[Zitat von al6042](#)

Dann teste bitte die beiliegende EFI...

Hab ich nun getestet: keine Besserung selbes Verhalten.
Auch NVRAM gelöscht. Bootet ins Verbotsschild, ohne DSDT bootet er sofort ohne Murren.

Beitrag von „al6042“ vom 22. August 2018, 09:36

Und er hängt immer am PTID Sensor?
welche BIOS-Version nutzt du?

Beitrag von „kaneske“ vom 22. August 2018, 10:01

Ich nutze BIOS F6 und hatte auch F7 getestet.
Ja genau da hängt er immer fest.

Beitrag von „al6042“ vom 22. August 2018, 10:08

Ich kann leider erst heute Abend wieder weiter nach Lösungen suchen.

Beitrag von „kaneske“ vom 22. August 2018, 10:17

Ganz großes DANKE!

Habe nochmals mehrfach herunter und hoch gefahren, nun macht er das auch mit der DSDT...komisch...

anscheinend läuft das System nun wieder sauber, aber warum der manchmal hängt kann ich immer noch nicht deuten.

Beitrag von „Noir0SX“ vom 30. September 2018, 16:17

[@Thogg Niatiz](#) was mache ich hier [DSDT Sammelthread \(Hilfe und Diskussion\)](#) nach deiner Anleitung falsch.

Wenn ich, hoffe mal es ist korrekt umgesetzt, das so mache kann ich die Schnittstelle nicht mehr benutzen.

```
13246 Scope (_SB.PCI0)
13247 {
13248     Device (I2C0)
13249     {
13250         Name (_ADR, Zero)
13251         Name (LINK, "\\_SB.PCI0.I2C0")
13252         Method (_PSC, 0, NotSerialized) // _PSC: Power State Current
13253     }
13254 }
13255
13256 Compiler Summary
13257 Line Code Message
13258 13248 3141 Missing dependency (Device object requires a _HID or _ADR in same scope)
```

```
13246 Scope (_SB.PCI0)
13247 {
13248     Device (I2C0)
13249     {
13250         Name (_ADR, Zero)
13251         Name (LINK, "\\_SB.PCI0.I2C0")
13252         Method (_PSC, 0, NotSerialized) // _PSC: Power State Current
13253     }
13254 }
13255
13256 Compiler Summary
13257 Line Code Message
13258 13295 6074 Name already exists in scope (.ADR)
```

```

13291 // (LEqual (SPR0, 0x02))
13292 {
13293     Scope (_SB.PCI0.I2C0)
13294     {
13295         //Name (_ADR, 0x00150000) // _ADR: Address
13296         Method (XDSM, 4, Serialized) // _DSM: Device-Specific Method
13297         {
13298             If (PCIC (Arg0))
13299             {
13300                 Return (PCID (Arg0, Arg1, Arg2, Arg3))
13301             }
13302             Return (Zero)
13303         }
13304     }
13305 }
13306 }
13307
13308 Scope (_SB.PCI0)
13309 {
13310     // ...
13311 }
13312
13313
13314

```

Line	Code	Message
13295	6074	Name already exists in scope (_ADR)

```

13246 Scope (_SB.PCI0)
13247 {
13248     Device (I2C0)
13249     {
13250         //Name (_ADR, Zero)
13251         Name (_ADR, 0x00150000) // _ADR: Address

```

```

13246 Scope (_SB.PCI0)
13247 {
13248     Device (I2C0)
13249     {
13250         //Name (_ADR, Zero)
13251         Name (_ADR, 0x00150000) // _ADR: Address
13252         Name (LINK, "\\_SB.PCI0.I2C0")
13253         Method (_PSC, 0, NotSerialized) // _PSC: Power State Current
13254     }

```

Beitrag von „Panther“ vom 5. Oktober 2018, 08:03

Ich würde mich hier einmal reinhängen; Ich habe meinen [E470](#) zum Laufen bekommen > Dennoch erhalte ich immer einen Hinweis nach einem Neustart, dass ich ein Problem hatte und ein Report erzeugt wird. Der Report sagt folgendes aus:

Spoiler anzeigen

Zur Frage: Ich habe mir via F4 unter CLOVER die DSDT erstellt. Liegt im Ordner ORIGINAL:

1. MaciASL konnte die DSDT öffnen.
2. Ein Compile verlief ohne Fehler.

Wenn ich es richtig verstanden haben > sind alle HW-Spezifischen Infos in einer DSDT. Hier kann ich unterschiedliche Anpassungen vornehmen. Oder? Das Thema der Batterie und weitere Anpassungen.

Brauche ich dann noch die KextsToPatch Einträge in der config.plist?

Warum diese Frage:

Der Patch: **com.apple.driver.AppleIntelSKLGraphicsFramebuffer** ist aktiv - nur wenn dieser aktiv ist, startet mein Hacki ansonsten habe ich eine Bootschleife.... liegt wohl am [DVMT](#) welches ich nicht im BIOS setzen kann.

Kann ich dies alles via DSDT lösen/beheben?

Ich habe doch einen HD620 > KBL und kein SKL -> Sorry, stehe auf dem Schlauch und sehe den Wald vor lauter Bäumen nicht :(.

Im Anhang meine config.plist und meine DSDT.aml > aus dem ORIGINAL Ordner.

Danke Euch. VG Panther

Beitrag von „derHackfan“ vom 10. Oktober 2018, 09:59

[Panther](#) kannst du den Panic Report bitte in einen Spoiler schieben, in der mobilen Ansicht scrollt man sich sonst einen Wolf. 😄

Beitrag von „Harper Lewis“ vom 10. Oktober 2018, 10:24

Ich empfehle hier die Verwendung von WhateverGreen + Devices > Properties in der

config.plist. Eine Anleitung ist [hier](#) zu finden.

Beitrag von „Panther“ vom 10. Oktober 2018, 10:54

[Zitat von derHackfan](#)

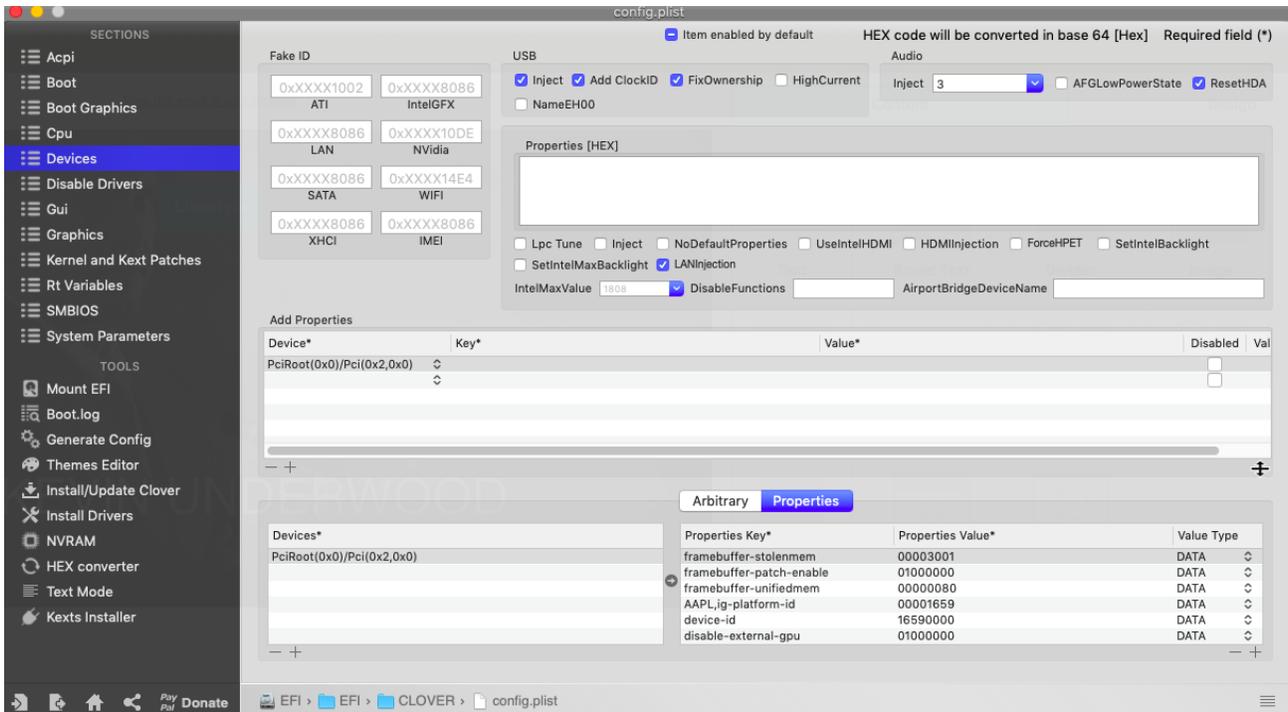
[Panther](#) kannst du den Panic Report bitte in einen Spoiler schieben, in der mobilen Ansicht scrollt man sich sonst einen Wolf. 😄

Erledigt und Danke für den Hinweis 😊

Hinweis:

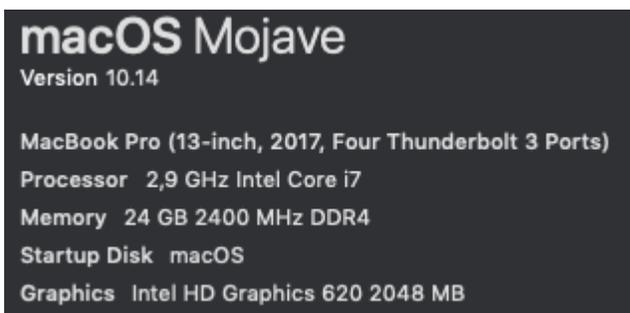
[Harper Lewis](#) Ich habe deine Link verfolgt und mich ein wenig eingelesen. Hierzu habe ich ein paar Fragen?

1. Ich habe ./gfxutil -f IGPU ausgeführt.
2. Unter Devices eingebunden und mir werden die Properties angezeigt.



- Unter Key und Value habe ich bei **Devices** nichts eingetragen. Zumindest konnte ich nichts finden.

Die Grundsätzliche Frage: Meine HD620 läuft ohne Probleme - nur die KernlPanic nach einem Neustart zum Thema FrameBuffer nervt ein wenig. Was muss ich hier nun einstellen?



[Noir0SX](#) : Hast Du hier ggfs. einen Tipp für mich!

Unter KextsToPatch habe ich folgende Einträge:

Comment
Remove 15 port limit completely 10.14 B2
Boot graphics glitch 10.13.x (IGPUs)
Skylake APIC fix
Disable minStolenSize less or equal fStolenMemorySize a...
KBL-5912000-Port_0x5-DP2HDM
KBL-5912000-Port_0x6-DP2HDM
KBL-5912000-Port_0x7-DP2HDM
IONVMeFamily - ignore FLBAS bit:4 being set

Danke für eure Unterstützung!

VG Panther.

Beitrag von „Harper Lewis“ vom 10. Oktober 2018, 12:50

Add Properties und Arbitrary bitte leeren. Ebenso die ig-platform-id und Inject Intel entfernen. Dann solle es mit WEG passen, wenn die Properties unter Devices > Properties korrekt eingetragen wurden. Wie hast du bisher die NVDIA deaktiviert? Aja, die Kexts-To-Patch-Einträge (bis auf port limit und IONVME, falls die benötigt werden) bitte ebenfalls deaktivieren und falls unter Devices für die iGPU eine FakeID eingetragen ist, muss die auch raus.

Beitrag von „Panther“ vom 10. Oktober 2018, 13:04

[Harper Lewis](#) :

1. Properties: Ok, sind leer - wurden automatisch übernommen - gelöscht!
2. Arbitrary: Ok, sind leer.

3. ig-plattform-id und Inject-Intel: Dies hatte bei mir zum Effekt der funktionierenden HD620 geführt?!
 1. 😊 - ich versuche es mal ohne ...
 4. NVIDA: BootArgument: Disable
 5. FakeID: Nicht gesetzt.
 6. Kexts-To-Patch: Disable gesetzt.
-

Beitrag von „Harper Lewis“ vom 10. Oktober 2018, 13:07

Sieht doch gut aus, dann viel Erfolg! Bei NVIDA Disable bin ich mir nicht sicher, ob das in Kombination mit disable-external-gpu zu Problem führen kann. Müsstest du ausprobieren.

Beitrag von „Panther“ vom 10. Oktober 2018, 13:10

Zitat von Harper Lewis

Sieht doch gut aus, dann viel Erfolg!

Rückfrage: Kann ich ig-plattform-id vollständig leeren oder. muss "irgendetwas" eingetragen werden? Ich gehe von Variante 1 aus: LEER?!

Danke.

Bitte [HD620: WhateverGreen & Lilu: FrameBuffer Kernal Panic unter System Reports](#) weitermachen - ich möchte diesen Thread nicht mit meinem Problem "belästigen"! - Danke.

Beitrag von „KMBeatz“ vom 11. Dezember 2018, 01:38

Hallo liebe Hackintosh Gemeinde.

Versuche grad eine Installaion von Mojave 10.14.2 auf einem Lenovo Miix 510 Tablet mit Hd5200.

Habe es geschafft das soweit alles läuft.

Nur Touch geht nicht und Trackpad.

Wollte die DSDTaml Kompilieren aber nix da 😊

Fehler Arg1.

Wie kann ich den Fehler beseitigen?

Beitrag von „Harper Lewis“ vom 11. Dezember 2018, 08:39

Ich würde Arg0 bis Arg3 komplett entfernen, so dass in der Zeile nur noch Return (SDSM) steht. SDSM ist anscheinend vom Typ Integer.

Code

1. If (CondRefOf (_SB.PCI0.SAT0.SDSM))
2. {
3. Return (SDSM)
4. }

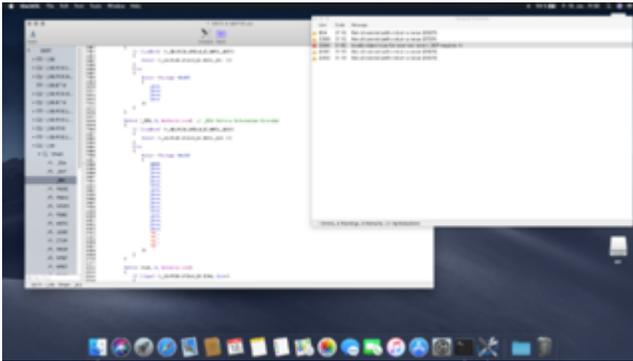
Beitrag von „KMBeatz“ vom 11. Dezember 2018, 12:50

Super hat geklappt.

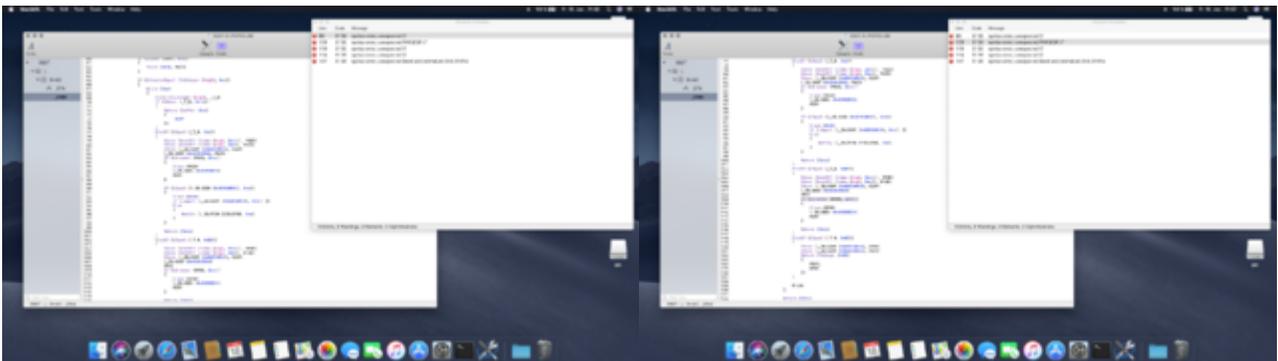
Danke vielmals.

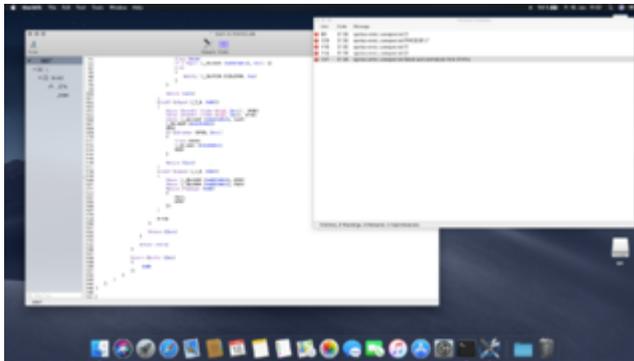
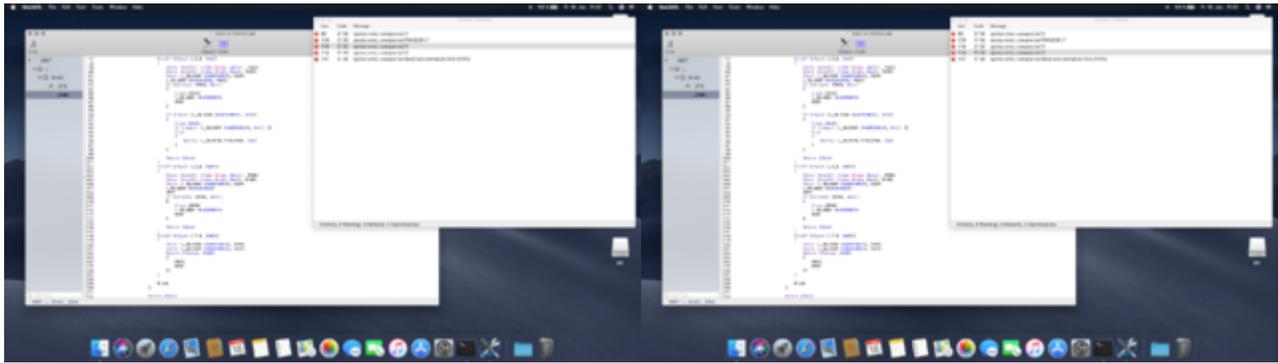
Beitrag von „an3k“ vom 18. Januar 2019, 22:14

Zwar verstehe ich Programmiersprachen (if-then-else, while, Klammern, etc.) aber in Bezug auf xxDTs ist der Inhalt das Problem. In meinem Fall habe ich 17 von 20 xxDTs die ohne Fehler kompilieren, wie ich die drei reparieren kann war und ist mir nicht klar.

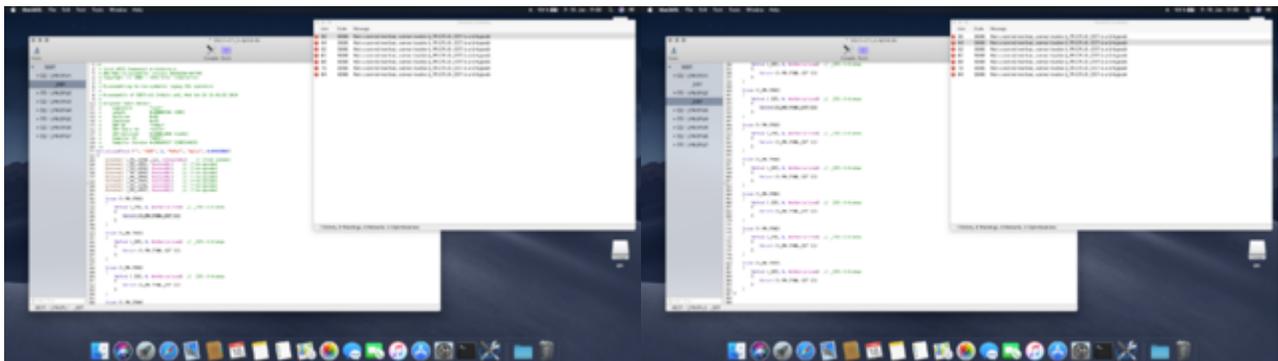


_SCP ?? Sehe ich nirgendwo und wie soll Zero dies definieren? Und warum sind die restlichen Zeros kein Problem? Und schon weiß ich nicht weiter 😞





Dank dem zweiten Posting hier von [derHackfan](#) scheint man "wahllos" (so sieht es für mich aus) löschen zu können. Ich habe mal als Test OLDV in Zeile 87, DFUE in Zeile 108 und OLDV in Zeile 113 auskommentiert und konnte die SSDT ohne Fehler kompilieren. Habe diese gepatchte SSDT aber nicht benutzt da ich keine Ahnung habe ob man da einfach so löschen kann. Wie kommen die Texte da hin, wenn sie "angeblich" falsch sind? Und kann man also "einfach so" was löschen?



Und ich bin wieder an dem Punkt wo ich absolut keine Ahnung mehr habe 😞

Da ich nicht nur diese SSDTs patchen möchte sondern auch verstehen will woher man weiß was man wie ändern muss, würde ich es auch sehr gerne verstehen. Hoffe jemand erklärt sich

zu einem der beiden Punkte bereit 😊

Beitrag von „KMBeatz“ vom 19. Januar 2019, 21:46

Hallo liebe Hackintosh Gemeinde.

Kann bei einem Asus Laptop die DSDT nicht kompilieren.

Wäre Lieb wenn einer mir helfen könnte.

MFG

Beitrag von „griven“ vom 22. Januar 2019, 22:42

Hier mal fehlerbereinigt zurück: [DSDT.aml](https://www.hackintosh-forum.de/attachment.php?attachmentid=1044)

Beitrag von „derHackfan“ vom 22. Januar 2019, 22:48

[Zitat von an3k](#)

Da ich nicht nur diese SSDTs patchen möchte sondern auch verstehen will woher man weiß was man wie ändern muss, würde ich es auch sehr gerne verstehen.

Ist das noch aktuell und kannst du mal die besagte SSDT hier im Thread hochladen?

Beitrag von „an3k“ vom 25. Januar 2019, 21:26

[Zitat von derHackfan](#)

Ist das noch aktuell und kannst du mal die besagte SSDT hier im Thread hochladen?

Ja, ist es. Hier die SSDTs. War die letzten Tage dabei wieder ein sauberes Windows Image zu erstellen und dann Dual-Boot mit schöneren Clover-Einträgen zu basteln. Habe fertig 😊

Beitrag von „KMBeatz“ vom 26. Januar 2019, 00:45

Hallo.

Habe wieder mal ein Problem mit einer DSDT.aml.

Kriege den Fehler nicht weg.

Parseop_Notify, expecting

Beitrag von „McRudolfo“ vom 26. Januar 2019, 16:23

Versuche es mal hiermit.

Beitrag von „griven“ vom 1. Februar 2019, 22:02

[KMBeatz](#) und war der Versuch von Erfolg gekrönt?

Beitrag von „dstyl“ vom 2. Februar 2019, 11:19

Hab ne komplett zermurkste DSDT, und würde vor allem die ssdt1.dsl patchen um native helligkeitskontrolle zu bekommen.

DSDT(Muss gepacht werden da komplett fehlerhaft)

15083, 6126, syntax error, unexpected PARSEOP_IF

Code

```
1. If (LNotEqual (SMD0, 0x02))
2. {
3. Name (_HID, "INT3442") // _HID: Hardware ID
4. Method (_HRV, 0, NotSerialized) // _HRV: Hardware Revision
5. {
6. Return (LHRV (SB10))
7. }
8.
9. Method (_CRS, 0, NotSerialized) // _CRS: Current Resource Settings
10. {
11. Return (LCRS (SMD0, SB00, SIR0))
12. }
13.
14. Method (_STA, 0, NotSerialized) // _STA: Status
15. {
16. Return (LSTA (SMD0))
17. }
18. }
```

Alles anzeigen

15102, 6126, syntax error, unexpected PARSEOP_IF

Code

```
1. If (LEqual (SMD0, 0x02))
2. {
3. Name (_ADR, 0x00150000) // _ADR: Address
```

```

4. Method (_DSM, 4, Serialized) // _DSM: Device-Specific Method
5. {
6. If (PCIC (Arg0))
7. {
8. Return (PCID (Arg0, Arg1, Arg2, Arg3))
9. }
10.
11. Return (Buffer (One)
12. {
13. 0x00
14. })
15. }
16. }

```

Alles anzeigen

15119, 6126, syntax error, unexpected '}', expecting \$end and premature End-Of-File
nach 2 schließenden klammern.

Die SSDT1.dsl wäre für die gpu und das display zuständig

106, 6030, Access width of Field Unit extends beyond region limit

Code

```

1. OperationRegion (NVHM, SystemMemory, 0x5EAAF418, 0x03E3)
2. Field (NVHM, DWordAcc, NoLock, Preserve)
3. {
4. LMID, 16,
5. LPID, 16,
6. LHOR, 16,
7. LVER, 16,
8. SDIL, 32,
9. SDOL, 32,
10. GACD, 16,
11. GATD, 16,
12. NACD, 16,
13. LDES, 8,
14. DHPE, 8,
15. DHPS, 8,
16. USPM, 8,

```

17. GPSP, 8,
18. TLPL, 8,
19. TLSN, 8,
20. DOSF, 8,
21. CTLT, 8,
22. PWMF, 16,
23. MIND, 16,
24. MAXD, 16,
25. GTTP, 32,
26. GPSC, 32,
27. MXML, 32,
28. MXM3, 1600,
29. MX2L, 32,
30. MX32, 1600,
31. K3DL, 32,
32. K3DB, 400,
33. GVKL, 32,
34. GVKB, 1920,
35. GDKL, 32,
36. GDKB, 1920
37. }

Alles anzeigen

Wäre cool wenn mir da jemand helfen könnte 😊

Anmerkung: hab die firmware error Fehlermeldungen alle aaskommentiert, weis halt nicht ob das sinnvoll war.

Beitrag von „KMBeatz“ vom 3. Februar 2019, 00:21

[McRudolfo](#)

Danke hat geklappt.

Beitrag von „McRudolfo“ vom 3. Februar 2019, 10:02

[KMBeatz](#)

Danke für die Rückmeldung!

Beitrag von „Arkturus“ vom 3. Februar 2019, 10:44

Ich habe hier mal eine Frage zur Geschäftsordnung 😊

Ich habe mir einen ein ThinkPad X1 carbon i7-5600 (3theGen.) zugelegt. Hier im Forum habe ich dazu nur wenig gefunden, aber woanders eine m.E. recht ausführliche Beschreibung, einschl. vorbereiteter *.dsl zum patchen. Unter Mojave hat das allerdings nicht funktioniert. Versuche mich gerade mit HS.

Den Sammelthread habe ich gerade gefunden und [hier](#) das WIKI für Desktops.

Worin besteht dabei der Unterschied zu Notebooks?

Sollte ich etwaige Fragen hier stellen oder einen eigenen Thread aufmachen?