

Erledigt

DSDT Fehler lassen sich nicht entfernen

Beitrag von „THack87“ vom 21. September 2018, 07:38

Hallo Forum 😊 ,

ich versuche meine DSDT momentan auf Vordermann zu bringen.
Leider kriege ich die Fehler einfach nicht raus 😞

Sobald ich versuche sie zu entfernen kommen leider wo anders wieder neue.

Ich hoffe das mir einer von euch evt. schnell helfen könnte dabei?
Habe die DSDT als Anhang hochgeladen.

Danke im Voraus!

MfG

Beitrag von „al6042“ vom 21. September 2018, 08:10

Wenn ich mit [Maciasl 1.3.1](#) die ersten beiden Fehler (Liste an Zeros und einmal Arg0) entferne, ist die DSDT im Nachgang sauber und zeigt nur noch Warnungen an.

ACHTUNG:

Nutze die aktuellste Version der dazugehörigen [.plist](#), damit alle Sourcen auch drinnen sind.

Mit der [Version 1.5.2](#) treten im Nachgang noch 5 Fehler auf.

Die ersten vier kannst du durch auskommentieren der betreffenden Zeilen klären

```
81 // External (_SB_.PCI0.GFX0.USB0, UnknownObj)
82 External (_SB_.PCI0.GFX0.STAT, UnknownObj)
83 External (_SB_.PCI0.GFX0.TCHE, UnknownObj)
84 // External (_SB_.PCI0.PEG0, UnknownObj)
85 // External (_SB_.PCI0.PEG0.PEGP, UnknownObj)
86 // External (_SB_.PCI0.PEG1, UnknownObj)
87 // External (_SB_.PCI0.PEG2, UnknownObj)
88 External (DIDX, UnknownObj)
89 External (GSMI, UnknownObj)
90 External (IGDS, IntObj)
```

Der letzte sagt auch genau, was zu tun ist.

Ändere Name (_HID, "pnp0c14") zu Name (_HID, "PNP0C14")

Beitrag von „THack87“ vom 21. September 2018, 21:28

[@al6042](#)

Sobald ich Arg0 entferne, ist alles voll mit Fehlern und Warnungen.

Zero sind dagegen natürlich sehr einfach wegzukriegen.

EDIT:

Hab den Fehler gefunden. Vielen Dank 😊

Hab auf 1.5.2 geupdated und jetzt hab ich nur noch Warnungen.

EDIT 2:

Das auskommentieren der Zeilen hilft beim speichern.

Beim nächsten öffnen der Datei sind die Fehler wieder vorhanden und ich muss sie neu

auskommentieren.

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

Welche Version von Maciasl nutzt du denn?

EDIT:

Na also... 😊

Beitrag von „THack87“ vom 21. September 2018, 22:09

[@al6042](#)

Hat geklappt und habe grade die SSDT in die DSDT eingefügt und mein System kann jetzt auch endlich ohne Probleme "Power Nappen"

Aber wie gesagt... jedes mal wenn ich die DSDT neu öffne muss ich die 4 Zeilen neu auskommentieren. Wie kommt das? Dies ist dann einfach verschwunden trotz speicherns.

Beitrag von „griven“ vom 21. September 2018, 22:10

Korrekt der Compiler entfernt Kommentare beim übersetzen der DSL in AML...

Beitrag von „grt“ vom 22. September 2018, 09:46

der entfernt die gesamten kommentare, aber doch nicht die // oder /* .. */ und lässt den auskommentierten text stehen.

eigentlich sollten auskommentierte zeilen nach kompilieren und speichern als aml dort nicht mehr vorhanden sein.

oder versteh ich hier was falsch?

Beitrag von „Brumbaer“ vom 22. September 2018, 11:46

Für einen Compiler sind alle Kommentare gleich, er weiß nicht ob man Code auskommentiert oder eine Bemerkung schreibt. Kommentar ist Kommentar,

ACPI kennt asl (ACPI Source Language, ASL) und aml (ACPI machine language, AML) Dateien.

aml Dateien sind für das UEFI lesbar, aber für Menschen nur sehr schlecht lesbar. Sie entsprechen einem ausführbaren Programm. aml Dateien enthalten aber kein Programm im Sinne des Prozessors. Es enthält einen Zwischencode, den ein Interpreter-Programm auf dem Computer ausführt.

AML kennt keinen Zwischencode für Kommentare. deshalb können in einer aml Datei keine Kommentare gespeichert werden und gehen somit verloren.

asl Dateien sind für Menschen-lesbar, aber UEFI kann nichts damit anfangen. Sie entsprechen den Sourcen bzw. Quellen eines Programmes. Es sind Text Dateien mit anderem Namen. Da es Text-Dateien sind bleibt alles erhalten was man darin speichert. Da eine asl Datei nicht ausführbar ist, muss ihr Inhalt auch nicht im Sinne von ACPI korrekt sein und kann Fehler enthalten.

Aus einer asl Datei kann man mit Hilfe eines Compilers eine aml Datei machen. Dabei wird überprüft ob der Text in der asl Datei ein fehlerfreies "Programm" geschrieben in ASL ist. Nur dann wird aus der asl eine aml Datei.

MacIASL.

Entsprechend kennt auch MacIASL aml und asl Dateien. Allerdings heißen die asl Dateien in

MacIASL dsl Dateien.

MacIASL hat drei Komponenten, Editor, Compiler und Decompiler.

Die sichtbarste ist der Editor. Es ist ein normaler Texteditor mit etwas Schickschnack.

Der Editor bearbeitet keine aml Dateien, genau genommen auch nicht nur dsl/asl Dateien sondern alle Texte.

Da der Text die Grundlage für eine aml Datei bilden soll und wenn man den Text als Text speichern will standardmäßig die Endung dsl bekommt, sagt man halt es ist ein DSL oder AML Editor.

Da ein Texteditor allen Text, den man rein schreibt, speichert, speichert MacIASL was auch immer man reinschreibt, solange man ihn als dsl Datei speichert.

Will man nun eine für das UEFI verständliche aml Datei haben, muss man den Text kompilieren. MacIASL hat da 'nen Button für.

MacIASL versucht dann eine aml Datei zu erzeugen, die dem Text entspricht. Das gelingt nur, wenn der Text der Spezifikation der ASL entspricht, sprich keine Fehler hat.

Die aml Datei ist losgelöst vom angezeigten Text. Sie ist was eigenständiges, wie eine Übersetzung in eine andere Sprache. Und da die Sprache keine Kommentare kennt, werden die Kommentare nicht mit übersetzt, sondern alles was zwischen bzw. hinter den Kommentarzeichen steht in der Übersetzung weggelassen.

Also will man seine Kommentare behalten, muss man die Datei als dsl speichern und nicht nur die aml Datei behalten.

Ändert man die dsl, ändert sich die aml nicht automatisch. Man muss die dsl erneut kompilieren.

Die aml wird auch nicht gespeichert, wenn man MacIASL nicht sagt sie soll gespeichert werden.

Öffnet man eine aml Datei in MacIASL, so wird die Datei in eine dsl Datei verwandelt. Das ist das Umgekehrte zum dsl in aml Verwandeln und heißt deshalb decompilieren.

Decompilieren ist nicht so einfach, denn der Zwischencode ist nicht immer eindeutig. So haben ein "return 0" und ein "return" den selben Zwischencode. Endet eine Methode, die diesen Zwischencode enthält, nicht mit einem solchen Zwischencode, so ist es möglich, dass da ein return 0 vergessen wurde, und es wird eine Warnung angezeigt. Ein weiteres Problem ergibt sich dadurch, dass sich die aml datei auf andere Dateien beziehen kann, von denen MacIASL aber nicht weiß. Und manchmal gibt es Probleme mit der Interpretation von Gültigkeitsbereichen, etc.

MacIASL kann nur decompilieren, was auch da ist. Also können keine Kommentare aus der aml kommen, denn es sind keine drin.

Aber ...

macIASL fügt beim Decompilieren eigene Kommentare ein. Diese stammen nicht aus der aml, MacIASL erzeugt diese. D.h. wenn man solch einen Kommentar ändert, wird er auch nicht in der aml abgelegt und ist beim nächsten Mal wieder so wie ihn MacIASL einfügt.

Das Decompilieren erzeugt also eine dsl Datei. Diese wird angezeigt und editiert, alles ist so als ob man eine dsl datei geöffnet hätte. Es wird nicht die aml editiert. Das was man im Editor steht ist nicht das was wirklich in der aml Datei steht.

Speichert man eine Datei, die als aml geöffnet wurde, (die keine mehr ist, ist ja jetzt eine dsl), so versucht MacIASL sie standardmäßig als aml zu speichern.

Dazu wird der Text erst kompiliert und falls erfolgreich kompiliert, gespeichert. Das heißt Kommentare, die man noch am Bildschirm sieht sind schon nicht mehr in der aml Datei.

Egal ob man MacIASL mit einer aml oder dsl Datei geöffnet hat, MacIASL bietet immer die Möglichkeit das gerade angezeigte, als dsl oder aml Datei zu speichern. Allerdings muss der Text fehlerfrei kompilieren damit die Datei als aml gespeichert werden kann.

Arbeitet man mit Kommentaren muss man die Datei (zusätzlich) als dsl speichern, denn speichert man sie nur als aml Datei, so sind die Kommentare verloren.

Zusammenfassung

Man hätte es auch in zwei Zeilen sagen können.

1. MacIASL editiert immer nur dsl.
2. Auch wenn MacIASL einem glauben macht man editiere eine aml, editiert man doch in eine dsl und kompiliert sie insgeheim vor dem Speichern.
3. aml (ausführbare) Dateien kennen keine Kommentare.

Beitrag von „kuckkuck“ vom 22. September 2018, 12:13

[Zitat von Brumbaer](#)

Eine weiteres Problem ergibt sich dadurch, dass sich die aml datei auf andere Dateien

beziehen kann, von denen MacIASL aber nicht weiß.

Ebenfalls ein sehr wichtiger Tipp, denn zum korrekten Dekompilieren einer extrahierten DSDT.aml, müssen alle anderen ACPI Tabellen, zu denen sich Referenzen in der DSDT finden, beim Decompile mit einbezogen werden.

Beitrag von „grt“ vom 22. September 2018, 14:14

Zitat von Brumbaer

Für einen Compiler sind alle Kommentare gleich, er weiß nicht ob man Code auskommentiert oder eine Bemerkung schreibt. Kommentar ist Kommentar

schon klar 😊

was mir spanisch vorkam, ist, dass vormals auskommentierte zeilen nach speichern als aml und wieder öffnen wieder als "nicht kommentar" auftauchen sollen.

aber wenn das zeilen sind, die beim dekompile vom maciasl eingefügt werden, ist das klar. da kann man auskommentieren soviel man will, die tauchen wieder auf.

hätte ich doch mal bis nach oben scrollen sollen (die faulheit beim mit handy lesen... 😊)