

Erledigt

Clover, DSDT, SSDT..... ich bin überfordert :S

Beitrag von „Thogg Niatiz“ vom 24. November 2017, 15:27

Die Remarks sind von iASL eingefügte Kommentare, wie "// Device Address". Optimizations sind Änderungen wie 0x0 => Zero oder andere Optimierungen, die die Lesbarkeit und Verständlichkeit der C-ähnlichen Interpretation des Assemblercodes verbessern sollen. Beides geht beim Kompilieren in Assembler, also "speichern als ASL File" verloren und wird beim erneuten Öffnen mit MaciASL wieder erzeugt, also gibts da nichts auszubessern.

Bezüglich der dependency issues: Die Adresse ist meist eh vorhanden, nur in einen Scope ausgelagert. Darum ist das auch kein Fehler, sondern wird sozusagen als "bad practice" von iASL bemängelt. Auch daran lässt sich natürlich etwas machen, auch wenn es in der Regel echt nur kosmetisch für alle ACPI Fetischisten (wie mich 😄) ist, nämlich die Adresse vom Scope direkt ins Device geben. Kleine Bilderstrecke:



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. Kompilieren et voilà - das Mimimi des Compilers hat ein Ende 😊