

Umsetzung prüfen

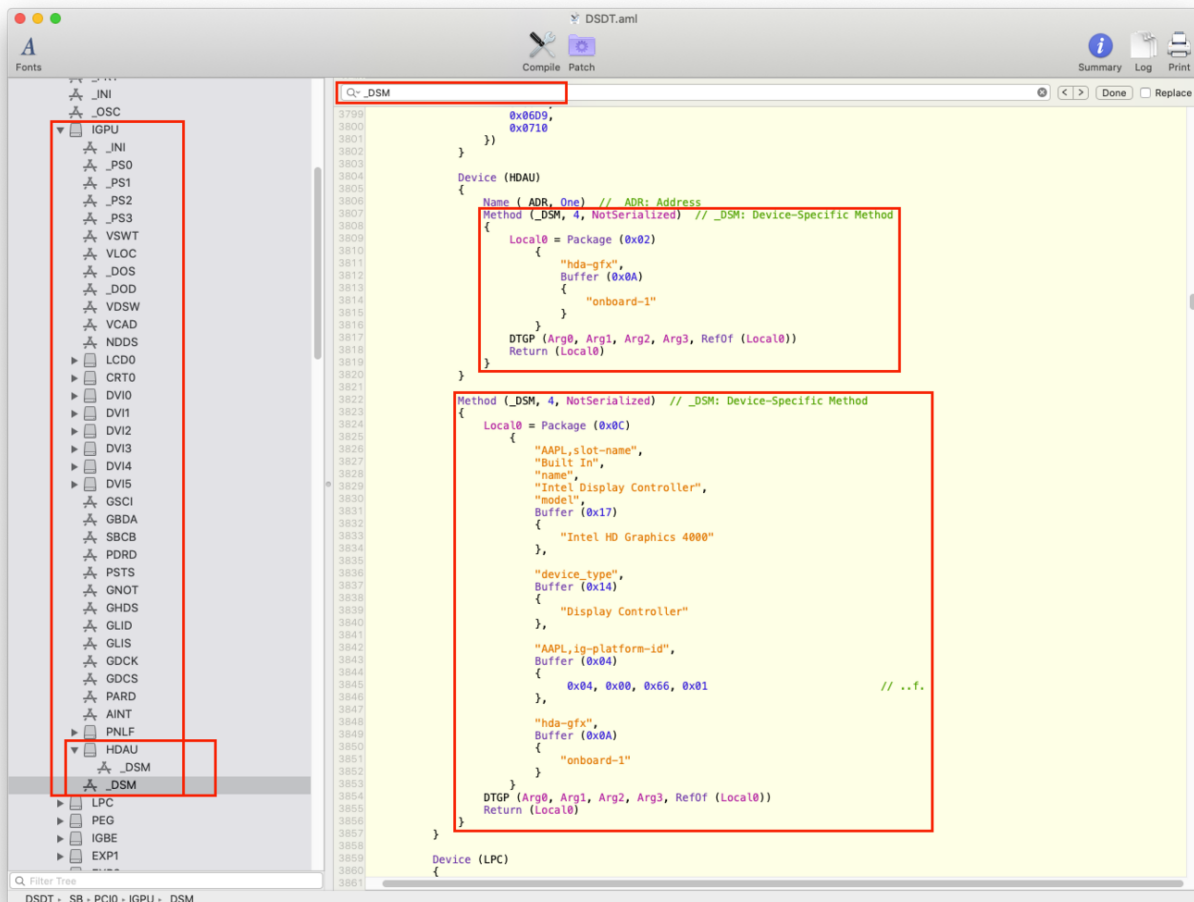
[WIP] Lenovo T530 ohne DSDT unter OpenCore

Beitrag von „apfelnico“ vom 15. September 2020, 21:07

5T33Z0

Die "_DSM-Methoden" nur für macOS zu beschreiben, ist recht einfach. Achte darauf, das nicht jede dieser Methoden macOS betreffen. Doch, alle sind "zusätzlich". Die original in der DSDT vorhandenen sind wohl gelöscht worden.

OK, verfähre nach diesem Beispiel:



The screenshot shows a DSDT editor window titled "DSDT.aml". On the left is a tree view of the ACPI namespace, with the path "IGPU -> HDAU -> _DSM" selected. The main editor area displays the following code:

```
3799         0x0609,
3800         0x0710
3801     }
3802 }
3803
3804 Device (HDAU)
3805 {
3806     Name (_ADR, One) // _ADR: Address
3807     Method (_DSM, 4, NotSerialized) // _DSM: Device-Specific Method
3808     {
3809         Local0 = Package (0x02)
3810         {
3811             "hda-gfx",
3812             Buffer (0x0A)
3813             {
3814                 "onboard-1"
3815             }
3816         }
3817         DTGP (Arg0, Arg1, Arg2, Arg3, RefOf (Local0))
3818         Return (Local0)
3819     }
3820 }
3821
3822 Method (_DSM, 4, NotSerialized) // _DSM: Device-Specific Method
3823 {
3824     Local0 = Package (0x0C)
3825     {
3826         "AAPL,slot-name",
3827         "Built In",
3828         "name",
3829         "Intel Display Controller",
3830         "model",
3831         Buffer (0x17)
3832         {
3833             "Intel HD Graphics 4000"
3834         },
3835         "device_type",
3836         Buffer (0x14)
3837         {
3838             "Display Controller"
3839         },
3840         "AAPL,ig-platform-id",
3841         Buffer (0x04)
3842         {
3843             0x04, 0x00, 0x66, 0x01 // ...f.
3844         },
3845         "hda-gfx",
3846         Buffer (0x0A)
3847         {
3848             "onboard-1"
3849         }
3850     }
3851     DTGP (Arg0, Arg1, Arg2, Arg3, RefOf (Local0))
3852     Return (Local0)
3853 }
3854
3855 Device (LPC)
3856 {
3857
3858
3859
3860 }
```

Wie du siehst, habe ich nach "_DSM" gesucht (Cmd-F). Am Beispiel der "IGPU" mal geschaut, hier siehst du zwei _DSM-Methoden. Die untere ist für das Device "IGPU", die darüber ist für das eingeschobene Device "HDAU" (High Definition Audio, also die Audioeinheit der internen Grafikkarte). Wo die Methoden jeweils enden, siehst du sehr genau durch die wunderbar gestaltete Einrückung des QuellCodes.

Schau dir jetzt an, was ich verändert habe:

```
3803 Device (HDAU)
3804 {
3805     Name (ADR, One) // ADR: Address
3806     If (_OSI ("Darwin"))
3807     {
3808         Method (_DSM, 4, NotSerialized) // _DSM: Device-Specific Method
3809         {
3810             Local0 = Package (0x02)
3811             {
3812                 "hda-gfx",
3813                 Buffer (0x0A)
3814                 {
3815                     "onboard-1"
3816                 }
3817             }
3818             DTGP (Arg0, Arg1, Arg2, Arg3, RefOf (Local0))
3819             Return (Local0)
3820         }
3821     }
3822 }
3823
3824
3825
3826
3827 If (_OSI ("Darwin"))
3828 {
3829     Method (_DSM, 4, NotSerialized) // _DSM: Device-Specific Method
3830     {
3831         Local0 = Package (0x0C)
3832         {
3833             "AAPL,slot-name",
3834             "Built In",
3835             "name",
3836             "Intel Display Controller",
3837             "model",
3838             Buffer (0x17)
3839             {
3840                 "Intel HD Graphics 4000"
3841             },
3842             "device_type",
3843             Buffer (0x14)
3844             {
3845                 "Display Controller"
3846             },
3847             "AAPL,ig-platform-id",
3848             Buffer (0x04)
3849             {
3850                 0x04, 0x00, 0x66, 0x01 // ..f.
3851             },
3852             "hda-gfx",
3853             Buffer (0x0A)
3854             {
3855                 "onboard-1"
3856             }
3857         }
3858         DTGP (Arg0, Arg1, Arg2, Arg3, RefOf (Local0))
3859         Return (Local0)
3860     }
3861 }
3862
3863
3864
3865
```

Die Einrückung musst du nicht machen, beim sichern, schließen und erneutem öffnen ist der Quellcode perfekt angepasst. Du siehst, dass ich vor der eigentlichen Methode

Code

1. If (_OSI ("Darwin"))
2. {

eigefügt habe, die Klammer ist geöffnet. Entsprechend der Syntax muss ich nach der Methode noch eine geschlossene Klammer hinzufügen. Somit wird der Inhalt, in diesem Fall die Methode "_DSM" nur ausgeführt, wenn die Systemabfrage positiv hinsichtlich "Darwin" erfolgt. Windows wird das nun ab sofort ignorieren.

Das ist eine einfache Übung. Da ich aber nicht die originale DSDT kenne, denke ich, es wird nicht nur bei diesen beschreibenden Methoden geblieben sein, die Windows auch sicher nicht unbedingt stören. Es sind etliche Devices umbenannt worden, und es fehlen die originalen _DSM-Methoden (die macOS wiederum nicht auswertet, die aber ihren Grund haben). Idealerweise wäre es also, wenn man mit den originalen _DSM-Methoden arbeitet (die macOS gar nicht stören) und zusätzlich die neuen Inhalte für macOS mit einer If-Schleife integriert.

Dann werden da noch weitere "Patches" enthalten sein etc pp.

Dennoch ist eine solche Übung nicht "für die Katz". Ich hoffe, ich konnte dir helfen, viel Spaß.

Edit:

Mal abgesehen von der Übung wäre es in der Tat hilfreich, zu klären, welche Patches denn tatsächlich überhaupt gebraucht werden. Aus verschiedenen Gründen rate ich von einer Bearbeitung der DSDT ab, man sollte schauen, was per SSDT geht und dort dann mittels Darwin-Abfrage arbeiten. ACPI-Patches in OpenCore würde ich auch nicht nutzen wenn irgend möglich, denn auch diese sind systemübergreifend dann vorhanden. Viele dieser Sachen erledigen heute schon Lilu/AppleALC/WhateverGreen – diese benennen Devices selbstständig für macOS genehm um. Und für Windows spielen die natürlich keine Rolle. DSM-Deklarationen (wenn nicht im Original enthalten) gern per SSDT, ansonsten per DeviceProperties über den Bootloader.