

# Quicksync, Virtual-Screen Abstürze und iGPU+ded. GPU mit Grafikbeschleunigung

Beitrag von „kuckkuck“ vom 31. Dezember 2017, 16:06

Welche Auswirkungen der Patch hat, kann ich mir denken, was den Patch aber nicht rechtfertigt. Eine solche Vorgehensweise ist meiner Meinung nach eindeutig in die Kategorie "Very dirty Hack" einzuordnen und sehr extrem.

Device specific Methods (`_DSM`) haben ihren Sinn und sind nicht ohne Grund in OEM ACPI Tabellen eingebaut. "Remove `_DSM` methods" in MaciASL würde alle `_DSM` Methoden aus dem ganzen geöffneten Table (zB DSDT) entfernen um zu verhindern das `_DSM` Methoden, die potenziell von premade Patches gesetzt werden könnten, nicht mit bereits vorhandenen `_DSM` Methoden in Konflikt kommen (1 Device, maximal 1 `_DSM`). `change _DSM to XDSM` sorgt dafür, dass nicht nur die `_DSM` Methoden aus einer Tabelle, sondern aus allen OEM ACPI Tabellen ins leere laufen. Das ist ungefähr gleichzusetzen mit dem Löschen ALLER `_DSM` Methoden aus ALLEN ACPI Tabellen. Das macht für mich absolut keinen Sinn und erst recht nicht wenn `change _DSM to XDSM` einer der ersten Patches ist, die unreflektiert angewandt werden...

Generell macht der Patch ja auch erst Sinn wenn von irgendwo überhaupt `_DSM` Properties injected werden, sprich aus SSDTs oder DSDTs oder durch Fixes in Clover/ACPI. Richtet man seine eigene ACPI Struktur ein, sollte man nicht blind nach Anleitungen vorgehen, sondern seine Patches auch auf das bereits vorhandene anpassen. Meine ACPI Struktur mit Ozmosis besteht aus einigen Renames, die in allen OEM Tabellen gemacht werden + 20 SSDTs, die für verschiedene Geräte unter anderem `_DSM` Methoden einfügen. Bei der Erstellung habe ich mir jedes Device in der OEM DSDT angeschaut, bevor ich eine passende `_DSM` Methode für das Device erstellt habe und in eine SSDT gesetzt habe. Nur in einem Fall hat das Device (IMEI) bereits eine `_DSM` Methode besessen, weshalb ich mit einer SSDT keine weitere `_DSM` injecten konnte. (SSDTs überschreiben keinen Code in DSDTs) Dies ist aber kein Grund für mich einfach wahllos erstmal ALLE `_DSM` Methoden im ACPI zu löschen/deaktivieren.

Stattdessen deaktiviere ich das Device HECI (=IMEI) mit meiner SSDT und setz in dieser eine `_DSM` Methode, die eine Kombination aus der bereits vorhandenen und meinen `_DSM` Einträgen ist. So geht kein Code verloren.

Code

1. Scope (\\_SB.PCI0)
2. {

3. Name (HECI.\_STA, Zero) // \_STA: Status
4. Device (IMEI)
5. {
6. Name (\_ADR, 0x00160000) // \_ADR: Address
7. Method (\_INI, 0, NotSerialized) // \_INI: Initialize
8. {
9. Store (Zero, \\_SB.PCI0.HECI.\_ADR)
10. }
- 11.
- 12.
13. Method (\_DSM, 4, NotSerialized) // \_DSM: Device-Specific Method

Alles anzeigen

Zum Code oben: Das Device HECI müsste für OSX eigentlich IMEI heißen. Ich benenne HECI jedoch nicht in IMEI um und versuche \_DSM zu setzen (das geht wie beschrieben nicht) sondern die SSDT setzt in der Method \_INI ein `zero` für \_ADR von HECI und deaktiviert somit HECI. Daraufhin injecte ich das Device IMEI, das eine \_DSM mit dem original Code + meinen \_DSM Properties besitzt.