

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

Beitrag von „elmacci“ vom 19. Oktober 2017, 16:18

[Zitat von al6042](#)

Das Thema habe ich auch und bin auf eine Lösung gespannt.

Aktuell gibt es m.E. folgende Varianten um das Problem zu lösen bzw. zu umgehen:

1) Workaround OHNE NvidiaGraphicsFixup.Kext

a) AppleGraphicsDevicePolicy-KextToPatch aktivieren von PikeA um das BlackScreen-Problem bei SMBIOS 6,1/15,1/17,1 zu lösen:

Name AppleGraphicsDevicePolicy
Find 626f6172 642d6964
Replace 626f6172 642d6978
Comment (c)Pike R. Alpha

b) NVWebDriverLibValFix.Kext einsetzen um die NVIDIA-Treiber zu validieren (iBooks-Transparenz-Problem). Nur notwendig bis Sierra 10.12.6. In High Sierra werden die WebDriver von Nvidia anders eingebunden.

<https://github.com/mologie/NVWebDriverLibValFix/releases>

2) Workaround MIT NvidiaGraphicsFixup.Kext

Das geht aktuell nur mit einer inoffiziellen Variante in Kombination mit darin enthaltenen neuen Bootargumenten.

Diese Variante ist hier zu finden:

<https://sourceforge.net/p/nvid.../239c/14de/83cc/9879/9588>

Ergänzend müssen in Clover unter Boot folgende beiden Bootargumente ergänzt werden:

- 1) -ngfxnovarenderer
- 2) ngfxpatchlist=pikera

Achtung: das erste hat einen Bindestrich, das zweite nicht

Wobei das letzte Bootargument eigentlich nicht notwendig sein sollte für die meisten. Es definiert nur welche Methode der Kext verwenden soll um das BlackScreen-Problem nach dem Boot bei den SMBIOS-Definitionen 6,1/15,1/17,1 zu umgehen. Wenn man dieses Bootargument nicht angibt wird die Standardmethode verwendet die auch im offiziellen Release eingesetzt wird.

Info am Rande: Nutze ich die Methode von Piker dann habe ich (mit meinem Setup - Your Mileage May Vary) keine Probleme mehr mit meinem Monitoren beim Aufwachen (vor allem nach längerem Sleep). Deshalb setze ich entweder den KextToPatch ein oder den NvidiaGraphicsFixup.Kext mit dem entsprechenden Bootargument.

Die dritte Variante ist natürlich, zurück auf Lilu 1.1.7 zu gehen und NvidiaGraphicsFixup 1.1.3 zu verwenden. Da trat das Problem noch nicht auf.

Der Entwickler wird voraussichtlich im nächsten Release von NvidiaGraphicsFixup die verschiedenen möglichen Bootargumente implementieren.

Zitat von ductator

Hmm, alles ganz komisch. Ich hab die neueste Shiki noch mit dabei und mit shikigva=12 läuft Quicksync, solange die Anwendung manuell freigeschaltet wurde in der Kext. Aber Airplay funktioniert komischerweise gar nicht mehr.

So wie ich das verstanden habe ist shikigva=12 ja nur eine Methode, den VARenderer für eine in der Kext definierte Auswahl an Apps zu "whitelisten", oder? Ich habe beide Methoden getestet und nutze aktuell Methode 2). Shiki habe ich auch im Einsatz, allerdings ohne ein Bootargument (trotz Intel HD530 in Kombination mit GTX 1080TI). Der Vorteil von shikigva=12 liegt ja darin, dass man dann auch QuickSync DECODING hat, nicht nur ENCODING - oder?!

cheers