

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

Beitrag von „kuckkuck“ vom 29. Januar 2018, 16:14

[@armut](#) Bitte IOReg Dumps nicht vom Terminal, sondern über den IORegistryEditor --> Save as. Außerdem ist der Fix mit deinem SMBios leider nicht möglich, mehr dazu gleich.

[@henties](#) Ein sehr schöner Beitrag für einige 👍

## Zitat von henties

Also den einzigen Eingriff den AGDPfix unternimmt ist die Aenderung von "Config2" auf none in der info.plist

Der fix und die Problematik ist mir komplett klar. Die configs (wie Config2) werden, wie du sicherlich weißt, ebenfalls in der Info.plist definiert. Bei none wird AppleGraphicsDevicePolicy.kext nicht geladen.

Config1 und Config3 stellen eine "unload" Funktion zur Verfügung. Auf diese kann statt dem AGDPFix zugegriffen werden. Besitzt das ACPI Device den Namen GFX1 und Config1 wird geladen, wird für dieses ACPI Device die AppleGraphicsDevicePolicy deaktiviert. Dies ist für die zweite GPU des MacPro6.1 vorgesehen, die keine Anschlüsse besitzt und GFX1 heißt. Jeder der Mac-F60DEB81FF30ACF6 benutzt kann also durchs umbenennen des Geräts in GFX1 die AppleGraphicsDevicePolicy umgehen - aber auch nur dieser. Bei [@armut](#) ist dies nicht der Fall, er benutzt iMac17,1 (und Mac-B809C3757DA9BB8D benutzt Config2).

Jeder, der einen der neuesten iMacs als SMBios nutzt und somit auf Config3 zugreift, könnte AGDP umgehen, indem er die GPU in iGPU umbenennt und somit wieder `unload` betätigt. Das ist aber wiederum nicht wirklich sinnvoll, da iGPU noch in ein paar anderen Kexts matched und dort für Probleme sorgen würde.

Ganz generell kann AGDP auch statt dem umbenennen auf `none` durch das aktivieren von `unload` deaktiviert werden. Das ist aber gehüpft wie gesprungen, die Info.plist wird dabei so oder so von Hand oder durch Script geändert, updateproof ist dies nicht.

Anstatt durch script und somit updategefährdet zu verändern, bestand die Idee von lvs1974 darin, die Properties per Kext zu ändern.

### [Zitat von henties](#)

Nun wuerde ich mich aber auch RIESIG freuen wenn einer derjenigen die weiter das Anwenden von Lilu.kext und Nvidiagraphicsfixup.kext befuerworten mal erklaren wuerden was der Einsatz dieser kexte den so im System veraendert/anrichted

Schau dir den SourceCode an... Dieses Schnipsel ist von [hier](#) und beinhaltet nicht alle dependencies, lässt sich aber trotzdem relativ einfach verstehen, wenn die Problematik bekannt ist:

### Code

```
1. bool NGFX::AppleGraphicsDevicePolicy_start(IOService *that, IOService *provider)
2. {
3.     bool result = false;
4.
5.     DBGLOG("ngfx", "AppleGraphicsDevicePolicy::start is called");
6.     if (callbackNGFX && callbackNGFX->orgAgdpStart)
7.     {
8.         char board_id[32];
9.         if (WIOKit::getComputerInfo(nullptr, 0, board_id, sizeof(board_id)))
10.        {
11.            DBGLOG("ngfx", "got board-id '%s'", board_id);
12.            auto dict = that->getPropertyTable();
13.            auto newProps = OSDynamicCast(OSDictionary, dict->copyCollection());
14.            OSDictionary *configMap = OSDynamicCast(OSDictionary, newProps->getObject("ConfigMap"));
15.            if (configMap != nullptr)
16.            {
17.                OSString *value = OSDynamicCast(OSString, configMap->getObject(board_id));
18.                if (value != nullptr)
19.                {
20.                    DBGLOG("ngfx", "Current value for board-id '%s' is %s", board_id, value->getCStringNoCopy());
21.                    if (!configMap->setObject(board_id, OSString::withCString("none")))
22.                    {
23.                        SYSLOG("ngfx", "Configuration for board-id '%s' can't be set, setObject was failed.", board_id);
24.                    }
25.                }
26.            }
27.            else
28.            {
29.                SYSLOG("ngfx", "Configuration for board-id '%s' can't be set, setObject was failed.", board_id);
30.            }
31.        }
32.        return result;
33.    }
34.}
```

```

23. DBGLOG("ngfx", "Configuration for board-id '%s' has been set to none", board_id);
24. that->setPropertyTable(newProps);
25. }
26. else
27. {
28. SYSLOG("ngfx", "ConfigMap key was not found in personalities");
29. OSSafeReleaseNULL(newProps);
30. }
31. }
32.
33. result = callbackNGFX->orgAgdpStart(that, provider);
34. DBGLOG("ngfx", "AppleGraphicsDevicePolicy::start returned %d", result);
35. }
36.
37. return result;
38. }

```

Alles anzeigen

Die BoardID (Mac-...) wird über das IOKit abgefragt und mit den BoardIDs in ConfigMap abgeglichen. Je nachdem was vorhanden ist, werden Properties neu gesetzt oder der Wert auf none gesetzt (ebenfalls abhängig vom ngfxpatch bootarg. Ngfxpatch=cfgmap wäre der "none" Eintrag; =vit9696 deaktiviert den kompletten Board-ID Check und =pikera ersetzt einfach Board-ID mit Board-IX, wodurch wie bei =vit9696 Default-->none geladen wird). Der Rest sind größtenteils nur Failsafes, falls beispielsweise ConfigMap nicht in AGDP vorhanden ist. Der Mechanismus ist also praktisch identisch zu der Idee von AGDPFix, nur halt in Kext-Form.

Ich glaube es ist klar, dass eine Kext-basierte Lösung wesentlich besser/angenehmer als ein Script ist, ganz zu schweigen davon, dass dabei keine Apple-eigenen Daten dauerhaft verändert werden. NvidiaGraphicsFixup ist OpenSource, man muss also kein "kein Hacker auf "gut Glueck" sein". "Stricksachen, von Solo Artisten, die mann selber nicht versteht" müssen das auch nicht sein, nebenbei ist Lilu und NvidiaGraphicsFixup auch von zwei verschiedenen Entwicklern 😊

Ich glaube trotzdem **nicht**, dass NvidiaGraphicsFixup perfekt und 100% durchdacht ist, denn wie man sieht passieren manchmal sporadische Fehler, erst recht dann, wenn man sich aus dem allgemein bekannten rausbewegt und gerne neue Wege eingeht.

Ich sehe aber auch keinen Sinn darin Lilu und alle seine Plugins zu verheißen und zu umgehen, denn diese sind häufig sehr nützlich und beinhalten eine menge Wissen von Entwicklern die meist wissen was sie tun. Aber das soll sich jeder selber überlegen 😊