

Erledigt

GTX770 PCIroot Eintrag in DeviceProperties löst KP aus

Beitrag von „calabash“ vom 12. April 2020, 14:38

Hallo,

Mit Hackintool exportierte ich alle PCIroot Devices.

pcidevicelist.plist und config.plist mit Xcode/ProperTree geöffnet und die Einträge kopiert.

Der Eintrag der dGPU löst die Kernel Panic aus:

Code

1. `<key>AAPL,slot-name</key>`
2. `string>Internal@0,1,0/0,0</string>`
3. `<key>device_type</key>`
4. `<string>VGA compatible controller</string>`
5. `<key>hda-gfx</key>`
6. `<string>onboard-2</string>`
7. `<key>model</key>`
8. `<string>GK104 [GeForce GTX 770]</string>`

Die Fehlermeldung mit der Kernel Panic:

```
IOReturn CoreAnalyticsHub::_newUserClientGatad(task_t, void *, UInt32, OSDictionary *, IOUserClient **):257:Successfully opened
virtual IOReturn CoreAnalyticsUserClient::clientMemoryForType(UInt32, IOOptionBits *, IOMemoryDescriptor **):269: got memory descriptor with 16432
hfs: mounted Backup on device disks2
Notice - new kext com.apple.driver.KextExcludeList, v15.1.1 matches prelinked kext but can't determine if executables are the same (no UUIDs).
deferred rematching count 0
IOG flags 0x3 (0x51)
Generation from SMC report as 1
NVDASStartup: Official
panic(cpu 0 caller 0xfffff801ac4c32a): Kernel trap at 0xfffff7f9e923a06, type 14=page fault, registers:
CR0: 0x000000000001003b, CR2: 0x0000000000000000, CR3: 0x000000001fcc3000, CR4: 0x0000000000003626e0
RAX: 0x0000000000000000, RBX: 0xfffff8208c9bd64, RCX: 0xfffff801b6873008, RDX: 0x0000000000000002
RSP: 0xfffff8208c9bd60, RBP: 0xfffff8208c9bd40, RSI: 0xfffff801b6231e08, RDI: 0x0000000000000000
RB: 0xfffff801b4e4360, R9: 0x0000000000000000, R10: 0x0000000000000002, R11: 0x000000007ffffefff
R12: 0x0000000000000000, R13: 0xfffff7f9e923fad, R14: 0x0000000000000000, R15: 0xfffff803b533440
RFL: 0x0000000000010246, RIP: 0xfffff7f9e923a06, CS: 0x0000000000000000, SS: 0x0000000000000010
Fault CR2: 0x0000000000000000, Error code: 0x0000000000000000, Fault CPU: 0x0, PL: 0, VF: 1

Backtrace (CPU 0), Frame : Return Address
0xfffff8208c9b7c0 : 0xfffff801ab215cd mach_kernel : _handle_debugger_trap + 0x49d
0xfffff8208c9b810 : 0xfffff801ac5a3c5 mach_kernel : _kdp_l386_trap + 0x155
0xfffff8208c9b850 : 0xfffff801ac4bf7e mach_kernel : _kernel_trap + 0x4ee
0xfffff8208c9b8a0 : 0xfffff801aac7e40 mach_kernel : _return_from_trap + 0xe0
0xfffff8208c9b8c0 : 0xfffff801ab2b2c97 mach_kernel : _DebuggerTrapWithState + 0x17
0xfffff8208c9b9c0 : 0xfffff801ab21887 mach_kernel : _panic_trap_to_debugger + 0x227
0xfffff8208c9ba10 : 0xfffff801b2c2c7c mach_kernel : _panic + 0x54
0xfffff8208c9ba80 : 0xfffff801ac4c32a mach_kernel : _sync_ios_to_llcs + 0x2aa
0xfffff8208c9bc00 : 0xfffff801ac4c828 mach_kernel : _kernel_trap + 0x598
0xfffff8208c9bc50 : 0xfffff801aac7e40 mach_kernel : _return_from_trap + 0xe0
0xfffff8208c9bc70 : 0xfffff7f9e923a06 com.apple.nvidia.NVDASStartup : __ZN11NVDASStartup5probeEP9IOServicePI + 0x666
0xfffff8208c9ba40 : 0xfffff801b20841a mach_kernel : __ZN9IOService15probeCandidatesEP12OSOrderedSet + 0x7aa
0xfffff8208c9ba70 : 0xfffff801b207ade mach_kernel : __ZN9IOService14doServiceMatchEJ + 0x2de
0xfffff8208c9ba50 : 0xfffff801b20a296 mach_kernel : __ZN15_IOConfigThread4mainEPv + 0x186
0xfffff8208c9bfa0 : 0xfffff801aac713e mach_kernel : _call_continuation + 0x2e

Kernel Extensions in backtrace:
com.apple.nvidia.NVDASStartup(14.0) [B7E8FE38-5A44-3E80-9A30-3458583BFF31] 0xfffff7f9e922008->0xfffff7f9e925fff
dependency: com.apple.lokit.IOPCIFamily(2.9) [181F388B-9212-3CF9-94F8-8FEF803ACEC4] 0xfffff7f9b531000

BSD process name corresponding to current thread: kernel_task
Boot args: -v keepsyms=1

Mac OS version:
19E266

Kernel version:
Darwin Kernel Version 19.0.0: Tue Aug 14 22:03:12 PDT 2018; root:xnu-4903.202~2/RELEASE_ARM_T8020
Kernel UUID: A80A7EE-3D03-3C21-91AD-5719079D7AF6
Kernel slide: 0x000000001a000000
Kernel text base: 0xfffff801aa000000
__HIB text base: 0xfffff801a9000000
System model name: (MacPro1,1) (Mac-7BA5B2D9E4200094)
System shutdown begun: NO
Panic diagnostics file available: YES (0x0)

System uptime in nanoseconds: 13412408774

** In Memory Panic Stackshot Succeeded ** Bytes Traced 255584 **
Translating kernel state, please wait:
100..Recorded panic stackshot in corefile at offset 0x6000, compressed to 80655 bytes
```

Kann man das Fixen?

OC: 0.5.7

Beitrag von „luxus13“ vom 12. April 2020, 14:48

nimm mal Whatevergreen.kext raus

Ich verwende auch in meinem 3. iMac eine GTX770

LG

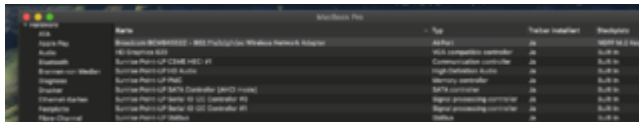
Beitrag von „calabash“ vom 12. April 2020, 15:03

Ohne WEG und mit dem DeviceProperty Eintrag löst es die selbe KP aus.

Das paradoxe ist, das ohne Einträge alles sauber bootet. Auch mit WEG.

Beitrag von „anonymous_writer“ vom 12. April 2020, 23:50

Die PCIROOT EINTRAG IN DEVICEPROPERTIES sind gut um zu Prüfen ob alle Device funktionieren.



Können aber auch zu Problemen führen wie hier KP oder ein nicht funktionierender Sleep.

Wenn die Karte ohne DEVICEPROPERTIES funktioniert einfach weg lassen.

Beitrag von „calabash“ vom 19. April 2020, 14:20

[anonymous_writer](#): hast du vllt dazu eine Erklärung?

ich bin dem [Guide](#) gefolgt um den DRM Status zu prüfen.

Mit dem [VDADecoderChecker](#) wirft er mir eine Fehlermeldung aus:

Code

1. VDADecoderCreate failed. err: -12473
2. An error was returned by the decoder layer. This may happen for example because of bitstream/data errors during a decode operation. This error may also be returned from VDADecoderCreate when hardware decoder resources are available on the system but currently in use by another process.
3. VDADecoderCreate failed. err: -12473

Testing DRM:

FairPlay 1.x. -> OK

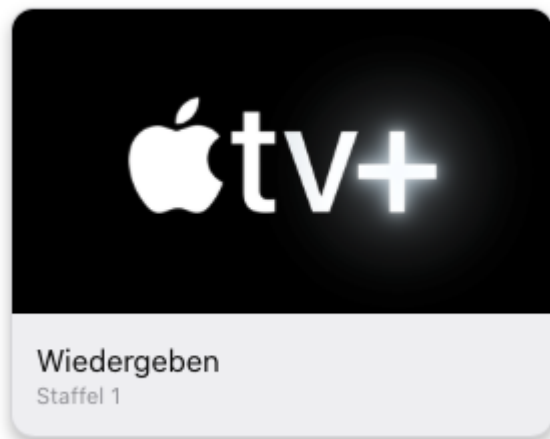
FairPlay 2.x/3.x -> OK

FairPlay 4.x.:

Wenn ich einen Trailer ansehe oder ein gekauftes Video runterlade dann spielt es mit der TV app.

Klicke ich auf

Optionen zum Ansehen



Abonniert

Dann erscheint der Film für ein paar Sekunden im Großformat und schließt dann.

Habe ich nun ein funktionierendes DRM oder muss ich etwas patchen?

Update: Gelöst laut [WhateverGreen's DRM chart](#) mit
shikigva=256