

Erledigt

Verschieden Methoden um Geräte zu den Systeminformationen - PCI hinzuzufügen

Beitrag von „Noir0SX“ vom 22. April 2019, 15:05

Allgemein

Das Hinzufügen (Sichtbar machen) der Geräte und erscheinen unter Systeminformationen - PCI ist für einige nur Kosmetik, kann aber mit bestimmten zusätzlichen Werten auch helfen Geräte ihre Funktionen zu integrieren.

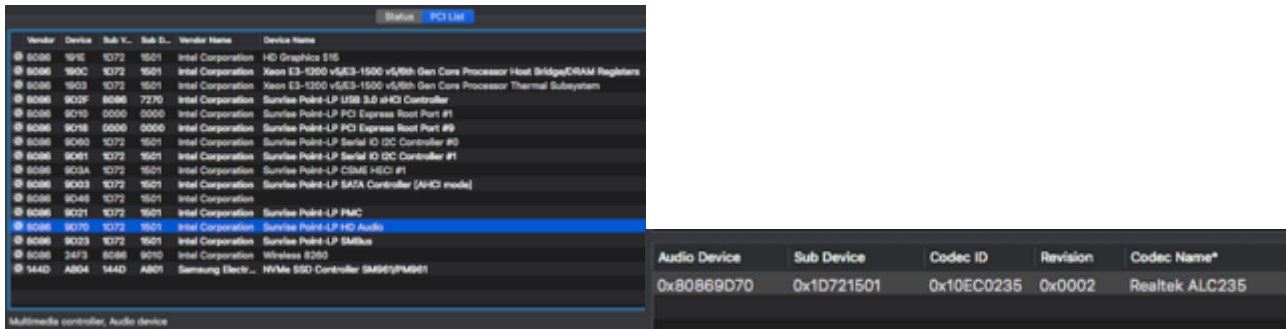
Jede Variante hat dabei bestimmt ihre Vor- und Nachteile und jeder muss für sich herausfinden mit was er besser umgehen kann.

Man kann auch dies Methoden miteinander verwenden, das heisst nicht unbedingt das diese sich stören. Wobei ich, wenn ich mich eine Zeit für was entschieden habe, das dann auch komplett damit umsetzte. Einzige Ausnahme man bekommt etwas damit nicht hin und ist auch auf Variante x noch angewiesen.

Welche Variante ist derzeit Euer Favorit oder bei Fragen, Anregungen weiteren Methoden kann gern hier [Verschieden Methoden um Geräte zu den Systeminformationen - PCI hinzuzufügen \(Hilfe und Diskussionen\)](#) geschrieben werden.

Auf geht es

Hier aufgeführt sind die verschiedenen Arten Anhand vom Beispiel eines Audio Devices



Code

1. AAPL,slot-name = Built In
2. model = Audio device
3. device_type = Multimedia-controller
4. name = Sunrise Point LP HD Audio
5. layout-id = 3

Methoden

1. [DSDT](#)
2. [SSDT](#)
3. [Clover Arbitrary](#)
4. [Clover Properties](#)
5. [PropertyInjector.kext](#)
6. [DevPro](#)
7. [OpenCore Properties](#)

Alle aufgeführten Methoden sind von mir auch auf das Beispiel hin getestet, mit einer Ausnahme von DevPro da mir dort die Hardware und Erfahrung unter OZ (Ozmosis) fehlt bzw. das gewählte Beispiel so nie existiert.

8. Tool (kommt vll später noch, wenn es die Zeit erlaubt)

Beitrag von „Noir0SX“ vom 22. April 2019, 15:07

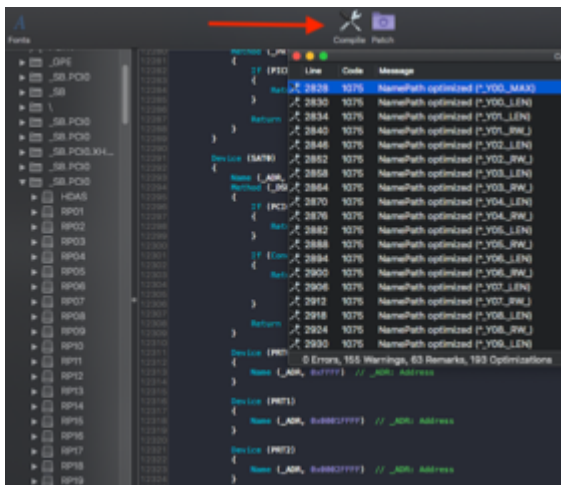
DSDT

Die DSDT extrahieren wir beim Start von Clover mit der F4 Taste.

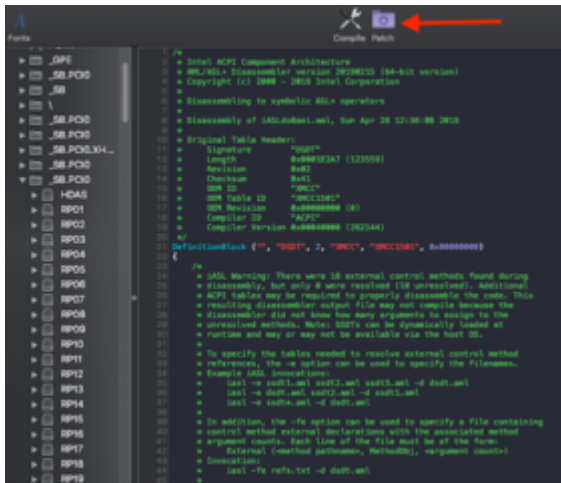
Zum bearbeiten dieser Datei kann man [MaciASL](#) verwenden. Die verwendete Weiterentwicklung kommt von den Entwicklern von Acidanthera und ist auf Github zu finden.

Die DSDT wird im Programm geöffnet und sollte keine Error enthalten. Sehen kann man das durch Klick auf Compile.

Wenn es doch so ist muss man diese erst bereinigen. Wie man das macht ist einige male im Forum bzw. im Wiki des Forums beschrieben.

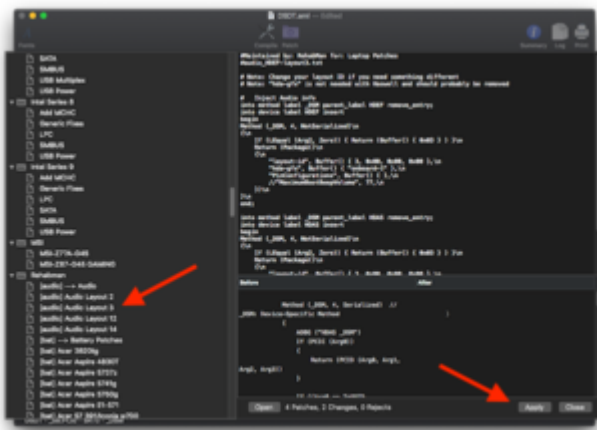


Mit Klick auf Patch öffnet sich ein neues Fenster

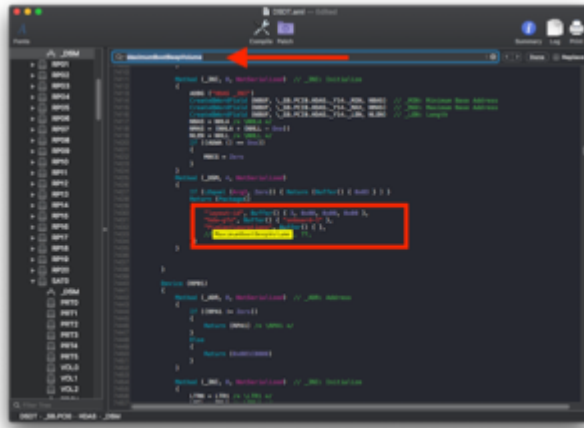


Auf der linken Seite sind schon viele fertige Patches für die DSDT enthalten. Wir suchen einen passenden für uns raus, so unter dem roten Pfeil zusehen. Ein Blick auf die rechte Seite oben zeigt uns diesen, dabei merke (in dem Fall MaximumBootBeepVolume)oder schreibe ein Wort hinein welches nicht so oft in so einer Datei vorkommt. Das macht im nächsten Schritt die Suche der eingefügten Zeilen leichter.

Als letztes übernimmt man den Patch noch mit Apply.



Wenn wir den eben eingefügten Code gefunden haben,



können wir diesen mit unseren Beispiel ergänzen bzw. ersetzen.

Code

1. "AAPL,slot-name", Buffer () { "Built In" },
2. "model", Buffer () { "Audio device" },
3. "name", Buffer () { "Sunrise Point-LP HD Audio" },
4. "device_type", Buffer () { "Multimedia controller" },
5. "layout-id", Buffer () { 0x03, 0x00, 0x00, 0x00 },

Nach dem Einfügen muss das ganze noch abgespeichert werden. Dabei sollte man darauf achten das dies als Maschine Language Binary passiert, sonst kann Clover nichts damit anfangen. Der damit entstandene Code sieht nach dem speichern und erneuten öffnen so aus. Das Formatieren, Einrücken etc übernimmt dabei das Programm.

```

11800
11801     {
11802         Method _SMC, 4, NotSerialized // _SMC Device-Specific Method
11803     {
11804         IF ((Arg0 == Zero))
11805         {
11806             Return (Buffer (One))
11807         }
11808         Else
11809         {
11810             // ..
11811         }
11812     }
11813     Return (Package (One))
11814     {
11815         "AAPL,slot-name",
11816         Buffer (One)
11817     }
11818     {
11819         "Built In",
11820     }
11821     {
11822         "AAPL",
11823         Buffer (One)
11824     }
11825     {
11826         "Audio device",
11827     }
11828     {
11829         "name",
11830         Buffer (One)
11831     }
11832     {
11833         "Sunrise Point LP HD Audio",
11834     }
11835     {
11836         "Device type",
11837         Buffer (One)
11838     }
11839     {
11840         "Multimedia controller",
11841     }
11842     {
11843         "layout-id",
11844         Buffer (One)
11845     }
11846     {
11847         One, One, One, One // ....
11848     }
11849 }

```

Diese entstandene Datei kopieren wir nun nach EFI - CLOVER - ACPI - patched.



Dies ist nach einen Reboot unter Systembericht - PCI zusehen.

Mit der layout-id hat man nun auch Effekt das AppleAlc greift und den Ton ausgibt.





Code

1. `<p align="right"> Zum Seitenanfang</p><hr>`

Beitrag von „Noir0SX“ vom 22. April 2019, 15:08

SSDT

Zum Erstellen der Datei kann man [MaciASL](#) verwenden. Die verwendete Weiterentwicklung kommt von den Entwicklern von Acidanthera und ist auf Github zu finden.

Um nicht bei Null anzufangen und alles zuschreiben kann man vorhandene Dateien nutzen. So wie hier <https://github.com/RehabMan/OS-X-Clover-Laptop-Config> von RehabMan und befindet sich im Verzeichnis hotpatch.

Die Datei fürs Beispiel Audio findet man unter dem Device Namen HDEF.

Nach dem öffnen der Datei mit MaciASL, kann man unseren Beispiel Code

Code

1. `"AAPL,slot-name", Buffer () { "Built In" },`
2. `"model", Buffer () { "Audio device" },`
3. `"name", Buffer () { "Sunrise Point-LP HD Audio" },`
4. `"device_type", Buffer () { "Multimedia controller" },`
5. `"layout-id", Buffer () { 0x03, 0x00, 0x00, 0x00 },`

mit dem original Code ersetzen. Das ist relativ einfach zu finden, wenn man sich an so was wie

model, AAPL, slot-name, device_type und name orientiert.

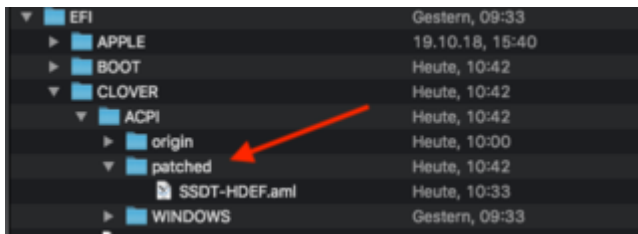
```

1 // Automatic injection of ACPI properties
2 #ifndef ML_DEFNITION_LOCK
3 #define ML_DEFNITION_LOCK 1
4 #endif
5
6 #ifndef ML_DEFNITION_LOCK
7 #define ML_DEFNITION_LOCK 1
8 #endif
9
10 #ifndef ML_DEFNITION_LOCK
11 #define ML_DEFNITION_LOCK 1
12 #endif
13
14 // Note: If your ACPI table (SSDT/ASL) does not define HDEF for AML or HDEF
15 // add this device definition (by uncommenting it)
16 //
17 // #define ML_DEFNITION_LOCK
18 // #define ML_DEFNITION_LOCK
19 // #define ML_DEFNITION_LOCK
20
21 // Inject properties for audio
22 #method ML_DEFNITION_LOCK, 1
23 {
24     if (LsIsA(AML, "HDEF") || (LsIsA(AML, "HDEF") && !LsIsA(AML, "HDEF")))
25     {
26         Local0 = Package()
27         {
28             "AAPL,slot-name", Buffer() { "HDEF" },
29             "AAPL,device", Buffer() { "Audio Device" },
30             "AAPL,slot-id", Buffer() { "Multimedia controller" },
31             "AAPL,slot-id", Buffer() { "HDEF", "HDEF", "HDEF", "HDEF" },
32         }
33     }
34     if (LsIsA(AML, "HDEF"))
35     {
36         CreateMethod(AML, "HDEF", 0, AML)
37         AML = VMPH.AML
38     }
39     // The user can disable "HDEF" injection by defining VMPH or setting VMPH=0
40     // because the "HDEF" is always at index 2 (eg. "HDEF" follows "slot-id")
41     Local1 = 0
42     if (LsIsA(AML, "HDEF")) { Local1 = 1 }
43     if (LsIsA(AML, "HDEF")) { Local1 = 1 }
44     if (LsIsA(AML, "HDEF")) { Local1 = 1 }
45     Return(Local1)
46 }
47 #endif
48 #endif
49 #endif

```

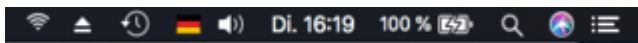
Nach dem Einfügen muss das ganze noch abgespeichert werden. Dabei sollte man darauf achten das dies als Maschine Language Binary passiert.

Sonst kann Clover nichts damit anfangen. Diese entstandene Datei kopieren wir nun nach EFI - CLOVER - ACPI - patched.



Dies ist nach einem Reboot unter Systembericht - PCI zusehen.

Mit der `layout-id` hat man nun auch Effekt das AppleAlc greift und den Ton ausgibt.



Code

1. `<p align="right"> Zum Seitenanfang</p><hr>`

Beitrag von „Noir0SX“ vom 22. April 2019, 15:08

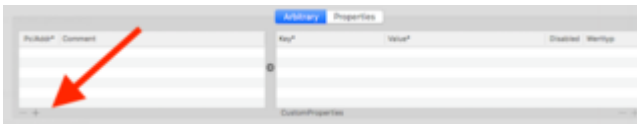
Clover Arbitrary

Um die Clover Config.plist mit den entsprechenden Werten zu füllen, benutzen wir den [Clover Configurator immer aktuell \(mackie100 Credits\)](#)

Unter dem Menüpunkt Devices befindet sich unten,



folgende Auswahl zum eintragen der Werte. Auf der linken Seite wird der Pfad zum Device mit Hilfe des Pluszeichens eingetragen. Diesen kann man ganz leicht mit dem von ermitteln.



Neben dem DPCIManager gibt es ein Tool für das Terminal [dspci MuntashirAkon](#), das ziehen wir nach dem Download ins Terminal und erhalten dann folgende Ausgabe.



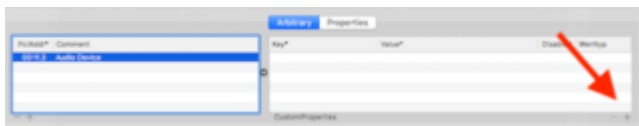
Wenn man dahinter `-j > test.txt` eingibt, erhält man die benötigten Werten in einer Textdatei abgelegt.

Die Ausgabe im Terminal zeigt ganz vorn den Wert, den man in der CloverConfig unter der Spalte PCIAddr

00:1f.3 einträgt. Als Comment daneben kann ein Kommentar wie Audio Device eingetragen werden.

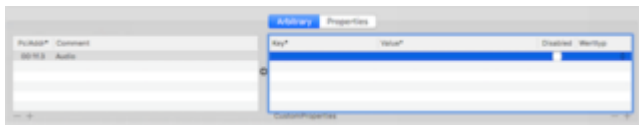


Auf der rechten Seite, ein Klick auf das Pluszeichen erstellt eine neue Zeile,

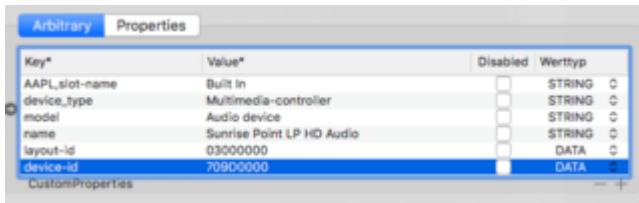


die aus den Werten `Key`, `Value`, `Disabled` und `Werttyp` besteht.

Werttypen sind dabei vorgegeben Data, String und Number.



Durch das setzen der Keys `model`, `AAPL`, `slot-name`, `device_type` und `name` in der Spalte Value als String (Wort), aus unseren Beispiel oben, ist unser Ziel eigentlich schon erreicht.



Mit der `layout-id` und der Spalte Value als Data, hat man nun auch Effekt das AppleAlc greift und den Ton ausgibt.

Dies ist dann nach speichern der Config.plist und nach einen Reboot unter Systembericht - PCI zusehen.



Code

1.

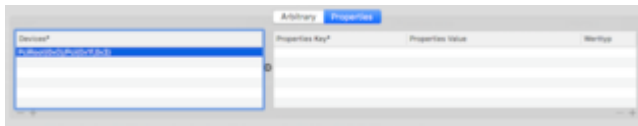
[Zum Seitenanfang](https://www.hackintosh-forum.de/forum/thread/42439-verschieden-methoden-um-ger%C3%A4te-zu-den-systeminformationen-pci-hinzuzuf%C3%BCgen/# "Zum Seitenanfang")

Beitrag von „Noir0SX“ vom 22. April 2019, 15:09

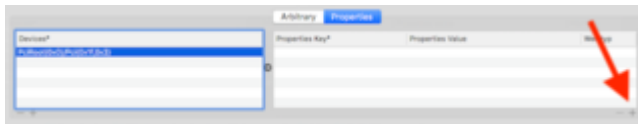
Clover Properties

Um die Clover Config.plist mit den entsprechenden Werten zu füllen, benutzen wir den [Clover Configurator immer aktuell](#)([mackie100 Credits](#))

Unter dem Menüpunkt Devices befindet sich unten,

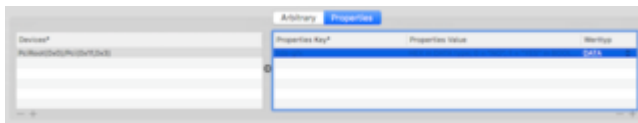


Auf der rechten Seite, ein Klick auf das Pluszeichen erstellt eine neue Zeile,

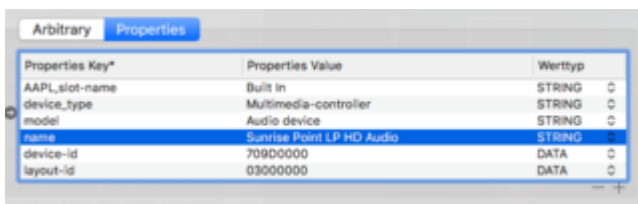


die aus den Werten Properties Key, Properties Value und Werttyp besteht.

Werttypen sind dabei vorgegeben Data, String und Number.

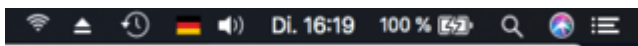


Durch das setzen der Properties Keys `model`, `AAPLslot-name`, `device_type` und `name` in der Spalte Value als String (Wort), aus unseren Beispiel oben, ist unser Ziel eigentlich schon erreicht.



Mit der `layout-id` und der Spalte Properties Value als Data, hat man nun auch Effekt das AppleAlc greift und den Ton ausgibt.

Dies ist dann nach speichern der Config.plist und nach einen Reboot unter Systembericht - PCI zusehen.



Code

1. `<p align="right"> Zum Seitenanfang</p><hr>`

Beitrag von „Noir0SX“ vom 22. April 2019, 15:10

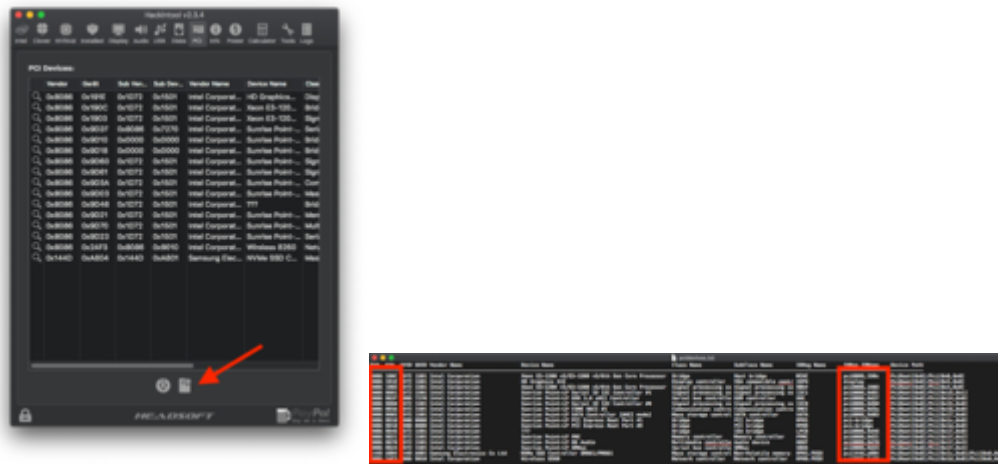
PropertyInjector.kext

Besser und ausführlicher Beschreiben wie es [Brumbaer](#) hier [Geräte Eigenschaften \(Device Properties\) ohne DSDT Patch ändern](#). tut, kann ich es nicht !!!

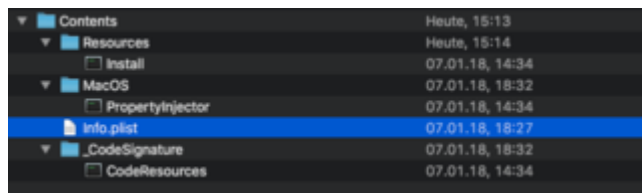
Darum nur ein paar Zeilen anhand unseres Beispiels des Audio Devices.

Die benötigten Werte kann man ganz leicht mit dem [Hackintool \(ehemals Intel FB-Patcher\)](#) von [Headkaze](#) ermitteln.

Dazu rufen wir im Tool den Menüpunkt PCI auf und gehen auf Export. So wird eine Textdatei mit den benötigten Werten abgelegt.



Nach dem Download des Beispiel Kext, kann man den Inhalt und somit die benötigte info.plist mit einem Rechtsklick auf dem Kext unter Paketinhalt zeigen erreichen.



Im geöffneten Zustand (z.B. unter Xcode) tragen wir unter 1 einen Namen ein, hier vll passend Audio, unter 2 wird der Wert aus der Kombination VID DID oder IOReg IOName aus der Textdatei des Hackintool verwendet.

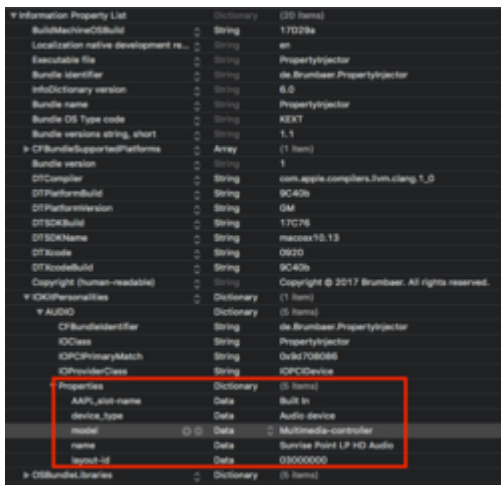
Dies beginnt mit 0xDIDVID also im Beispiel 0x9d708086

Key	Type	Value
Information Property List	Dictionary	(20 items)
BuiltMachineOSBuild	String	17D29a
Localization native development re...	String	en
Executable file	String	PropertyInjector
Bundle identifier	String	ds.Brunbear.PropertyInjector
InfoDictionary version	String	6.0
Bundle name	String	PropertyInjector
Bundle OS Type code	String	KEXT
Bundle versions string, short	String	1.1
CFBundleSupportedPlatforms	Array	(1 item)
Bundle version	String	1
DTCompiler	String	com.apple.compilers.llvm.clang.1_0
DTPlatformBuild	String	9C40b
DTPlatformVersion	String	GM
DTSDKBuild	String	17C76
DTSDKName	String	macosx10.13
DTXcode	String	0920
DTXcodeBuild	String	9C40b
Copyright (human-readable)	String	Copyright © 2017 Brunbear. All rights reserved.
IOKitPersonalities	Dictionary	(1 item)
AUDIO	Dictionary	(5 items)
CFBundleIdentifier	String	ds.Brunbear.PropertyInjector
IOClass	String	PropertyInjector
IOCPPrimaryMatch	String	PropertyInjector
IOProviderClass	String	IOPCIDevice
Properties	Dictionary	(0 items)
OSBundleLibraries	Dictionary	(5 items)

Unter dem Eintrag mit dem Pfeil (Properties), kann mittels des Pluszeichen neue Zeilen erzeugt werden, welche wir mit unseren Beispiel Werten füllen.

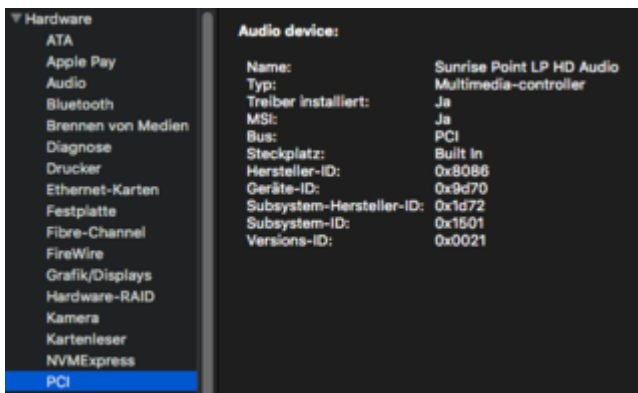
Key	Type	Value
Information Property List	Dictionary	(20 items)
BuiltMachineOSBuild	String	17D29a
Localization native development re...	String	en
Executable file	String	PropertyInjector
Bundle identifier	String	ds.Brunbear.PropertyInjector
InfoDictionary version	String	6.0
Bundle name	String	PropertyInjector
Bundle OS Type code	String	KEXT
Bundle versions string, short	String	1.1
CFBundleSupportedPlatforms	Array	(1 item)
Bundle version	String	1
DTCompiler	String	com.apple.compilers.llvm.clang.1_0
DTPlatformBuild	String	9C40b
DTPlatformVersion	String	GM
DTSDKBuild	String	17C76
DTSDKName	String	macosx10.13
DTXcode	String	0920
DTXcodeBuild	String	9C40b
Copyright (human-readable)	String	Copyright © 2017 Brunbear. All rights reserved.
IOKitPersonalities	Dictionary	(1 item)
AUDIO	Dictionary	(5 items)
CFBundleIdentifier	String	ds.Brunbear.PropertyInjector
IOClass	String	PropertyInjector
IOCPPrimaryMatch	String	0x3708066
IOProviderClass	String	IOPCIDevice
Properties	Dictionary	(0 items)
OSBundleLibraries	Dictionary	(5 items)

APL, slot-name, model, device_type und name haben dabei den Wert String und layout-id den Wert Data



Mit der `layout-id` und der Zeile Properties als Data, hat man nun auch Effekt das AppleAlc greift und den Ton ausgibt.

Dies ist dann nach speichern der Info.plist und nach einen Reboot unter Systembericht - PCI zusehen.



Vorlage des "leeren" Kextes befindet sich im Anhang.

Code

1. `<p align="right"> Zum Seitenanfang</p><hr>`

Beitrag von „Noir0SX“ vom 24. April 2019, 15:27

DevPro

- Platzhalter -

Code

1. `<p align="right"> Zum Seitenanfang</p><hr>`

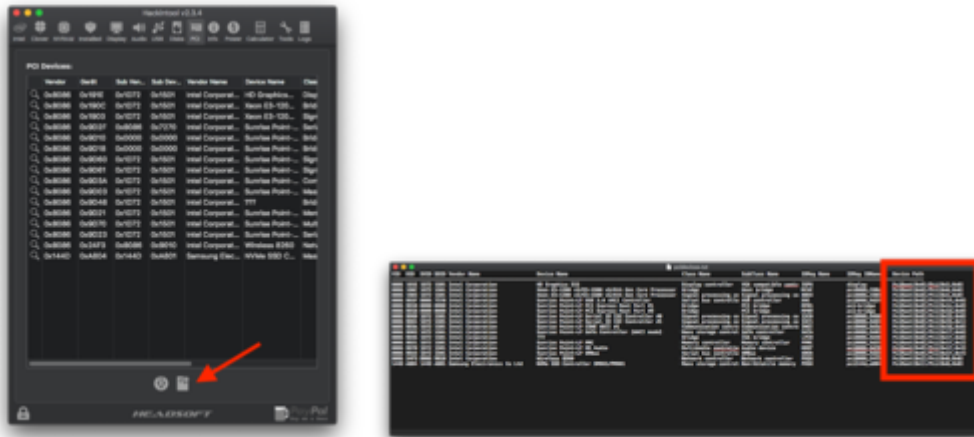
Beitrag von „Noir0SX“ vom 24. April 2019, 15:28

OpenCore Properties

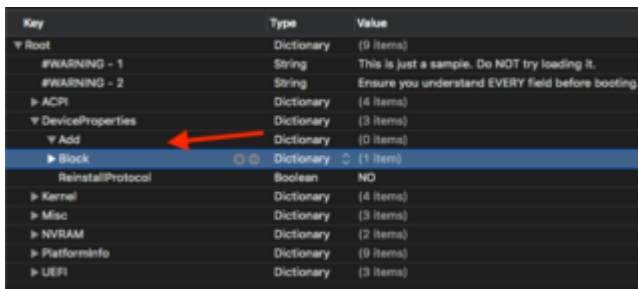
OpenCore steht noch am Anfang und kann daher auch anders sein als der Zeitpunkt diese Anleitung

Die benötigten Werte kann man ganz leicht mit dem [Hackintool \(ehemals Intel FB-Patcher\)](#) von [Headkaze](#) ermitteln.

Dazu rufen wir im Tool den Menüpunkt PCI auf und gehen auf Export. So wird eine Textdatei mit den benötigten Werten abgelegt.



Diese Datei enthält in der letzten Spalte die DevicePfade `PciRoot(0x0)/Pci(0x1f,0x3)`, die wir in das Feld DeviceProperties - ADD von der sample.plist, später in config.plist umbenennen, eintragen können.



Unter diesem Eintrag kann mittels des Pluszeichens neue Zeilen erzeugt werden, welche wir mit unseren Beispiel Werten füllen.

Key	Type	Value
√ Root	Dictionary	(9 items)
#WARNING - 1	String	This is just a sample. Do NOT try loading it.
#WARNING - 2	String	Ensure you understand EVERY field before booting.
▸ ACPI	Dictionary	(4 items)
√ DeviceProperties	Dictionary	(3 items)
√ Add	Dictionary	(1 item)
√ PciRoot[0x0]/Pci[0x1f,0x3]	Dictionary	(0 items)
▸ Block	Dictionary	(1 item)
ReinstallProtocol	Boolean	NO
▸ Kernel	Dictionary	(4 items)
▸ Miac	Dictionary	(3 items)
▸ NVRAM	Dictionary	(2 items)
▸ PlatformInfo	Dictionary	(9 items)
▸ UEFI	Dictionary	(3 items)

AAPL,slot-name, model, device_type und name haben dabei den Wert String und layout-id den Wert Data

Code

1. <div class="warning notice">
2. Achtung:

3. Die Werte sollten am Ende alle auf Data umgestellt werden und nicht wie davor geschrieben als String vorliegen</div>

√ Root	Dictionary	(9 items)
#WARNING - 1	String	This is just a sample. Do NOT try loading it.
#WARNING - 2	String	Ensure you understand EVERY field before booting.
▸ ACPI	Dictionary	(4 items)
√ DeviceProperties	Dictionary	(3 items)
√ Add	Dictionary	(1 item)
√ PciRoot[0x0]/Pci[0x1f,0x3]	Dictionary	(5 items)
AAPL,slot-name	String	Built In
model	String	Audio device
device_type	String	Multimedia-controller
name	String	Sunrise Point LP HD Audio
layout-id	Data	<09000000>
▸ Block	Dictionary	(1 item)
ReinstallProtocol	Boolean	NO
▸ Kernel	Dictionary	(4 items)
▸ Miac	Dictionary	(3 items)
▸ NVRAM	Dictionary	(2 items)
▸ PlatformInfo	Dictionary	(9 items)
▸ UEFI	Dictionary	(3 items)

Mit der layout-id und in der Zeile als Data, hat man nun auch Effekt das AppleAlc greift und den Ton ausgibt.

Dies ist dann nach speichern der config.plist und nach einen Reboot unter Systembericht - PCI zusehen.



Sample.plist Stand 19.05.2019

Code

1. `<p align="right"> Zum Seitenanfang</p><hr>`

Beitrag von „Noir0SX“ vom 28. April 2019, 20:15

- Platzhalter -

Beitrag von „Noir0SX“ vom 12. Oktober 2019, 11:24

Um diesen Thread erweitern zu können wurden die Kommentare und Diskussionen hier her verschoben: [Verschieden Methoden um Geräte zu den Systeminformationen - PCI hinzuzufügen \(Hilfe und Diskussionen\)](#) bitte hier nicht mehr antworten, Danke