

Erledigt

Sammelthread für IOREG-Befehle -> Welche kennt ihr denn noch?

Beitrag von „al6042“ vom 20. März 2018, 00:41

Hallöchen,

viele von Euch haben schon mal vom IORegistryExplorer gehört, der in seiner ganzen Pracht eine IOREG-Datei mit den wirklich spannenden Daten Eures Systems erstellen und im Unterstützungsfall sehr viele sachdienliche Hinweise aufzeigen kann.

Das geht aber auch ohne das Programm und zwar über den Terminal-Befehl `ioreg`.
Die Hilfe im Terminal gibt uns folgende Infos zu dem Befehl:

Code

1. usage: `ioreg [-abfilrtx] [-c class] [-d depth] [-k key] [-n name] [-p plane] [-w width]`
2. where options are:
3. `-a` archive output
4. `-b` show object name in bold
5. `-c` list properties of objects with the given class
6. `-d` limit tree to the given depth
7. `-f` enable smart formatting
8. `-i` show object inheritance
9. `-k` list properties of objects with the given key
10. `-l` list properties of all objects
11. `-n` list properties of objects with the given name
12. `-p` traverse registry over the given plane (IOService is default)
13. `-r` show subtrees rooted by the given criteria
14. `-t` show location of each subtree
15. `-w` clip output to the given line width (0 is unlimited)
16. `-x` show data and numbers as hexadecimal

Alles anzeigen

Wenn man den einfach so im Terminal ausführt, ohne zusätzlichen Parameter, wird man sofort

mit einer Vielzahl von Daten überschüttet.

Da wir aber meist wissen, was wir suchen *räusper* geht das auch mit gezielten Kombinationen aus Parametern.

Der erste Ansatz ist schon mal die Unterscheidung der Bereiche `IOService`, `IODeviceTree` und `IOACPIPlane`

IOService zeigt die mit Kext versorgten Geräte im System und deren eingestellten Parameter an.

IODeviceTree zeigt zusätzlich spannende Hardware-Eigenschaften, wie Hersteller- und Geräte-ID (vendor-id, device-id)

IOACPIPlane gibt zusätzlich ACPI-Details der einzelnen Geräte an.

Hier mal ein Beispiel für die Broadcom BCM4360 in meinem Kabylake-System.

Der Befehl für die Abfrage im Bereich `IODeviceTree` sieht wie folgt aus:

Code

```
1. ioreg -p IODeviceTree -n ARPT -r
```

"-p" gibt den Startpunkt für die Abfrage an - in diesem Falle "IODeviceTree"

"-n" gibt die Details des nachfolgenden Gerätenamens aus - in diesem Falle "ARPT"

"-r" gibt auch verzweigte Unterinfos aus...

Das Ergebnis könnt Ihr wiederum hier bestaunen:

Spoiler anzeigen

Vergleicht mal in dem Ergebnis die Werte für "vendor-id" und "device-id" zum Wert von "IName"... 😊

Und hier der Befehl für die Abfrage zur Broadcom BCM4360 unter "IOService":

Code

1. `ioreg -f -p IOService -n ARPT -r`

Und passend dazu natürlich auch das Ergebnis:

Spoiler anzeigen

Wie Ihr unschwer erkennen könnt, wird hier der "IODeviceTree"-Anteil und die Infos zum eingesetzten Kext (AirPort_BrcmNIC) aufgeführt.

Ihr werdet Euch nun fragen "Und warum jetzt dieser Thread? Soll er uns doch alle möglichen Kombinationen des Befehls hier einfach auflisten..." 😊

NeNeNe... Ich möchte von Euch wissen, welche spannenden Infos Ihr mit dem Befehl aus dem Terminal locken könnt... 😄

Was nicht erwünscht ist, sind Abfragen, welche die SMBIOS- oder NVRAM-Daten des Systems ermitteln und präsentieren würden.

Na dann... Los geht's und viel Spaß beim Tüfteln...

Beitrag von „al6042“ vom 20. März 2018, 00:46

Erster!!! 😄

Befehl zum Prüfen des Inhalts von "HDEF", dem internen AudioCodec:

Code

1. `ioreg -p IODeviceTree -n HDEF -r`

Ergebnis:

Spoiler anzeigen

Übrigens -> Man kann im Terminal mit dem "grep"-Befehl die Ausgabe auf einzelne Objekte einschränken:

Befehl:

Code

1. `ioreg -p IODeviceTree -n IGPU | grep hda-gfx`

Ergebnis:

Code

1. `al6042-iMac:~ al6042$ ioreg -p IODeviceTree -n IGPU | grep hda-gfx`
2. `|| "hda-gfx" = <"onboard-1">`

Befehl:

Code

1. `ioreg -p IODeviceTree -n GFX0 | grep hda-gfx`

Ergebnis:

Code

1. `al6042-iMac:~ al6042$ ioreg -p IODeviceTree -n GFX0 | grep hda-gfx`
2. `|||| "hda-gfx" = <"onboard-2">`

Beitrag von „derHackfan“ vom 20. März 2018, 09:38

Auch wenn ich kein Freund von IORegistryExplorer bin, hier einige weitere Terminal Befehle.

Code

1. `ioreg -l -w0 -p IODeviceTree > derHackfan.ioreg.txt`

Spoiler anzeigen

Code

1. `ioreg -l -w0 > ioreg.txt`

Spoiler anzeigen

Code

1. `ioreg -p IODeviceTree -n HDEF | grep layout-id`

Spoiler anzeigen

Beitrag von „Noir0SX“ vom 23. März 2018, 14:59

Batterie-Infos auslesen

```
ioreg -l | grep Capacity
```

Beitrag von „raymccoy“ vom 10. April 2020, 08:25

Hallo, wie kann ich mit ioreg herausfinden ob eine Kamera erkannt wird. Kann ich nach Kamera suchen? Was ist der beste Weg festzustellen ob eine Kamera erkannt wird?

Beitrag von „JimSalabim“ vom 12. Juli 2022, 17:19

[al6042](#) Weißt du vielleicht, wie ich mir über ioreg anzeigen lassen, ob mein Apple TV als Display mit dem Mac verbunden ist?

Ich würde über EventScripts gerne nen Befehl auslösen, sobald das Apple TV verbunden ist.

Das sähe dann auszugsweise so aus:

```
if "Apple TV" is in (do shell script "ioreg -l | grep 'XYXYXYXYXY?") then
tell application "Shortcuts" to run shortcut "Fernseher anschalten"
end if
```

Mir fehlt jetzt eigentlich nur der Teil hinter "grep". 😊

Beitrag von „al6042“ vom 12. Juli 2022, 18:00

Dafür müsstest du mit dem IORegistryExplorer mal schauen, wo sich in der IOReg ein Eintrag für den "Apple TV" zu finden ist, wenn die Verbindung hergestellt ist.

Damit kannst du den jeweiligen IODeviceTree eingrenzen und den richtigen "grep"-String identifizieren.

Da ich keinen Apple TV habe, kann ich leider nicht selbst nachschauen.

Beitrag von „JimSalabim“ vom 12. Juli 2022, 20:41

Danke dir! Leider taucht das Apple TV im IORegistryExplorer überhaupt nicht auf. Liegt wohl daran, dass es ein AirPlay-Gerät ist.

Wobei ich hier nicht ganz durchsteige. In den Systeminformationen ist es jedenfalls zu sehen.

Apple M1 Max:

Chipsatz-Modell:	Apple M1 Max
Typ:	GPU
Bus:	Integriert
Gesamtanzahl der Kerne:	32
Hersteller:	Apple (0x106b)
Metal-Familie:	Unterstützt, Metal GPUFamily Apple 7

Displays:

Farb-LCD:

Displaytyp:	Integriertes Liquid Retina XDR-Display
Auflösung:	3024 x 1964 Retina
Hauptdisplay:	Ja
Synchronisierung:	Aus
Eingeschaltet:	Ja
Helligkeit automatisch anpassen:	Ja
Verbindungstyp:	Intern

Apple TV Matthias:

Auflösung:	1920 x 1080 (1080p FHD - Full High Definition)
UI sieht aus wie:	1920 x 1080 @ 63.00Hz
Framepuffertiefe:	24-Bit Farbe (ARGB8888)
Synchronisierung:	Aus
Verbindungstyp:	AirPlay
Virtuelles Gerät:	Ja

EDIT: Hurra, ich hab einen anderen Befehl gefunden, der mir das Apple TV anzeigt:

```
system_profiler SPDisplaysDataType
```

So bekomme ich es jetzt hin. 😊