

Erledigt

## Tool um Ozmosis-Parameter auszulesen/manipulieren

**Beitrag von „Sascha\_77“ vom 29. Februar 2016, 19:09**

Ich habe hier ein kleines Programm gebaut welches übersichtlich alle vorhandenen Ozmosis-Parameter ausliest und parallel dazu eine Default.plist mit diesen Werten auf dem Desktop anlegt (Dateiname hat eine Uhrzeit hinten dran damit nicht evtl. schon vorhandene Default.plist's überschrieben werden). Bitte schaut Euch das mal an und sagt mir ggf. wo ich noch was verändern/hinzufügen soll.

---

**Beitrag von „al6042“ vom 29. Februar 2016, 19:38**

Genial,

vielen Dank...

ich teste das Tool sobald ich von meinem Montags-Stammtisch zurück bin. 😊

---

**Beitrag von „YogiBear“ vom 29. Februar 2016, 19:42**

[@al6042](#) Gehst du etwa der Methalle fremd? 😊

[@Sascha\\_77](#) Klasse! und Danke schon mal. Meinst du es würde auch mit der 167X-MASS funktionieren?

---

**Beitrag von „Sascha\_77“ vom 29. Februar 2016, 19:51**

Test es einfach mal an. Kann ja nix passieren. Habe mir jetzt die anderen Ozmosisvorlagen noch nicht angeschaut ob diese Zahlenwurst identisch ist.

Klar aber kann es auch für die andere Version machen.

EDIT:

Grad mal in die XMas geschaut ... ist identisch. Sollte also bei Dir klappen.

---

### **Beitrag von „al6042“ vom 29. Februar 2016, 19:54**

Wat mut dat mut... 😏

---

### **Beitrag von „griven“ vom 29. Februar 2016, 21:32**

Sehr coole Sache 😄

Das mit der GUI wäre natürlich richtig fein und noch feiner wäre es wenn es einen Export in die defaults.plist zulassen würde, das wäre ein Traum. Aber auch so schon wirklich gut. Nimm bitte im Breereich SMBIOS noch die beiden Keys HardwareAdress und HardwareSignature mit auf sind meiner Meinung nach nicht so ganz unwichtig mit Blick auf Dienste wie iMessage und FaceTime 😄

---

### **Beitrag von „Sascha\_77“ vom 29. Februar 2016, 21:45**

Mit der .plist kann ich mal schauen wenn das andere soweit läuft.

Hardwareadress und Signature sind im Script schon vorhanden. Habe sie aber auskommentiert, da ich bei diesen beiden Werten bei mir nichts auslesen konnte und eine

Fehlermeldung bekam.

Habe aber die Datei jetzt aktualisiert mit Address und Signature.

---

### **Beitrag von „al6042“ vom 29. Februar 2016, 21:52**

Respekt...

das sieht doch schon mal richtig spannend aus.

Bei mir wird aber der Wert für "BiosVersion" nicht angezeigt.

Sind die Werte jetzt eigentlich aus dem NVRAM oder aus der Defaults.plist ausgelesen?

---

### **Beitrag von „Sascha\_77“ vom 29. Februar 2016, 22:00**

Bei BiosVersion zeigt er mir bei mir auch nichts an.

Er liest das aus dem NVRam.

Habe gerade noch was ergänzt. Wenn ein Wert nicht gelesen werden kann kommt jetzt keine Fehlermeldung mehr. So "müllt" er das Fenster nicht zu.

Wenn du

```
nvram 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BiosVersion
```

im Terminal ausführst kommt dann ein Wert zurück?

EDIT:

Hat sich erledigt. Ich hatte im Script einen Buchstaben vergessen. Datei ist aktualisiert. Sollte nun klappen.

---

### **Beitrag von „griven“ vom 29. Februar 2016, 22:07**

HardwareAdress und Signature funktionieren prima ist echt cool. Der Wert für CSR-Active-Config sieht komisch aus gibt bei mir "g" zurück was komisch aussieht...

---

### **Beitrag von „Sascha\_77“ vom 29. Februar 2016, 22:08**

Bei mir siehts so aus:

CSR-Active-Config : g%00%00%00

Ist das nicht richtig? Habe diesen wert mit nvrnm -p ausgelesen ohne so eine Zahlenwurst davor.

---

### **Beitrag von „griven“ vom 29. Februar 2016, 22:09**

bei mir isses halt nur ein g also so:

Code

1. CSR-Active-Config : g
2. Boot-Args : nvda\_drv=1 -xcpm

die Boot-Args passen aber 😊

---

## Beitrag von „al6042“ vom 29. Februar 2016, 22:10

Bei einem manuell Test wird der Wert geliefert:

Code

1. Al6042-iMac:~ al6042\$ nvram 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BiosVersion
2. 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BiosVersion  
IM142.88Z.0118.B00.1309031249

Eventuell ein Zahlendreher bei der GUID oder ein Buchstabendreher beim Variablennamen....



Wenn da jetzt noch die Werte der GUID 1F8E0C02-58A9-4E34-AE22-2B63745FA101 drinnen wären, ist das Teil komplett... aber da gibt es viele Angaben, die grundlegend mal nicht gesetzt sind, was aber auch nicht heisst, dass sie nicht aktiv wären... z.B.:

Code

1. nvram 1F8E0C02-58A9-4E34-AE22-2B63745FA101:DisableNvidiaInjection

bringt mit einer unbearbeiteten Defaults.plist unter 167X im den

Code

1. (iokit/common) data was not found

obwohl die Variable aus dem ROM heraus schon als "%00" gesetzt ist.  
Deswegen musste ich nach dem Flashen explizit den Wert in der Defaults.plist als "%01"  
eintragen, sodass beide Monitore genutzt werden konnten.

EDIT:

Hier mal der gesamte Datenbestand:

Code

1. BaseBoardAssetTag : Base Board Asset Tag#
2. BaseBoardSerial : xxxxxxxxxxxxA1B2C
3. BiosDate : 03/09/13
4. BiosVersion :
5. ChassisAssetTag : iMac-Aluminum
6. EnclosureType : %0d
7. FirmwareFeatures : %17%14%00%80
8. FirmwareFeaturesMask : 7%ff%03%c0
9. FirmwareRevision : %0a%00%01%00
10. FirmwareVendor : Apple
11. HardwareAddress : xx:xx:xx:xx:xx:xx
12. HardwareSignature : 139BAD96-24CE-5396-A6AB-F4EC38EBD274
13. Manufacturer : Apple Inc.
14. ProcessorSerial : xxxxxxxxxxxxx
15. ProductFamily : iMac
16. ProductId : Mac-27ADBB7B4CEE8E61
17. ProductName : iMac14,2
18. SystemSKU : System SKU#
19. SystemSerial : xxxxxxxxxxxxx
20. SystemVersion : 1.0
21. CSR-Active-Config : g%00%00%00
22. Boot-Args : darkwake=8

Alles anzeigen

---

**Beitrag von „griven“ vom 29. Februar 2016, 22:12**

Der %00 Wert heißt ja kein Wert also Variable nicht gesetzt wenn ich das richtig verstehe und in der OZ Logik somit kein Wert gesetzt also injection aktiv 😄

---

## Beitrag von „Sascha\_77“ vom 29. Februar 2016, 22:17

### [Zitat von griven](#)

bei mir isses halt nur ein g also so:

Code

1. CSR-Active-Config : g
2. Boot-Args : nvda\_drv=1 -xcpm

die Boot-Args passen aber 😄

Ja du hast Recht. Bei der Fehlerausgabenunterdrückung war ich etwas zu wenig. Habs korrigiert. Müsste jetzt gehen. 😄

---

## Beitrag von „al6042“ vom 29. Februar 2016, 22:18

Nope...

der %00 heisst NO, während der %01 YES lautet:

▼ Defaults:1F8E0C02-58A9-4E34-AE22-2960745FA101	Dictionary	9 key/value pairs
AAPLig-platform-id	Number	⌵ 220.332.035
BootEntryTemplate	String	⌵ \$label
Darwin-DiskTemplate	String	⌵ \$label
DarwinRecoveryDiskTemplate	String	⌵ \$label
DisableBootEntriesFilter	Boolean	⌵ YES
DisableIntelInjection	Boolean	⌵ NO
DisableNvidiaInjection	Boolean	⌵ YES
TimeOut	Number	⌵ 5
UserInterface	Boolean	⌵ YES

## Code

```
1. <key>Defaults:1F8E0C02-58A9-4E34-AE22-2B63745FA101</key>
2. <dict>
3. <key>AAPL,ig-platform-id</key>
4. <integer>220332035</integer>
5. <key>BootEntryTemplate</key>
6. <string>$label</string>
7. <key>DarwinDiskTemplate</key>
8. <string>$label</string>
9. <key>DarwinRecoveryDiskTemplate</key>
10. <string>$label</string>
11. <key>DisableBootEntriesFilter</key>
12. <true/>
13. <key>DisableIntelInjection</key>
14. <false/>
15. <key>DisableNvidiaInjection</key>
16. <true/>
17. <key>TimeOut</key>
18. <integer>5</integer>
19. <key>UserInterface</key>
20. <true/>
21. </dict>
```

Alles anzeigen

## Code

```
1. Al6042-iMac:~          al6042$          nvram          1F8E0C02-58A9-4E34-AE22-
   2B63745FA101:DisableBootEntriesFilter
2. 1F8E0C02-58A9-4E34-AE22-2B63745FA101:DisableBootEntriesFilter %01
3. Al6042-iMac:~          al6042$          nvram          1F8E0C02-58A9-4E34-AE22-
   2B63745FA101:DisableIntelInjection
4. 1F8E0C02-58A9-4E34-AE22-2B63745FA101:DisableIntelInjection %00
5. Al6042-iMac:~          al6042$          nvram          1F8E0C02-58A9-4E34-AE22-
   2B63745FA101:DisableNvidiaInjection
6. 1F8E0C02-58A9-4E34-AE22-2B63745FA101:DisableNvidiaInjection %01
7. Al6042-iMac:~ al6042$ nvram 1F8E0C02-58A9-4E34-AE22-2B63745FA101:UserInterface
8. 1F8E0C02-58A9-4E34-AE22-2B63745FA101:UserInterface %01
```



## Beitrag von „griven“ vom 29. Februar 2016, 22:20

Schon richtig ist ja boolean aber trotzdem führt no in dem Fall zur der bekannten Ausgabe (iokit/common) data was not found sprich das boolean No entspricht Variable nich vorhanden. True and False passt auf die defaults.plist...

---

## Beitrag von „Sascha\_77“ vom 29. Februar 2016, 22:25

[@al6042](#)

Wenn Du mir die Werte nennen kannst die bei GUID 1F8E0C02-58A9-4E34-AE22-2B63745FA101 drinstehen kann ich die ergänzen. Die gehen leider nicht aus den Ozmosis-Templates hervor.

---

## Beitrag von „al6042“ vom 29. Februar 2016, 22:26

Das heisst wenn der Fehler

Code

1. (iokit/common) data was not found

bei einer der Abfragen auftaucht ist das gleichbedeutend mit dem Wert "%00" oder dem Begriff "NO"?

EDIT:

Hier sind alle Werte der GUID für 1479 und 1669:

Code

1. Default Settings 1F8E0C02-58A9-4E34-AE22-2B63745FA101

2. =====
- 3.
- 4.
5. 'Key' || 'Value' || 'Note/Example'
6. =====||=====||=====
7. AcpiLoaderMode || INTEGER || Control ACPI Loader
8. UserInterface || BOOLEAN || Display User Interface/GUI
9. TimeOut || INTEGER || Time-out In Seconds
10. DisableAtiInjection || BOOLEAN || De/activate ATI GFX device property injection
11. AtiFramebuffer || STRING || Name of the specific ATI framebuffer
12. DisableNvidiaInjection || BOOLEAN || De/activate Nvidia GFX device property injection
13. DisableIntelInjection || BOOLEAN || De/activate Intel GFX device property injection
14. DisableVoodooHda || BOOLEAN || Disables loading VoodooHDA from firmware
15. EnableVoodooHdaInternalSpdif || BOOLEAN || Enable/Disable onboard S/PDIF header when using VoodooHDA from firmware
16. DisableBootEntriesFilter || BOOLEAN || Disables filtering of firmware generated boot entries
17. AAPL,snb\_platform\_id || INTEGER ||
18. AAPL,ig-platform-id || INTEGER ||
19. BootEntryTemplate || STRING || \$label \$guid
20. DarwinDiskTemplate || STRING || \$label \$platform \$major \$minor \$build
21. DarwinRecoveryDiskTemplate || STRING || \$label \$platform \$major \$minor \$build[code]

Alles anzeigen

Ab 167X gibt es von folgende Variable:

Code

1. SkipSmbusSpdScan || BOOLEAN || This will ignore memory SPD data and use instead original Type 17 SMBios tables

---

**Beitrag von „Sascha\_77“ vom 29. Februar 2016, 22:58**

So alle Werte von  
1F8E0C02-58A9-4E34-AE22-2B63745FA101  
ergänzt.

Feierabend für heute. 😊

---

### Beitrag von „griven“ vom 29. Februar 2016, 23:03

Läuft prima, sehr Geil danke Dir.

---

### Beitrag von „al6042“ vom 29. Februar 2016, 23:04

Genial... 😊

Vielen herzlichen Dank...

Hier mal das Ergebnis:

Code

1. Read Ozmosis-Parameters. Script by Sascha\_77
- 2.
- 3.
4. ----- Start -----
5. BaseBoardAssetTag : Base Board Asset Tag#
6. BaseBoardSerial : xxxxxxxxxxxxA1B2C
7. BiosDate : 03/09/13
8. BiosVersion : IM142.88Z.0118.B00.1309031249
9. ChassisAssetTag : iMac-Aluminum
10. EnclosureType : %0d
11. FirmwareFeatures : %17%14%00%80
12. FirmwareFeaturesMask : 7%ff%03%c0
13. FirmwareRevision : %0a%00%01%00
14. FirmwareVendor : Apple
15. HardwareAddress : xx:xx:xx:xx:xx:xx

16. HardwareSignature : 139BAD96-24CE-5396-A6AB-F4EC38EBD274
17. Manufacturer : Apple Inc.
18. ProcessorSerial : xxxxxxxxxxxxxx
19. ProductFamily : iMac
20. ProductId : Mac-27ADBB7B4CEE8E61
21. ProductName : iMac14,2
22. SystemSKU : System SKU#
23. SystemSerial : xxxxxxxxxxxxxx
24. SystemVersion : 1.0
25. CSR-Active-Config : g%00%00%00
26. Boot-Args : darkwake=8
27. -----
28. AcpiLoaderMode :
29. UserInterface : %01
30. TimeOut : %05
31. DisableAtiInjection :
32. AtiFramebuffer :
33. DisableNvidiaInjection:
34. DisableIntelInjection: %00
35. DisableVoodooHda :
36. EnableVoHdaInt.Spdif :
37. Disa.BootEntr.Filter : %01
38. AAPL,snb\_platform\_id :
39. AAPL,ig-platform-id : %03%00"%0d
40. BootEntryTemplate : \$label
41. DarwinDiskTemplate : \$label
42. DarwinRecov.DiskTemp.: \$label
43. ----- End -----

Alles anzeigen

Ich bin schwer beeindruckt und würde dich gerne, im Namen der Forumsteilnehmer als auch des Forums-Stabes, bitten, dieses tolle Tool in unseren Download-Bereich aufnehmen zu dürfen.

---

**Beitrag von „Sascha\_77“ vom 29. Februar 2016, 23:17**

Keine Ursache. Schön das es Euch gefällt. 😊

Klar könnt ihr das Tool dort einstellen.

Nächste Aktion wird dann das GUI sein.

---

### **Beitrag von „al6042“ vom 29. Februar 2016, 23:21**

Dann bedanke ich mich recht herzlich und stelle es gleich bereit... 😊

---

### **Beitrag von „griven“ vom 29. Februar 2016, 23:22**

Sehr gut das ist ne ziemlich nützliche Sache für alle User die OZ updaten wollen und bisher nicht mit einer defaults.plist arbeiten.

---

### **Beitrag von „al6042“ vom 29. Februar 2016, 23:34**

Vor allem kann man bei frischen Flash-Vorgängen den Usern gezielt klarmachen warum Dinge nicht gleich so tun, wie sie gerne erwarten. 😊

---

### **Beitrag von „griven“ vom 1. März 2016, 01:15**

Ich habe noch einen Fehler gefunden...

DisableNvidalInjection ist falsch bzw. steht falsch in der Doku es muss richtigerweise heißen DisableNvidialInjection

---

### **Beitrag von „mhaeuser“ vom 1. März 2016, 01:32**

So hieß es auch bis 1669 oder X-MASS (Schreibfehler in Oz, nicht der Doku).

---

### **Beitrag von „al6042“ vom 1. März 2016, 06:34**

Immer diese Copy/Paste-Fehler... niemandem darf man trauen... 😏  
Habe es in meiner Auflistung der Variablen geändert.

---

### **Beitrag von „Adnarel“ vom 1. März 2016, 13:15**

Schickes kleines Skript!  
So etwas will ich auch noch lernen bauen zu können 😊

Danke!

---

### **Beitrag von „Sascha\_77“ vom 1. März 2016, 18:40**

Da hier gerade auf der Arbeit nix los ist bin ich gerade dabei ein Export-Skript zu basteln. Heisst man kann die Werte die man ausgelesen hat direkt in eine fertige formatierte Default.plist speichern lassen. Hatte [@griven](#) ja zuvor schon angeregt.

Mal ne Frage. Kann man anstatt:

Code

1. `<key>csr-active-config</key>`
2. `<integer></integer>`

auch

Code

1. `<key>csr-active-config</key>`
2. `<string></string>`

nehmen?

Hier ist jetzt eine plist wie sie mein script erzeugt hat. Variablen sind jetzt aber leer da ich hier nicht an einem Hack sitze sondern vorm schnöden Windows. Ist das soweit ok? Und was ist mit Keys wie "AtiFramebuffer" die nicht im Template stehen aber mit in der GUID enthalten sind. Soll ich die alle pauschal noch ergänzen mit dann ggf. leeren Werten? Ansonsten könnte ich auch über eine IF-Abfrage prüfen lassen ob Werte enthalten sind und sie abhängig davon in die Default.plist schreiben lassen oder eben gar nicht. Also was ist besser. Keys mit leeren Strings/Integer oder die Keys tauchen gar nicht erst auf?

Spoiler anzeigen

EDIT:

Ich habe jetzt das Tool aktualisiert. Es erzeugt jetzt eine Default.plist auf dem Desktop.

---

### **Beitrag von „al6042“ vom 1. März 2016, 18:43**

Ich glaube nicht. dass das funktionieren wird.

Der Original Eintrag in der Defaults.plist ist ein DATA-Wert und sieht als XML so aus:

Code

1. <key>csr-active-config</key>
2. <data>
3. ZwAAAA==
4. </data>

---

### Beitrag von „nikoZ“ vom 1. März 2016, 18:57

It's this supposed to be run into real Macs for nvram check or it's only for Ozmosis parameters?  
[@Sascha\\_77](#) You did a great job!

---

### Beitrag von „al6042“ vom 1. März 2016, 19:07

That's a good question... too bad I don't have a real Mac flying around to test it... 😊  
But I checked it on my HP Laptop with Clover and it gave me the csr-active-config and the boot-args back.  
Everything else was blank, which means that Clover doesn't inject the SMBIOS infos into the NVRAM, but somewhere completely different... 😊

---

### Beitrag von „Sascha\_77“ vom 1. März 2016, 19:13

@al

Ich kriege den Wert aber nicht mit der GUID ausgelesen. Wirft dann einen Fehler aus. Muss ihn dann mit nvram -p ausgeben. Und da hat er halt nur dieses Format. Irgend eine Idee was man da machen kann?

EDIT: GUI in Progress. 😊



---

### Beitrag von „griven“ vom 1. März 2016, 21:12

Gute Frage vielleicht weiß [@Download-Fritz](#) da ne Antwort zu 😊

---

### Beitrag von „mhaeuser“ vom 1. März 2016, 21:38

Keine Ahnung wie diese Skripts funktionieren, aber %-Prefix heißt Hex... also müsstest du die zwei Stellen nach dem % als Hex parsen und so nutzen...

---

### Beitrag von „Sascha\_77“ vom 1. März 2016, 22:10

Hm irgendwie habe ich da ein kleines Verständninsproblem.

Um das Beispiel von al6042 aufzugreifen:

Dieser Wert muss in der XML stehen:

ZwAAAA==

Diesen Wert kriege ich (soll doch dem von da oben drüber entsprechen?)

g%00%00%00

von "nvram -p" angegeben. Mit der GUID kriege ich immer einen Fehler. Daher habe ich das mit nvram -p gemacht. 😞

---

## Beitrag von „al6042“ vom 1. März 2016, 22:13

Hex kann ja auch nicht gehen... da geht ja nur 1 bis F... und Gleichheitszeichen habe dort auch noch nicht gesehen... 😊

---

## Beitrag von „al6042“ vom 1. März 2016, 22:20

Das Data-Format ist ein Base64Binary.

Per Decode mit Hilfe der Seite <http://www.freeformatter.com/base64-encoder.html>

kommt bei Eingabe von "ZwAAAA==" der Decoded String "g" raus...

Das würde dann passen... 😊

---

## Beitrag von „Sascha\_77“ vom 1. März 2016, 22:22

Eben. Aber wie komme ich nun zum

ZwAAAA==

? Hat man diesen Wert wenn man über die GUID ausliest?

Code

1. nvram 7C436110-AB2A-4BBB-A880-FE41995C9F82:csr-active-config

Wie gesagt ... bei mir gibts einen Fehler daher frage ich jetzt.

---

## Beitrag von „griven“ vom 1. März 2016, 22:24

Nein, dass ist der Wert den man in den Defaults setzt. Der Wert lässt sich über die defaults entweder als Integer setzen oder eben als data Objekt ich nehme mal an, dass der integer Wert in das Dataobjekt umgewandelt wird bevor er injected wird.

---

## Beitrag von „mhaeuser“ vom 1. März 2016, 22:27

in g%00%00%00 ist also ein =? OK, alles klar...

Wie wär's, wenn man einfach nicht data, sondern integer nimmt?

---

## Beitrag von „al6042“ vom 1. März 2016, 22:28

Wenn du die NVRAM Abfrage meinst. sieht das Ergebnis so aus:

Code

1. nvram -p
  2. fmm-computer-name Al6042-iMac
  3. bluetoothActiveControllerInfo Z%06%0d%05%00%00%00 %16%1d%\%f3paZI
  4. SystemAudioVolumeDB %f0
  5. SystemAudioVolume A
  6. LocationServicesEnabled %01
  7. csr-active-config g%00%00%00
  8. boot-args darkwake=8
  9. EFIBluetoothDelay %b8%0b
- 

## Beitrag von „Sascha\_77“ vom 1. März 2016, 22:40

Genau das meine ich. Aber irgendwie peil ichs nicht wie ich von diesem Wert der bei csr-active-

config ausgespuckt wird so wandeln kann das es in die Default.plist geschrieben werden kann. Mit der Hex-Umrechnung komme ich hier doch gar nicht weiter. Und mit dieser Binary64 Sache auch nicht.

Ich kann zwar in der Shell auch Base64 decoden:

Code

```
1. echo `echo ZwAAAA== | base64 --decode`
```

Da kommt dann als Ergebnis "g" raus. Aber was ist jetzt noch mit den 3 Zahlenwerten die hinterher kommen?

---

**Beitrag von „griven“ vom 1. März 2016, 22:46**

[@Download-Fritz](#) es geht hier im Moment ja darum den aktuell gesetzten Wert zu ermitteln und das geht nun mal nur über `nvrnm -p` ich schätze das liegt wohl daran, dass die Einträge die in der `defaults.plist` unter `Defaults:7C436110-AB2A-4BBB-A880-FE41995C9F82` angesiedelt sind nicht über diese Adresse aus dem NVRAM auslesbar sind und so eben nur die Möglichkeit bleibt die Ausgabe von `nvrnm -p` zu interpretieren. Hierbei scheint es einen Unterschied zu machen ob man den Wert für die `csr-active-config` als `data` oder `integer` übergibt. In meiner `defaults` sieht das so aus:

Code

1. `<key>csr-active-config</key>`
2. `<integer>103</integer>`

`nvrnm -p` gibt ein kleines `g` als Wert für die `csr-active-config` zurück

Code

1. `CSR-Active-Config : g`

Nimmt man den Typ `Data` bekommt man den Wert zurück den `AL` auch bekommt. Die Frage ist halt wie lässt sich von dem Wert auf die `config` schließen die gerade aktiv ist.

---

### Beitrag von „griven“ vom 1. März 2016, 22:53

Ha ich glaube ich habe es 😄

Wenn man decodiert erhält man in dem Beispiel nun ein `g` dieser wiederum in hex umgewandelt ergibt `67` was dann tatsächlich auch der `csr-active-config 0x67` oder eben [SIP](#) komplett aus entspricht 😄 Der Wert `0x67` hex entspricht dem Wert `103` dezimal et voila wir haben den Wert den wir für die `defaults` brauchen 😄

---

### Beitrag von „Sascha\_77“ vom 1. März 2016, 22:53

Ich kann folgendes machen ... wenn "`g`" dort steht soll er in die `Default.plist` den Wert `103` schreiben. Müsste man eigtl. auch wie du grad geschrieben hast `griven` in der Shell ausrechnen lassen. Aber dann kann man sich das ja eigtl. sparen. Was kann denn sonst noch so da stehen

anstatt dem "g" bei halb abgeschaltetem [SIP](#) z.b.? Wenn ich die verschiedenen Varianten weiss kann ich das easy einbinden.

---

### **Beitrag von „griven“ vom 1. März 2016, 23:01**

Klar das macht Sinn nur steht ja nicht immer g da sondern halt nur wenn die [SIP](#) komplett aus ist. Ich denke der Wert der da steht muss halt wirklich einmal in Hex umgewandelt werden und von da dann in Integer um den richtigen Wert für alle möglichen Einstellungen der [SIP](#) zu ermitteln. Umwandlung in Hex geht mit der Bash hier ein Beispiel

Code

```
1. echo -n 'text' | xxd -ps | sed 's/[[:xdigit:]]{2}\x&/g'
```

und von hex in dec dann so

Code

```
1. echo $((16#ff))
```

wobei das ff eben dem Anteil nach dem 0x entspräche.

---

### **Beitrag von „Adnarel“ vom 1. März 2016, 23:04**

Interessant finde ich ja, dass es umgewandelt wird.

Der NVRAM wird vom System interpretiert/ausgelesen beim booten?

Und die Default.plist von Ozmosis bei einem NVRAM reset ausgelesen und ins NVRAM geschrieben?

Letztlich muss eigentlich der Umrechnungsweg beschriftet werden zum schreiben in die default.plist, den griven oben beschrieben hat. Das würde dann auch mal für andere Werte gelten, wie [SIP](#) halb ausgeschaltet.

Oder liege ich da falsch?

Zu langsam 😊

---

### Beitrag von „al6042“ vom 1. März 2016, 23:13

Ist ja geil was hier abgeht...

Ich kriege hier noch einen Crash-Kurs im Programmieren... 😊

---

### Beitrag von „mhaeuser“ vom 1. März 2016, 23:13

Wenn man integer verwendet, muss man auch die Größe angeben: `<integer size=32>...</integer>`

Und ich weiß, worum es geht, deswegen hab' ich ja auch gesagt, dass nirgends Gleichzeichen vorkommen. 😊

---

### Beitrag von „Sascha\_77“ vom 1. März 2016, 23:17

Alles klar Griven ... habs mal bisschen angepasst und zusammengebastelt. 😄

Code

1. `hex=`echo -n 'g' | xxd -ps | sed 's/[[:xdigit:]]\{2\}/&/g``
2. `echo $((16#$hex))`

Wirft den Wert 103 aus. 🤖👉👈

---

### **Beitrag von „griven“ vom 1. März 2016, 23:18**

Right, darauf wollte ich hinaus 😄

---

### **Beitrag von „Sascha\_77“ vom 2. März 2016, 00:20**

Sö. Nun setzt er in der Default.plist den richtigen Wert. Datei ist aktualisiert. Nun sollte eigentlich alles passen.

Ich bin weg für heut. N8.

---

### **Beitrag von „griven“ vom 2. März 2016, 00:25**

Die im Startpost aber noch nicht. oder?  
Ich wünsche Dir ne gute Nacht 😄

---

### **Beitrag von „Sascha\_77“ vom 2. März 2016, 00:31**

Doch sie ist aktualisiert. 😊

---

### **Beitrag von „griven“ vom 2. März 2016, 01:31**

Hum entweder schlägt mein blonder Kern wieder durch oder ich bin ein Sonderfall \*fg\*  
Bei mir zeigt es weiter g an...



Code

1. SystemVersion : 1.0
2. CSR-Active-Config : g
3. Boot-Args : nvda\_drv=1 -xcpm

---

### **Beitrag von „Sascha\_77“ vom 2. März 2016, 06:43**

Im Infofenster schon. Aber in der plist die er auf dem Desktop anlegt sollte 103 stehen.

Habe es nun auch im Infofenster geändert.

---

### **Beitrag von „al6042“ vom 2. März 2016, 07:27**

Bei mir wird die 103 im Infofenster und in der Default\_07\_18\_04.plist angezeigt.

Ich habe auch mal schnell mein BeyondCompare angeschmissen um die Unterschiede zwischen der Plist auf dem Desktop und in der EFI-Partition darzustellen.

Der Report liegt als HTML File bei.

EDIT:

Die erzeugt Plist auf dem Desktop lässt sich nicht mit PlistEdit Pro öffnen.

---

### **Beitrag von „Sascha\_77“ vom 2. März 2016, 07:40**

Habs auch vorhin noch geändert. 😊

Ja irgendwo scheint noch was nicht zu stimmen. Konnte die plist in XCode auch nicht öffnen

gestern. Ich hatte schon mit der template datei verglichen und konnte keine Unterschiede feststellen. Ist auch als UTF-8 encoded wie es sein soll. Tappe da derzeit noch im Dunkeln. Muss man derzeit dann noch mit einem Texteditor öffnen.

Hat die Datei schon jemand beim booten getestet ob die so eingelesen wird?

EDIT:

Ah du hast den Bericht drangehangen. Guck ich mir nachher mal an.

---

### Beitrag von „al6042“ vom 2. März 2016, 07:52

Ich habe sie noch nicht getestet, da die GUID in Zeile 67 deiner Plist um ein Zeichen zu kurz ist:  
Ist:

Code

1. `<key>Defaults:C436110-AB2A-4BBB-A880-FE41995C9F82</key>`

Soll:

Code

1. `<key>Defaults:[u]7[/u]C436110-AB2A-4BBB-A880-FE41995C9F82</key>`

Ausserdem bin ich mir nicht sicher, ob die Integer-Werte in folgenden Variablen nicht doch eher dezimale Zahlen sein sollten:

Deine Plist:

Code

1. `<key>EnclosureType</key>`
2. `<integer>%0d</integer>`
3. `<key>FirmwareFeatures</key>`
4. `<integer>%17%14%00%80</integer>`
5. `<key>FirmwareFeaturesMask</key>`

6. `<integer>7%ff%03%c0</integer>`
7. `<key>FirmwareRevision</key>`
8. `<integer>%0a%00%01%00</integer>`

Meine Plist:

Code

1. `<key>EnclosureType</key>`
2. `<integer>13</integer>`
3. `<key>FirmwareFeatures</key>`
4. `<integer>2147488791</integer>`
5. `<key>FirmwareFeaturesMask</key>`
6. `<integer>3221487415</integer>`
7. `<key>FirmwareRevision</key>`
8. `<integer>65546</integer>`

Dann sind im Programm verschiedene Werte nicht richtig zurückgegeben worden:

Code

1. AcpiLoaderMode :
2. UserInterface : %01
3. TimeOut : %05
4. DisableAtiInjection :
5. AtiFramebuffer :
6. DisableNvidiaInjection:
7. DisableIntelInjection: %00
8. DisableVoodooHda :
9. EnableVoHdaInt.Spdif :
10. Disa.BootEntr.Filter : %01
11. AAPL,snb\_platform\_id :
12. AAPL,ig-platform-id : %03%00"%0d
13. BootEntryTemplate : \$label
14. DarwinDiskTemplate : \$label
15. DarwinRecov.DiskTemp.: \$label

Alles anzeigen

Die aber in meiner Plist drinnen stehen:

## Code

1. `<key>AAPL,ig-platform-id</key>`
2. `<integer>220332035</integer>`
3. `<key>BootEntryTemplate</key>`
4. `<string>$label</string>`
5. `<key>DarwinDiskTemplate</key>`
6. `<string>$label</string>`
7. `<key>DarwinRecoveryDiskTemplate</key>`
8. `<string>$label</string>`
9. `<key>DisableBootEntriesFilter</key>`
10. `<true/>`
11. `<key>DisableIntelInjection</key>`
12. `<false/>`
13. `<key>DisableNvidiaInjection</key>`
14. `<true/>`
15. `<key>TimeOut</key>`
16. `<integer>5</integer>`
17. `<key>UserInterface</key>`
18. `<true/>`

Alles anzeigen

Und in deiner Plist garnicht auftauchen:

## Code

1. `<key>BootEntryTemplate</key>`
2. `<string>$label</string>`
3. `<key>DarwinCoreStorageTemplate</key>`
4. `<string>$label</string>`
5. `<key>DarwinDiskTemplate</key>`
6. `<string>$label</string>`
7. `<key>DarwinRecoveryDiskTemplate</key>`
8. `<string>$label ($platform.$major.$minor)</string>`
9. `<key>LinuxDiskOptionTemplate</key>`
10. `<string>initrd=$initramfs root=$root ro quiet</string>`
11. `<key>LinuxDiskTemplate</key>`
12. `<string>Linux $build</string>`
13. `<key>LinuxRescueDiskTemplate</key>`
14. `<string>Linux Rescue</string>`
15. `<key>LinuxRescueOptionTemplate</key>`

16. `<string>initrd=$initramfs root=$root ro quiet</string>`

Alles anzeigen

---

## Beitrag von „Sascha\_77“ vom 2. März 2016, 08:27

Die nicht im Template waren sind auch noch nicht eingebunden. Muss ich noch machen.

Ok also das andere muss auch als dezimal geschrieben werden. Versuch ich dann mal wie mit dem "g".

EDIT:

Also den Enclosure Type kriege ich umgerechnet:

Code

1. `echo $((16#0d))`
2. 13

Aber FirmwareFeatures z.b. geht nicht.

Code

1. `echo $((16#17140080))`
2. 387186816

Wie muss ich denn die `%17%14%00%80` an das echo übergeben damit ich auf 2147488791 komme?

Hm wenn ich 2147488791 von Dez auf Hex umrechne komme ich auf 80001417. Die Blöcke von `%17%14%00%80` scheint man in umgekehrte Reihenfolge einlesen zu müssen. Also

%80%00%14%17. Was ne Logik. 🤔 Das wird jetzt ein schönes Gefummel.

EDIT2:

So gehts 😁

Code

1. echo "%17%14%00%80" | awk -F%' '{ print \$6"\$5"\$4"\$3"\$2"\$1 }'
2. 80001417

Kann mal bitte einer prüfen ob die 1. Zeile unter OSX das selbe Ergebnis auswirft? Kann ich hier nicht prüfen da ich auf einer Debian-Kiste drauf bin. Sollte zwar klappen aber ich hatte schon Fälle wo ein Bash-Script was unter Debian funktioniert nicht unter OSX lief.

---

## Beitrag von „al6042“ vom 2. März 2016, 09:13

Oh... Cool...

Das mit den umgedrehten HEX-Blöcken kenne ich aus der DSDT.

Aus der PlatformID einer HD4000 in einem Laptop wird 01660003 zu 0x03, 0x00, 0x66, 0x01:

Code

1. Store (Package (0x08))
2. {
3. "AAPL,slot-name",
4. "Built In",
5. "name",
6. "Display Controller",
7. "AAPL,ig-platform-id",
8. Buffer (0x04)
9. {
10. 0x03, 0x00, 0x66, 0x01
11. },
12. "hda-gfx",
13. Buffer (0x0A)
14. {

15. "onboard-1"
16. }
17. }, Local0)
18. DTGP (Arg0, Arg1, Arg2, Arg3, RefOf (Local0))
19. Return (Local0)

Alles anzeigen

---

### **Beitrag von „Buuhr“ vom 2. März 2016, 09:33**

So - wir haben einen Blöden in der Runde, der die Defaults.plist ausprobiert hat.

nun, alle OS X Medien werden nicht mehr im BIOS als bootbar angezeigt.

Weder das System, noch die Recovery oder die Installation-Sticks.

Hat jemand eine Idee was ich tun kann?

---

### **Beitrag von „Sascha\_77“ vom 2. März 2016, 09:40**

Etwas Kopfzerbrechen bereitet mir noch dieser hier:

Code

1. AAPL,ig-platform-id : %03%00"%0d

Da habe ich noch nicht raus wie ich auf 220332035 kommen soll. Da ist jetzt dieses " mit drin. Und wenn ich das einfach weglasse passt es auch nicht. Da komme ich auf 851971.

[@Buuhr](#)

Du hast doch hoffentlich das nur von einem Test-Stick aus gemacht?

---

### **Beitrag von „Buuhr“ vom 2. März 2016, 09:45**

Ein Testsystem schon

aber mir wird überhaupt nichts mehr mit Ozmosis angezeigt.

Wie gesagt, selbst mein install-Stick wird nicht erkannt

---

### **Beitrag von „Sascha\_77“ vom 2. März 2016, 15:46**

[@al6042](#)

Kannst Du mir sagen bei welchen Keys überall true bzw. false gesetzt wird?

Code

1. `<key>DisableBootEntriesFilter</key>`
2. `<true/>`
3. `<key>DisableIntelInjection</key>`
4. `<false/>`
5. `<key>DisableNvidiaInjection</key>`
6. `<true/>`
7. `<key>UserInterface</key>`
8. `<true/>`

Die hattest Du ja schon genannt. Gibts noch mehr?



EDIT:Habe jetzt die Parameter mir auf einer GitHub-Seite angeschaut. Ich denke ich habe jetzt alle mit drin. Alle Hexwerte werden nun als Dezimal ausgegeben. Im Infofenster sowie in der Default.plist. Zumindest theoretisch. 😊

Konnte es noch nicht live testen, da ich wie schonmal gesagt auf einer Debian-Kiste bin. Hab das alles per Trockenübung zusammengebaut. Datei ist bereits ausgetauscht und wartet auf ein Testkaninchen. 😊

Spoiler anzeigen

---

### **Beitrag von „al6042“ vom 2. März 2016, 15:47**

[@Sascha\\_77](#)

TRUE würde ich grundsätzlich bei "DisableBootEntriesFilter" setzen.

Die anderen drei Angaben sind ja in Bezug auf das vorhandene System zu setzen. Grundsätzlich würde ich alle "DisableBlahBlahInjection" auf TRUE stellen... lieber starte ich sie per Anpassung im Nachgang, falls ich sie brauche.

[@Buuhr](#)

Damit die Settings für Oznois, oder Ozmosis selbst, nicht beim Booten genutzt werden, musst du beim Einschalten des Rechners mehrfach die ESC-Taste drücken, bis der Bildschirm rot wird. Damit deaktivierst du Ozmosis für diesen Bootvorgang und solltest dann wenigstens mit einem Bootstick oder einer Linux-LiveCD starten können, um den Fehler zu beheben.

---

### **Beitrag von „Sascha\_77“ vom 2. März 2016, 15:49**

Ok dann setze ich die noch per Default auf True und die anderen lass ich auf False.

Achso ja ... was machen wir denn jetzt mit dem hier?

Code

1. AAPL,ig-platform-id : %03%00"%0d

---

### Beitrag von „al6042“ vom 2. März 2016, 16:02

Das kann ich dir erst sagen, wenn ich zuhause am OZM-Rechner bin... 😊

---

### Beitrag von „Sascha\_77“ vom 2. März 2016, 16:04

Hehe naja geht sich ja eher darum diesen Wert in eine Dezimalzahl zu wandeln. 😊 Mir ist nicht ganz schlüssig was so ein " da drin überhaupt zu suchen hat. Ist ja bei keinem anderen Key. Nur bei dem.

Es soll ja der Wert 220332035 rauskommen. Den wiederum in Hex umgewandelt gibt das:

Code

1. D220003

sieht aber etwas anders aus wie

Code

1. %03%00"%0d

Das scheint irgendwie so als wenn " die Zahl 22 darstellen soll. Wäre kein Problem das Umwandeln zu lassen. Ist nur die Frage ob auch noch andere Sonderzeichen ausser das " dort

stehen können.

---

### **Beitrag von „al6042“ vom 2. März 2016, 16:06**

Deswegen möchte ich nochmal schauen, wie die Ausgabe zuhause aussieht damit ich dir eventuell noch Tipps geben kann.

---

### **Beitrag von „Sascha\_77“ vom 2. März 2016, 21:54**

Ok. Soweit ich mich entsinne hat er das " bei meinem zu Hause aber auch gemacht. Da hatte ich mich schon etwas gewundert gehabt.

EDIT:

Habe das Tool nun auf meinem Rechner getestet. Hier funktioniert es einwandfrei. Habe keinen Fehler mehr entdeckt.

Habs hochgeladen.

Textwindow:

Spoiler anzeigen

Default.plist

Spoiler anzeigen

EDIT2:

Nun habe ich rausgefunden warum man die plist nicht mit XCode und dergleichen öffnen konnte. Ich habe bei ein paar Werten "string" und "key" vertauscht. Jetzt kann sie XCode öffnen. Übrigens gibt es ein kleines shelltool mit welchem man .plists auf Fehler untersuchen kann:

Code

1. plutil -lint DATEINAME

---

### **Beitrag von „al6042“ vom 2. März 2016, 21:55**

Super...

Habe es eben mal getestet und die dazu gehörige Plist auch wieder mit Plistedit Pro öffnen können... 😊

Vielen Dank... werde die Version gleich im DL-Bereich reinhängen...

---

### **Beitrag von „griven“ vom 2. März 2016, 22:13**

Sieht sehr gut aus aber 2 Dinge sind mir aufgefallen. Wenn für ATI-Framebuffer ein Wert gesetzt ist wird der nicht korrekt ausgelesen und proviziert einen Scriptfehler

Code

1. scripts/getvars: line 59: 16#Eulemur: value too great for base (error token is "16#Eulemur")

und zum anderen fehlen noch 2 Einträge bei den Disktemplates

Code

1. <key>AndroidDiskOptionTemplate</key>
2. <string>initrd=\$initfs root=UUID=\$uuid androidboot.hardware=android\_x86 ro SRC=\$path</string>
3. <key>AndroidDiskTemplate</key>
4. <string>\$label</string>

keine Ahnung wie weit das jemand merken würde aber der Vollständigkeit halber sei es erwähnt.

---

### **Beitrag von „Sascha\_77“ vom 2. März 2016, 23:11**

Richtig. Der versucht einen Hexwert umzuwandeln wo keiner ist. Habe es nun abgeändert und die Datei aktualisiert. Sollte nun ohne Fehler durchlaufen.

Die 2 Template-Einträge füge ich demnächst noch hinzu.

Hier mal ein kleiner Zwischenstand bzgl. GUI:

---

### **Beitrag von „griven“ vom 2. März 2016, 23:28**

Das sieht schon extrem gut aus, I like it 😄

---

### **Beitrag von „OliverZ“ vom 3. März 2016, 09:41**

Geniales Tool, weil man damit auch sehen kann ob man Änderungen im Bios richtig gemacht hat.

Super Idee danke Sascha\_77.

---

**Beitrag von „Sascha\_77“ vom 3. März 2016, 10:36**

Die GUI-Elemente habe ich jetzt alle drin. Datei ist hier bereits ausgetauscht. Jetzt erscheint kein reines Textfenster mehr wie zuvor sondern eine grafische Oberfläche. Es wird aber wie zuvor beim starten von dem Tool eine .plist auf dem Desktop angelegt.

Das Abändern der Werte innerhalb der GUI hat derzeit noch keine Auswirkungen. Das kommt später noch. Hat derzeit nur Darstellungscharacter.

Desweiteren wird man sich auch die Werte wie bei der alten Version zusammengefasst in einem Textfenster anzeigen lassen bzw. als Textfile exportieren können.

Die 2 Keys von Griven und den Key von al6042 sind ebenfalls nun enthalten.

---

**Beitrag von „crusher“ vom 3. März 2016, 10:38**

Can I change anything via this app in my NVRAM or firmware? Or just a read my defaults?

---

**Beitrag von „Sascha\_77“ vom 3. März 2016, 10:49**

That is the next step. Changing Values in the GUI is at the moment without function. It´s only for displaying.

But it exports a .plist to your Desktop with the actual values if you run the tool.

## Beitrag von „crusher“ vom 3. März 2016, 12:02

Yes I see. This is good next step. Corection after export he change in Defaults schedule and numbers eg

Code

1. `<key>FirmwareFeatures</key>`
2. `<integer>0x80001417</integer>`
3. `<key>FirmwareFeaturesMask</key>`
4. `<integer>0xC003ff37</integer>`
5. `<key>FirmwareRevision</key>`
6. `<integer>0x0001000A</integer>`

he change to

Code

1. `<key>FirmwareFeatures</key>`
2. `<integer>2147488791</integer>`
3. `<key>FirmwareFeaturesMask</key>`
4. `<integer>201342967</integer>`
5. `<key>FirmwareRevision</key>`
6. `<integer>65546</integer>`

---

## Beitrag von „Sascha\_77“ vom 3. März 2016, 13:00

Update: Man kann jetzt via Buttons die Werte als Plist oder reines Textfile exportieren lassen. Auch übernimmt er jetzt Änderungen die in den Feldern gemacht wurden. Einlesen von plists ist nun auch möglich.

Als nächstes kommt dann die Möglichkeit die Werte aus der GUI direkt ins nvram zu schreiben.

EDIT:

Soweit habe ich die nvram Befehle zusammengestellt. Bitte schaut mal drüber ob ich nix übersehen habe:

Spoiler anzeigen

---

### **Beitrag von „griven“ vom 3. März 2016, 21:12**

Sieht gut aus, auf den ersten Blick habe ich nichts finden können das fehlt.

---

### **Beitrag von „mhaeuser“ vom 3. März 2016, 22:21**

Sieht echt klasse aus! Zwei Anmerkungen noch...

- 1) Wäre vielleicht nicht schlecht, wenn du boot-args und csr-active-config vom Rest abtrennen würdest, weil's schließlich keine Ozmosis-Variablen sind.
  - 2) Wie sieht's mit QT aus? Wäre cool, wenn der Spaß auch auf Windows oder Linux laufen würde. Für Windows würde es aber wohl ein Ticket schwerer werden, weil der NVRAM-Zugriff soweit ich weiß nur via Win32 möglich ist.
- 

### **Beitrag von „Sascha\_77“ vom 3. März 2016, 22:33**

Danke. 😊 Die Buttons muss ich aber noch vernünftig anordnen.

Mit QT kann ich nicht dienen. Aber vllt. findet sich ja hier jemand der das auf dieser Plattform in Angriff nehmen könnte.

Aber die boot-args und csr tauchen aber im Ozmosis Template auf. Oder meinst Du jetzt nur optisch die beiden absetzen?



---

### **Beitrag von „griven“ vom 3. März 2016, 22:35**

Er meint, denke ich, sie optisch abzugrenzen. Genau genommen sind die Boot-args kein Teil von OZ sondern eben eher ein Teil von OS-X 😄

---

### **Beitrag von „mhaeuser“ vom 3. März 2016, 22:35**

Ja, ich meinte optisch. Im Ozmosis-Template taucht nur csr-active-config auf, wobei ich mir nicht sicher bin, warum überhaupt. Bis auf Debugging (Logs, ...) wird mit der Variable nichts angestellt.

---

### **Beitrag von „Sascha\_77“ vom 3. März 2016, 23:33**

Ok steht auf meiner ToDo-Liste.

EDIT:

Wollte heute zwar nix mehr machen aber wie das so ist ... 😊

Jetzt brauche ich nur noch ein Logo/Icon. Hatte schon mit Yogibär gesprochen. Wenn jemand ein Logo basteln möchte so kann er das gerne tun. Unten rechts ist ne Menge Platz für ein schönes Bild. 😄 Ich bin in solchen Sachen nicht so künstlerisch begabt.

---

### **Beitrag von „YogiBear“ vom 3. März 2016, 23:43**

Ganz oben links ist ein hübsches Logo versteckt 😊

Bevor es ans Logo geht sollte aber imho der Name geklärt sein. Für die Idee "Phasis" hätte ich spontan eine Idee, die ich versuchen werden morgen mal umzusetzen, jedoch bin ich auch nicht der Grafiker (mehr so mit Picknickkörben und Honig), dh. ein Pixelschubser mit mehr Skills müsste es dann nochmal in sauber zeichnen.

---

### **Beitrag von „mhaeuser“ vom 3. März 2016, 23:47**

Falls du gerne noch mehr Struktur hättest, die GUID 1F8E0C02-58A9-4E34-AE22-2B63745FA101 hat Einstellungen ("Settings") und 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102 Überschreibungen ("Overrides"). Wäre bestimmt auch cool, wenn man die Werte mit fester Form auch entsprechen anzeigen würde (z.B. die einzelnen Bits für csr-active-config und AcpiLoaderMode und außerdem die verschiedenen Variablen für die Templates ("\$label \$platform \$major \$minor \$build").

Die "NVRam Section" würde ich persönlich "Apple Boot" nennen, weil alle Werte ja teil des NVRAM sind.

Zieht schon ziemlich gut aus, reife Leistung! Und übrigens hast du Post. 😊

---

### **Beitrag von „Sascha\_77“ vom 3. März 2016, 23:53**

Ok Yogibär da bin ich mal gespannt. 😊 Habe temporär mal 2 Grafiken eingesetzt damit es nicht ganz so trist aussieht.

NVRam Section ist jetzt Apple Boot. Mit der Struktur schaue ich evtl. später mal.

Der Write to NVRam Button ist derzeit noch ohne Funktion.

---

## Beitrag von „mhaeuser“ vom 3. März 2016, 23:57

Als Logo wäre doch das Oz-Logo nicht verkehrt, oder?

Hat DeR KöNiG freundlicherweise hier hochgeladen: <http://www.insanelymac.com/forum...sis/page-154#entry2176524>

---

## Beitrag von „Sascha\_77“ vom 4. März 2016, 09:23

### [Zitat von Download-Fritz](#)

Falls du gerne noch mehr Struktur hättest, die GUID 1F8E0C02-58A9-4E34-AE22-2B63745FA101 hat Einstellungen ("Settings") und 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102 Überschreibungen ("Overrides"). Wäre bestimmt auch cool, wenn man die Werte mit fester Form auch entsprechen anzeigen würde (z.B. die einzelnen Bits für csr-active-config und AcpiLoaderMode und außerdem die verschiedenen Variablen für die Templates ("label \$platform \$major \$minor \$build").

Das könnte man evtl. über Tooltips machen. Weiss gerade aber nicht wirklich wie Dein Vorschlag "auf dem Papier" aussehen soll. Kannst Du mal konkrete Beispiele aufzeigen?

---

## Beitrag von „mhaeuser“ vom 4. März 2016, 13:53

AcpiLoaderMode und csr-active-config sind ja Bitmasken, deren Werte könne man als Checkboxes darstellen.

Für die Boot Templates könnte man Tooltips oder ein einfaches Label nehmen. 😊

---

## Beitrag von „Sascha\_77“ vom 4. März 2016, 15:45

Du meinst bei AcpiLoaderMode sowas?

Code

1. ACPI\_LOADER\_MODE\_DISABLE 0x00000000
2. ACPI\_LOADER\_MODE\_ENABLE 0x00000001
3. ACPI\_LOADER\_MODE\_DUMP 0x00000002
4. ACPI\_LOADER\_MODE\_WINDOWS 0x00000008
5. ACPI\_LOADER\_MODE\_UPDATE\_LEGACY 0x00000045

Die habe ich z.b. zur Erläuterung schon in die Tooltips gepackt bei dem Textfenster. Da jetzt noch Checkboxen in der GUI anzubringen frachtet das Fenster irgendwie noch mehr zu.

Ich habe jetzt eine Version hochgeladen die u.a. das Schreiben ins NVRam simuliert. Wenn man diesen

Button drückt erzeugt er auf dem Desktop eine "nvram\_debug.txt" Datei. Dort drinnen sind die Befehle die er ausgeführt HÄTTE. Hier würde ich Euch bitten zu schauen ob alle Parameter korrekt enthalten sind.

Wenn die soweit ok sind kann ich dieses Script dann scharfschalten.

Ferner habe ich noch ein paar Fehler beseitigt insofern, dass z.b. bei DisableAtiInjection, sofern im nvram keine Werte ausgelesen werden können (iokit/common data was not found) er automatisch den Wert 1 erzeugt und z.b. bei Disable Voodoo HDA eine 0 setzt sofern im nvram nichts vorhanden ist.

---

## Beitrag von „apatchenpub“ vom 4. März 2016, 17:12

habe hier auch noch die komplette Liste für eine Bitmaske

Configuration	MEM	N/A	SRAM	STrace	Intern	Debug	PIB	FS	Keats	Clover	SRAM
carutil enabled --no-internal	00	0	0	0	0	0	0	0	0	0x00	100000000
carutil enabled	10	0	0	0	1	0	0	0	0	0x10	100000000
carutil enable --without keat	11	0	0	0	1	0	0	0	1	0x11	110000000
carutil enable --without fs	12	0	0	0	1	0	0	1	0	0x12	110000000
carutil enable --without debug	14	0	0	0	1	0	1	0	0	0x14	110000000
carutil enable --without dtrace	16	0	0	1	1	0	0	0	0	0x16	000000000
carutil enable --without swan	50	0	1	0	1	0	0	0	0	0x50	100000000
carutil disabled	77	0	1	1	1	0	1	1	1	0x77	100000000
carutil disabled (no internal)	67	0	1	1	0	0	1	1	1	0x67	100000000

---

## Beitrag von „mhaeuser“ vom 4. März 2016, 19:40

[Zitat von Sascha 77](#)

Code

1. ACPI\_LOADER\_MODE\_DISABLE 0x00000000
2. ACPI\_LOADER\_MODE\_ENABLE 0x00000001
3. ACPI\_LOADER\_MODE\_DUMP 0x00000002
4. ACPI\_LOADER\_MODE\_WINDOWS 0x00000008
5. ACPI\_LOADER\_MODE\_UPDATE\_LEGACY 0x00000045

ACPI\_LOADER\_MODE\_UPDATE\_LEGACY hat den Wert 0x45 (0x40 ist ja kein Bit) und in der Liste fehlt "ACPI\_LOADER\_MODE\_DARWIN 0x00000004" 😊

---

## Beitrag von „Adnarel“ vom 5. März 2016, 10:30

Nächstes Feature Request:

Die vorgefundenen Ozmosis/Apple Boot Einstellungen, sofern konvertierbar, in eine Clover config.plist parsen.  
(SMBios, Boot-Args...)

Das ist bestimmt jede Menge if/elif ...

Dazu müssten wir jetzt mal eine Übersetzungstabelle der Variablennamen anfertigen. (Muss

jetzt mal gerade wieder arbeiten gehen, und da kann ich leider nichts nebenbei machen, sonst hätte ich damit schon mal angefangen...)

