

Erledigt

SMBIOS mit OZ 1479 richtig setzen - jeder Rechner ein Unikat

Beitrag von „Griven“ vom 20. Juni 2015, 02:33

Das Thema SMBIOS ist spätestens seit der Einführung von FaceTime, iMessage und iCloud wieder ein brandaktuelles Thema, denn will man diese Dienste nutzen ist es unabdingbar ein passendes SMBIOS zu erstellen das den Rechner zu einem Unikat macht aber wie macht man das unter Ozmosis 1479m und wo kommen die Werte her die da eingetragen werden müssen? Hier ein kleiner Überblick wie Ihr Euren Hackintosh zu einem Unikat macht.

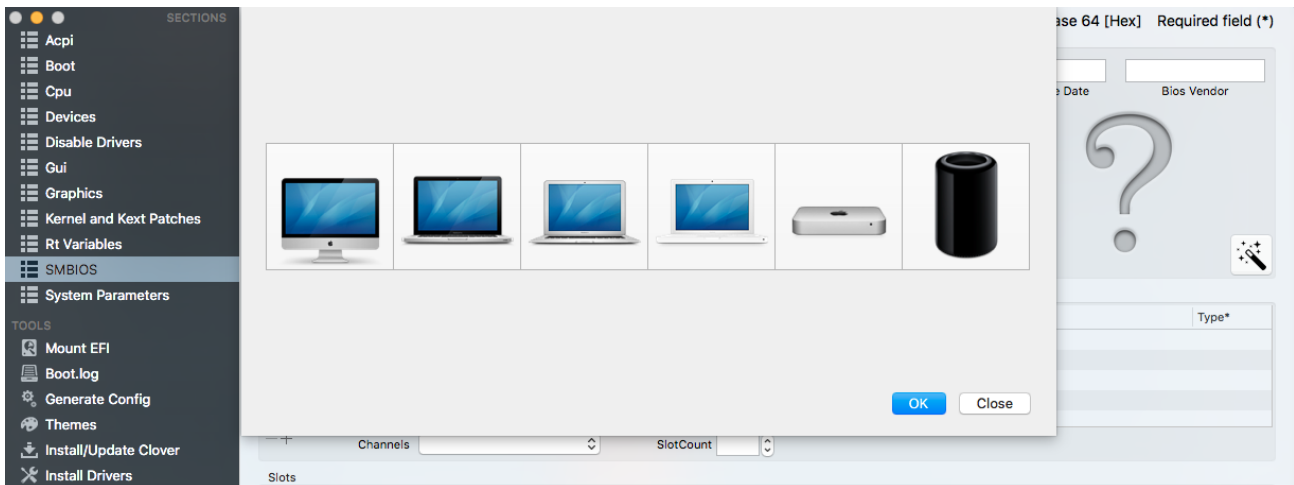
Grundsätzliches:

Das SMBIOS ist mehr als reine Kosmetik denn es hat weit mehr Aufgaben als die Anzeige unter "über Diesen Mac" zu befeuern vielmehr dient es dazu einen Computer eindeutig zu identifizieren und ganz nebenher beeinflusst es auch einen ganzen haufen Dinge innerhalb von OS-X die maßgeblich dazu beitragen ob Euer Rechner gut läuft oder eben auch nicht. Das SMBIOS nimmt zum Beispiel ganz erheblichen Einfluss darauf wie zum Beispiel das PowerManagement geregelt ist oder ob Euer Hackintosh bei kompatibler WLAN/Bluetooth Karte Handoff beherrscht oder nicht usw. . Ausgestattet mit diesem Wissen sollte es klar sein, dass man ein SMBIOS wählt das möglichst nahe an der verbauten Hardware ist sprich es macht keinen Sinn einen MacPro6.1 zu wählen weil die Tonne so schön stylisch ist wenn in der Kiste dann am ende ein Core2Duo steckt und umgekehrt macht es genau so wenig Sinn stoisch immer nur MacPro3.1 zu wählen weil das ja das kompatibelste Model ist. Die Zeiten der reinen Kosmetik sind spätestens seit Mavericks vorbei...

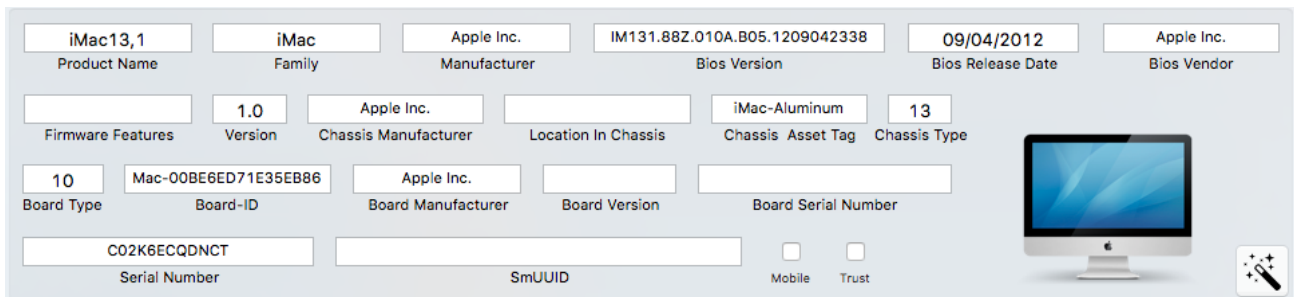
Woher die Werte nehmen?

Um an passende Werte für den Hackintosh zu kommen bietet sich aktuell am ehesten eine aktuelle Version das CloverConfigurators an ([klick](#)) denn dessen SMBIOS Generator liefert verlässlich brauchbare Werte. Einmal geladen und gestartet ignorieren wir einfach die Meldung das keine config.plist gefunden wurde und quittieren das Angebot die EFI Partition zu mounten mit einem NO. Anschließend klicken wir zielstrebig links auf den Punkt SMBIOS und in dem sich dann öffnenden Bereich auf den Zauberstab & voilà schon haben wir alles, was wir brauchen





Aber hier erzähle ich Euch sicher auch nichts Neues. Wählt das passende System mit Bedacht aus (Für Desktops geeignet sind am ehesten iMac und MacPro). Mein System1 läuft als iMac 13.1 passend zu seiner Konfiguration (Z7X Chipsatz, IvyBridge CPU). Einmal gewählt und mehrfach auf "Shake" geklickt habt Ihr fast alle Werte, die nötig sind.



SMBIOS über den NVRAM setzen...

Die bei weitem einfachste und vielfach auch problemloseste Variante das SMBIOS bei OZ zu setzen führt über den NVRAM. Seit OZ1479m ist der NVRAM bei nahezu allen Mainboards beschreibbar demnach empfiehlt es sich diesen Weg zu gehen. Damit man den NVRAM aus OS-X heraus beschreiben kann wird zunächst das Terminal gestartet und anschließend der folgende Befehl gefolgt vom eigenen PW eingegeben:

Code

1. `sudo -s`

ist alles richtig gelaufen quittiert OS-X Eure Eingabe mit

Code

1. bash-3.2#

Einmal das Terminal so vorbereitet öffnet Ihr Euch den Texteditor und erstellt eine neue Datei in die Ihr das folgende hinein kopiert:

Code

1. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BaseBoardAssetTag="Base Board Asset Tag#"
2. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BaseBoardSerial=
3. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BiosDate=
4. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BiosVersion=
5. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BoardVersion=
6. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:ChassisAssetTag=
7. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:EnclosureType=
8. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:FirmwareFeatures=2.147.488.791
9. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:FirmwareFeaturesMask=3.221.487.415
10. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:FirmwareRevision=65.546
11. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:FirmwareVendor=Apple
12. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:HardwareAddress=
13. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:HardwareSignature=
14. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:Manufacturer="Apple Inc."
15. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:ProcessorSerial=
16. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:ProductFamily=
17. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:ProductId=
18. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:ProductName=
19. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:SystemSKU="System SKU#"
20. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:SystemSerial=
21. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:SystemVersion=1.0

Alles anzeigen

wenn auch das erledigt ist geht es daran die Werte zu komplettieren hierbei gilt überall da, wo nachdem dem "=" schon etwas steht lassen wir die Finger von bei allen anderen Werten kommen entweder die Werte aus dem CloverConfigurator rein oder an den Stellen wo die Werte nicht aus dem Configurator kommen gibt es einen Verweis darauf wo sie herkommen. Also los geht's...

Code

1. BaseBoardAssetTag="Base Board Asset Tag#"
2. BaseBoardSerial= <- Serial Number
3. BiosDate= <- Bios Release Date
4. BiosVersion= <- Bios Version
5. BoardVersion= <- Board Type
6. ChassisAssetTag= <- Chassis Asset Tag
7. EnclosureType= <- Chassis Type
8. FirmwareFeatures=2.147.488.791
9. FirmwareFeaturesMask=3.221.487.415
10. FirmwareRevision=65.546
11. FirmwareVendor=Apple
12. HardwareAddress= <- Mac Adresse von EN0 aus Systeminformationen -> Netzwerk -> Ethernet im Format xx:xx:xx:xx:xx:xx
13. HardwareSignature= <- eine beliebige UUID -> Terminal -> uuidgen eingeben und kopieren
14. Manufacturer="Apple Inc."
15. ProcessorSerial= <- Serial Number
16. ProductFamily= <- Family
17. ProductId= <- Board-ID
18. ProductName= <-Product Name
19. SystemSKU="System SKU#"
20. SystemSerial= <- Serial Number
21. SystemVersion=1.0

Alles anzeigen

Habt Ihr alles zusammen könnte das zum Beispiel (**NICHT** kopieren, ist ein Beispiel) so aussehen:

Code

1. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BaseBoardAssetTag="Base Board Asset Tag#"
2. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BaseBoardSerial=C02K6ECQDNCT
3. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BiosDate=09/04/2012
4. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BiosVersion=IM131.88Z.010A.B05.1209042338
5. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BoardVersion=10
6. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:ChassisAssetTag=iMac-Aluminum
7. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:EnclosureType=13
8. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:FirmwareFeatures=2.147.488.791

9. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:FirmwareFeaturesMask=3.221.487.415
10. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:FirmwareRevision=65.546
11. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:FirmwareVendor=Apple
12. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:HardwareAddress=40:0b:23:1c:35:10
13. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:HardwareSignature=B5D37D4E-90EC-4DE6-B8D7-6373D34EFE31
14. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:Manufacturer="Apple Inc."
15. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:ProcessorSerial=C02K6ECQDNCT
16. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:ProductFamily=iMac
17. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:ProductId=Mac-00BE6ED71E35EB86
18. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:ProductName=iMac13,1
19. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:SystemSKU="System SKU#"
20. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:SystemSerial=C02K6ECQDNCT
21. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:SystemVersion=1.0

Alles anzeigen

Habt Ihr alles zusammen muss nur noch Zeile für Zeile ins Terminal kopiert werden und anschließend der Rechner neu gestartet werden und das war es schon, Euer System besitzt ab jetzt eine eindeutige Definition mit der es eigentlich kein Problem mehr sein sollte alle Apple Dienste zu nutzen.

Was aber wenn sich der NVRAM nicht beschreiben lässt?

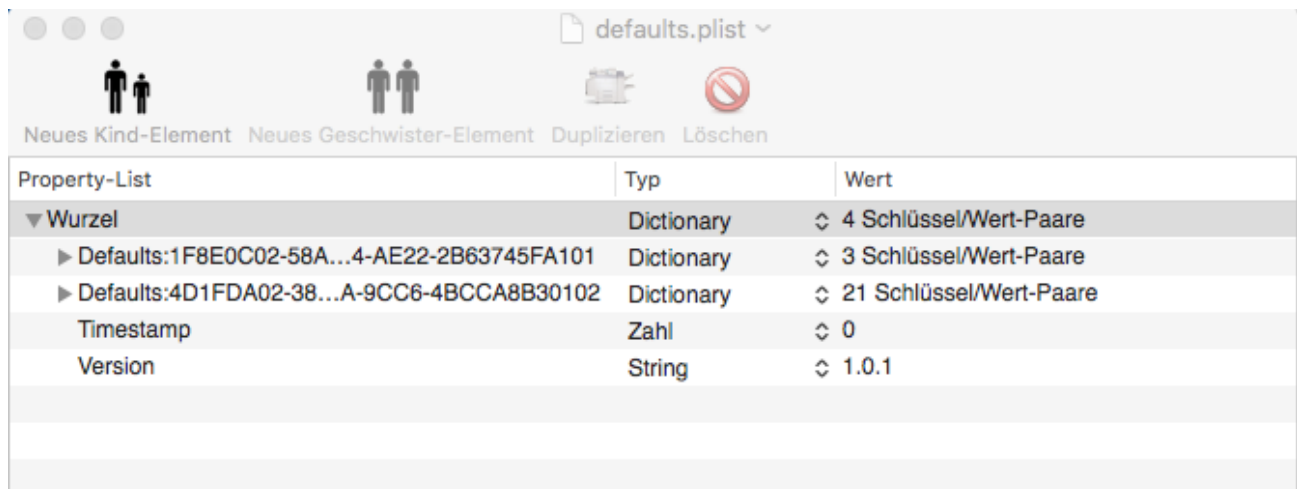
Obwohl die 1479m eigentlich die NVRAM Sperre bei allen Boards aufheben sollte kommt es immer wieder vor, dass sich einzelne Firmwares doch weigern den NVRAM zu beschreiben aber keine Panik auch hier lässt sich das SMBIOS selbstverständlich ändern nur eben nicht über den NVRAM. Damit es in dem Fall mit einem perfekten SMBIOS klappt müssen die Werte in eine .plist Datei eingetragen werden die dann auf die /EFI Partition wandert. Im großen und ganzen sind die Vorbereitungen die selben Sprich ihr könnt Euch zum ermitteln der Werte weitestgehend an den oberen Teil der Anleitung halten.

Was brauche ich um die defaults.plist zu erstellen bzw. zu bearbeiten?

Neben der OzmoisDefaults.plist (-> hier aus dem Ozmosis Paket extrahiert [defaults.plist.zip](#)) benötigt Ihr noch einen plist Editor Eurer Wahl oder das Programm textedit aus OS-X. Als plist Editor eignet sich entweder der in Xcode enthaltene Apple eigene plist Editor oder aber PlistEdit Pro welches sich als kostenlose trial herunterladen lässt. Egal was Ihr verwendet es unterscheidet sich lediglich in der Handhabung, die eingetragenen Werte sind überall die selben.

Alles zusammen, was ist zu tun?

Wenn wir alles zusammen haben geht es darum die Werte in die defaults.plist einzutragen. Hierzu öffnen wir zunächst die heruntergeladene plist Datei im PlistEditor (ich benutze PlistEdit Pro) zum bearbeiten. Direkt nach dem Öffnen der defaults.plist fällt auf, dass die Struktur den NVRAM Befehlen oben nachempfunden ist.



Property-List	Typ	Wert
▼ Wurzel	Dictionary	↕ 4 Schlüssel/Wert-Paare
▶ Defaults:1F8E0C02-58A...4-AE22-2B63745FA101	Dictionary	↕ 3 Schlüssel/Wert-Paare
▶ Defaults:4D1FDA02-38...A-9CC6-4BCCA8B30102	Dictionary	↕ 21 Schlüssel/Wert-Paare
Timestamp	Zahl	↕ 0
Version	String	↕ 1.0.1

Für unser Vorhaben ist nur der 2. Knoten (Defaults:4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102) interessant denn analog zum NVRAM beherbergt auch die defaults.plist die Informationen zum SMBIOS unter dieser Adresse was ersichtlich wird, wenn wir den Knoten einfach mal aufklappen.

▼ Defaults:4D1FDA02-38...A-9CC6-4BCCA8B30102	Dictionary	↕ 21 Schlüssel/Wert-Paare
BaseBoardAssetTag	String	↕ Base Board Asset Tag#
BaseBoardSerial	String	↕
BiosDate	String	↕
BiosVersion	String	↕
BoardVersion	String	↕
ChassisAssetTag	String	↕
EnclosureType	Zahl	↕ 13
FirmwareFeatures	Zahl	↕ 2.147.488.791
FirmwareFeaturesMask	Zahl	↕ 3.221.487.415
FirmwareRevision	Zahl	↕ 65.546
FirmwareVendor	String	↕ Apple
HardwareAddress	String	↕

Da die einzelnen Punkte genau so benannt sind, wie bei den NVRAM Befehlen ist der Rest eigentlich simples Copy&Paste sprich einfach die Werte jeweils passend zu den jeweiligen Punkten in die defaults.plist einfügen und anschließend speichern. Einmal erzeugt und gespeichert müssen wir Ozmosis jetzt nur noch davon überzeugen unsere Defaults auch zu verwenden. Hierzu binden wir uns zunächst mal die EFI Partition unserer ersten Platte ein (hier sucht OZ per default nach diesen Daten). Dazu einfach ein Terminal öffnen und folgenden Befehl eingeben

Code

1. diskutil list

und erhalten anschließend in etwa die folgende Ausgabe

Code

1. diskutil list
2. /dev/disk0 (internal, physical):
3. #: TYPE NAME SIZE IDENTIFIER
4. 0: GUID_partition_scheme *128.0 GB disk0
5. 1: EFI EFI 209.7 MB disk0s1
6. 2: Apple_HFS yose1 127.0 GB disk0s2
7. 3: Apple_Boot Recovery HD 784.2 MB disk0s3
8. /dev/disk1 (internal, physical):
9. #: TYPE NAME SIZE IDENTIFIER
10. 0: GUID_partition_scheme *60.0 GB disk1
11. 1: EFI EFI 209.7 MB disk1s1
12. 2: Apple_HFS capitán 59.2 GB disk1s2
13. 3: Apple_Boot Recovery HD 650.0 MB disk1s3

Alles anzeigen

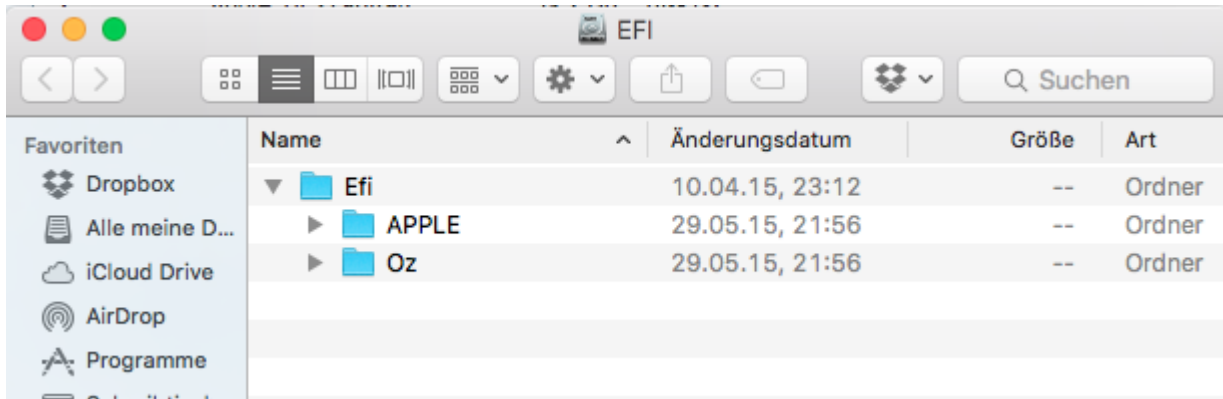
Die erste Platte trägt den Namen disk0 und die auf ihr enthaltene EFI Partition den Namen disk0s1. Damit wir die EFI Partition nun ins System einbinden können geben wir dann die folgenden Befehle ein

Code

1. mkdir /Volumes/efi
2. sudo mount -t msdos /dev/disk0s1 /Volumes/efi

und schon ist die Partition eingebunden. Als letzten Schritt müssen wir nur noch unsere eben

erstellte defaults.plist in das passenden Verzeichnis kopieren. Öffnet Ihr die eben eingebundene EFI Partition gibt es hier einen Ordner EFI der 2 Unterordner enthält nämlich einen Namens Apple und einen anderen Namens OZ...



Die defaults.plist wird nun einfach direkt in den OZ Ordner eingefügt und im Anschluss der Rechner heruntergefahren. Ist der Rechner heruntergefahren schalten wir ihn wieder ein und gehen direkt ins Bios. Um die neuen Werte aus der default.plist nun zu übernehmen drücken wir im Bios die Tasten CMD+ALT+P+R gleichzeitig woraufhin sich der Bildschirm erst Rot dann Orange und letztlich grün färbt und der Rechner neu startet voilà das wars schon Ihr habt erfolgreich ein SMBIOS gesetzt.

Beitrag von „Banane“ vom 20. Juni 2015, 07:17

Hallo, ich bin da ziemlich neu mit ozmosis. Wieso muss man bei
BaseBoardAssetTag="Base Board Asset Tag#"

keine Zahlen eintragen sondern nur Base Board Asset Tag# mit der Raute?

Beitrag von „JoshuaOD“ vom 20. Juni 2015, 13:03

Gutes TUT, leicht zu verstehen und anzuwenden ABER..... mein NVRAM ist leider nicht

beschreibbar. Habe mir das BIOS für das Board hier runtergeladen installiert und nach viel Verzweiflung das System zum laufen bekommen 😊
Jetzt frage ich mich wie ich es hinbekomme das SMBIOS richtig zu editieren? Bin da ehrlich hab daran nicht gedacht das man es ändern sollte.

Beitrag von „Griven“ vom 20. Juni 2015, 18:14

Eingangspost um Plan B erweitert 😊
[Banane](#) der BaseBoardAssetTag bleibt unberührt sprich so übernehmen, wie es da steht 😊

Beitrag von „JoshuaOD“ vom 21. Juni 2015, 16:28

Danke Griven!!! 

Es ist wirklich prima wie man hier durch eure Hilfe weiter kommt, macht bloß so weiter!

Beitrag von „clfaster“ vom 25. Juni 2015, 12:11

Gibt es eine Möglichkeit den NVRAM wieder zurückzusetzen? Nachdem ich die oben genannten Schritte durchgeführt habe startet mein Yosemite gar nicht mehr. Das Apple Logo erscheint, der Ladebalken bewegt sich circa 1/3, danach wird der Bildschirm schwarz und mein Monitor schaltet sich kurze Zeit später aus.

Ich habe bereits versucht mein CMOS zurückzusetzen und Motherboard auf defaults, hat allerdings nicht wirklich etwas bewirkt.

Das Problem ist vor allem das selbst der Installationsstick an der gleichen Stelle abbricht.

Beitrag von „steff89“ vom 25. Juni 2015, 12:53

Hallo cfaster,

Ich wollte auch mal den NVRAM löschen. Es gibt mehrere Möglichkeiten.

Die einfachste ist über das Terminal. einfach "sudo nvram -d" eingeben und alle Werte sollte gelöscht sein.

Da du jedoch nicht mehr in OSX kommst gibt es noch eine weitere Möglichkeit die bei mir funktioniert hat:

1. Original BIOS vom Hersteller zurückflashen
2. BIOS CMOS Reset durchführen (lies dir dafür am Besten mal das Manual deines Mainboards durch)

Jetzt sollte dein Mainboard wieder im Originalzustand wie bei Auslieferung sein und du kannst Ozmosis neu flashen.

Hört sich zwar kompliziert an, aber das Ganze ist eigentlich in 15 Minuten erledigt.

PS: Noch eine Frage in eigener Sache. Ich bin mir unsicher welche System ich für das SMBIOS verwenden soll für meinen Hackintosh (siehe Signatur). Man liest hier ja immer von MacPro3,1 als Goldstandard, aber die hatten ja Xeon Prozessoren verbaut und haben ja auch ein paar Jahre auf dem Buckel... Laut MacTracker gibt es eine Version des 5k iMac mit 4790k Prozessor. Würde es also Sinn machen iMac15,1 zu verwenden? Hat jemand damit Erfahrung? Was würdet ihr mir empfehlen?

Viele Grüße,

Steffen

Beitrag von „Doe1John“ vom 25. Juni 2015, 13:08

Wenn dein Hacki auf MacPro3.1 läuft, ist er am flexibelsten.

Beitrag von „steff89“ vom 25. Juni 2015, 13:30

Was meinst du mit "am flexibelsten". Ziel ist es doch dass das System am stabilsten läuft und die Hardware am besten unterstützt.

Viele Grüße,
Steffen

Beitrag von „Griven“ vom 25. Juni 2015, 13:48

Ansonsten kann man einen NVRAM reset auch unter Ozmosis einfach erreichen indem man ins Setup geht und dann die Tasten CMD+ALT+P+R drückt (hilft natürlich nur dann was, wenn die defaults.plist nicht auf der /EFI Partition gelagert ist sprich die Einträge direkt in den NVRAM geschrieben wurden).

Beitrag von „clfaster“ vom 25. Juni 2015, 14:27

[steff89](#)

Danke für den Tipp!

Habe genau diesen iMac genommen! Danach ging nichts mehr 😊 Also ich würde liebe die Finger davon lassen, weiß nicht vielleicht liegt das an der Grafikkarte, in denen sind meines Wissens nur mehr ATI Karten verbaut?!

Danke auch an Griven, habe dies nun gemacht, danach hats immer noch nicht geklappt, aber nach einem erneuten [BIOS Flash](#) läuft alles wieder.

Ich werde es wohl mit einem anderen SMBIOS versuchen.

lg

Beitrag von „powjk“ vom 28. Juni 2015, 20:30

Vielen Dank für die Anleitung, mit deren Hilfe ich mein Grafikproblem durch Anlegen einer default.plist lösen konnte.

Um iMessage zum Laufen zu bekommen habe ich diverse Beiträge durchgelesen und es ergeben sich noch folgende Verständnisfragen:

Handelt es sich bei der hier genannten "BaseBoardSerial" um den Wert, der an anderer Stelle auch als "MLB" (Main Logic Board) bezeichnet wird und im Clover Configurator unter SMBIOS, sowie im iMessageDebug "Board Serial Number" heißt?

Wenn das so ist, ergibt das Verwenden der Serial Number unter diesem Wert (wie oben beschrieben) laut iMessageDebug bei mir ein "(null)" hinter BoardSerialNumber (siehe Anhang).

Das ist ein KO für iMessage, oder?

Wo und wie lässt sich eine funktionierende MLB generieren?

Beitrag von „steff89“ vom 28. Juni 2015, 20:50

Soweit ich weiss ist die BaseBoardSerial einfach die Serial Nummer die auf 17 Zeichen verlängert wird. Also zum Beispiel einfach 123456 anhängen.

Viele Grüße,
Steffen

Beitrag von „Griven“ vom 28. Juni 2015, 20:59

Ist richtig wobei das verlängern bei OZ eigentlich nicht nötig ist da OZ das automatisch erledigt, aber auch nicht schadet 😊

Beitrag von „powjk“ vom 29. Juni 2015, 00:18

Vielen Dank für die sehr hilfreiche Antwort steff89 u. Griven!

Nachdem ich die BaseBoardSerial in meiner default.plist um fünf Ziffern ergänzt habe (und den Reset im Bios durchgeführt habe), wurde sie mir in iMessageDebug vollständig angezeigt.

Und was soll ich sagen - iMessage funzt! 😊

In meinem Fall hat OZ die Ergänzung auf 17 Zeichen offensichtlich nicht automatisch erledigt. Vielleicht gibt es da einen Unterschied ob NVRAM oder .plist-Datei?

Beitrag von „Griven“ vom 29. Juni 2015, 23:11

Wäre mir nicht bekannt das es da Unterschiede gibt denn im Grunde macht OZ ja nichts anderes als die Daten aus der plist zu lesen und in den NVRAM zu schreiben (im Übrigen auch, wenn der NVRAM nicht dauerhaft beschreibbar ist) bei Systemen ohne dauerhaft beschreibbaren NVRAM eben bei jedem booten aufs Neue...

Beitrag von „Mach-O_64“ vom 30. Juni 2015, 22:54

Ich versuche es gerade mit der MacPro3,1 Definition.
Eines macht mir Probleme:

```
bash-3.2# NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:ProductFamily=Mac Pro
```

```
NVRAM: Error getting variable - 'Pro': (iokit/common) data was not found
```

```
bash-3.2#
```

Vllt mit _?

Beitrag von „Griven“ vom 30. Juni 2015, 23:02

Bei den NVRAM Befehlen müssen bei String Inhalten immer Anführungszeichen gesetzt werden

sprich...

Code

1. Integer Wert = `sudo nvram someaddress:somename=12345`
2. Binärwert = `sudo nvram someaddress:somename=%01` oder `%00`
3. String = `sudo nvram someaddress:somename="some string content"`
4. Hexwert = `sudo nvram someaddress:somename=xx:xx:xx:xx`

Die Anführungszeichen sind wichtig um OS-X zu sagen, dass ein String in den NVRAM geschrieben werden soll. Das NVRAM Kommando kann ohne Anführungszeichen Sonderzeichen wie eben das Lesezeichen nicht interpretieren.

Beitrag von „Mach-O_64“ vom 30. Juni 2015, 23:07

Ich habe heute einen [BIOS Update](#) gemacht. Seitdem kann man den NVRAM nicht mehr beschreiben. Aber das ist ja jetzt OffTopic.

Beitrag von „Griven“ vom 30. Juni 2015, 23:18

Update auf was?

Beitrag von „Mach-O_64“ vom 30. Juni 2015, 23:19

Ozmosis von F7 auf F8. Ich erstelle gleich einen Thread.
Habe gerade ein großes Problem.

Beitrag von „Bernardo“ vom 1. November 2015, 16:24

Hallo,

ich bekomme auch BoardSerialNumber: (null)

Die Serial Number habe ich auf 17 Stellen aufgefüllt. Nach dem Neustart und Abfrage mit

iMessageDebug habe ich immer noch (null)

Hat jemand eine Idee warum das nicht geht?

Beitrag von „MacGrummel“ vom 1. November 2015, 17:34

Wie hast Du denn die Nummer eingegeben? Ein falsches oder fehlendes Zeichen, und Du landest im Nirvana..

Beitrag von „Bernardo“ vom 1. November 2015, 17:43

habe `sudo nvram 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:SystemSerial=C02LY4A1DWYN79426` ins Terminal eingegeben.

Beitrag von „MacGrummel“ vom 1. November 2015, 17:51

Richtig wäre wohl [\(wie hier!\)](#):

Code

```
1. sudo nvram 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BaseBoardSerial=C02LY4A1DWYN79426
```

Beitrag von „Bernardo“ vom 1. November 2015, 17:59

habe ich auch versucht. Zu sehen sollte es doch nach einem Neustart sein?

edit: nach dem zweiten Neustart ist der Wert erst eingetragen gewesen. 😞

edit: habe iMessage und FT ausprobiert... soll den Support anrufen. Muss ich zuerst noch was löschen? Oder nur warten?

Beitrag von „mhaeuser“ vom 1. November 2015, 18:08

Ja, sollte es. Wenn nicht, unterstützt AptioFix die NVRAM-Zugriffsweise dieser Firmware nicht.

Beitrag von „Bernardo“ vom 1. November 2015, 18:24

bedeutet das jetzt, dass ich mir das SMBIOS nochmal neu zusammensammeln muss? Z.B. als iMac?

Beitrag von „MacGrummel“ vom 1. November 2015, 18:27

Auch wenn sich Download-Fritz sicher besser mit der Hard-und Software von Hackintosh-Rechnern auskennt 🙄 : Wenn Du alle Werte einzigartig generiert und der Mac-Logik so halbwegs entsprechend eingegeben hast: wir haben etliche Z97M-Rechner in iMessages & Facetime laufen! Es sollte also gehen! Was ich bei meinem Z97MX-G5 gemacht hab, als mich das Warten genervt hatte: Ich hab mit diesem Rechner das Passwort im Apple-Account geändert. Danach ging es in ner Viertelstunde! Vorher hab ich auch schon zwei Tage gewartet. ALSO: Warten! Leider ist Apple für die Vergabe der Zugänge zuständig und der Sonntag ist da immer so eine Sache!!

Beitrag von „Bernardo“ vom 1. November 2015, 18:35

ich werde mal warten da der App Store funktioniert und meine App's da noch vorhanden sind.
Eine neue Apple ID kann ich mir immer noch holen.
Vielen Dank bis dahin, melde mich wieder um zu berichten ob's geklappt hat.

Beitrag von „Sheldonator“ vom 8. Dezember 2015, 15:37

Hallo aus dem nebligen Graz!

Ich werde meinen Hackintosh mit einem neuen Board/CPU (Gigabyte GA-Z97M-DS3H) versorgen und habe deshalb versucht mich über das Forum auf den neuesten Stand zu bringen. Blicke hoffentlich halbwegs durch, und werde nächste Woche die neue Hardware mit dem neuen Ozmosis für El Capitan installieren.

Zum Thema "Systemdefinition SMBIOS" habe ich da noch ein Verständnisproblem. So wie ich das kapiert habe, kann man sich die Seriennummer etc. über den CloverConfigurator/SMBIOS Generator generieren lassen. Ich bilde mir ein, in älteren Beiträgen gelesen zu haben, dass es besser wäre die Seriennummer eines originalen Macs zu nehmen (in meinem Fall ein älteres MacBook Air (2010). Dazu meine Fragen. Stimmt das noch und wenn ja, ist das in meinem Fall empfehlenswert (da hier ja eine völlig andere Hardware verbaut ist)? Und kommt das System dann bei der Anmeldung an die Apple Dienste (iMessage etc.) nicht in irgendeinen Konflikt, da ja von 2 unterschiedlichen Computern die selben Seriennummern gemeldet werden. Oder habe ich da was komplett missverstanden? Bei neuen Hackintosh habe ich im übrigen geplant die Systemdefinition iMac14,2 zu nehmen.

Bitte um kurze Info dazu.

Auch will ich mich bei dieser Gelegenheit bei allen Bedanken, die hier so gute Arbeit leisten und ihre Zeit investieren. Ohne euch würde ich mir diese Installation nie zutrauen. VIELEN DANK!!

Beitrag von „tidow“ vom 8. Dezember 2015, 19:36

So, ich habe mich bei gemacht, und mein Hackti auf i Mac 13,2 geändert.
Wie kann ich das kontrollieren. ist das den so richtig?
Modellname: iMac

Modell-Identifizierung: iMac13,2

Prozessortyp: Intel Core i5

Prozessorgeschwindigkeit: 3,1 GHz

Anzahl der Prozessoren: 1

Gesamtanzahl der Kerne: 4

L2-Cache (pro Kern): 256 KB

L3-Cache: 6 MB

Speicher: 12 GB

Boot-ROM-Version: #####

SMC-Version (System): 1.30f3

Seriennummer (System): #####

Hardware-UUID: #####

Also geht nicht, FaceTime gibt eine Fehler Aus. Bei der Aktivierung ist ein Fehler aufgetreten.



Beitrag von „MacGrummel“ vom 8. Dezember 2015, 22:39

Die iMac13,2-Einstellung klappt leider selten komplett. Such Dir lieber eine andere Konfiguration. In den [Erklärungen zu iMessages](#) ist auch eigentlich ausreichend erklärt, wie das geht. Und wenn der Configurator nicht reicht, gibt es da noch ein anderes Würfel-Programm [Apple MAC Hack](#), dass sehr gute Ergebnisse bringt.

Das mit den Daten eines richtigen Mac war nur im letzten Winter für einige Zeit richtig. Da hatte Apple eine Änderung der iMessage-Dienste nach der nächsten ausprobiert. Jetzt müssen das eben keine Daten von einem anderen Rechner sein (weil nur noch genau ein Rechner mit einem Datensatz funktioniert), sondern nur noch möglichst genau der Apple-Logik entsprechen..

Beitrag von „tidow“ vom 9. Dezember 2015, 09:31

Und was mache ich jetzt. Soll ich die Anleitung von Griven noch mal machen mit z.b. 13.1-IMac. Er Hat ja das gleiche Board.

Werden die jetzigen NVRAM Daten überschrieben? oder was muss ich noch beachten.

Danke für Deine Hilfe.

So, habe es geändert auf iMac 13.1 und mit Apple Mac Hack die generierten Code geändert. hat gut geklappt FaceTime und Messenger laufen.i Cloud auch.Danke Ohne Euch würde ich immer der Ratlose sein. A small green parrot icon with its wings spread, used as a signature or reaction.

Beitrag von „MacGrummel“ vom 9. Dezember 2015, 12:48

Schön zu hören bzw. lesen! Da für haben wir uns halt im letzten Winter die Zeit am Rechner totgeschlagen..

Beitrag von „Sheldonator“ vom 9. Dezember 2015, 14:52

Hallo,

Habe ich verstanden, Danke an MacGrummel für die Darstellung. Werde mich dann nächste Woche über die Installation wagen. Gebe euch bescheid wie es geklappt hat.

Vielen Dank nochmals! 😊

Beitrag von „daniel14513“ vom 10. Dezember 2015, 09:05

Hallo zusammen,

da ich meinen Rechner gerade frisch mit OZ und El Capitan 10.11.2 aufgesetzt habe, wage ich mich nicht so richtig an die SMBIOS Variante (in NVRAM schreiben) ran. Bei Yosemite konnte man die Systemdefinition über Multibeast installieren lassen, ist das heutzutage noch möglich? Was sind die Unterschiede zwischen der Variante die [@griven](#) beschreibt und der Mutlibeast Variante?

Beitrag von „tidow“ vom 10. Dezember 2015, 09:35

Hi, wenn ich mich nicht Täusche ist die Multibeats-Geschicht nicht für OZ geeignet. Mit MB hast Du ja einen Bootlader installiert.

Da wird den die Datei Extra/SMBios eingerichtet. Schreibt nicht in den (NVRAM). Außerdem ist dann alle Doppelt gemoppelt, und es wird zu Störungen kommen. Du hast dann 2

FakeSMC.kext, die sich selten vertragen.
Also mutig ans Werk mit der Beschreibung von [@griven](#)
Es kann auch bestimmt jemand Über TeamViewer helfen.
iMac 13.1 passt gut für unsere Boards

Beitrag von „daniel14513“ vom 10. Dezember 2015, 09:40

das wäre toll mit TeamViewer , ich habe halt nur keine Lust zum 235. mal El Capitan neu zu installieren und von Vorne anzufangen. Bin gerade echt froh das alles so sauber läuft 😊

Beitrag von „tidow“ vom 10. Dezember 2015, 11:24

aber es lohnt sich. Der Rechner läuft irgend wie schneller,entspannter und alles haut hin.



Beitrag von „daniel14513“ vom 10. Dezember 2015, 11:52

"Habt Ihr alles zusammen muss nur noch Zeile für Zeile ins Terminal kopiert werden und anschließend der Rechner neu gestartet werden und das war es schon"
aus dem Texteditor kopieren und in das Terminal einfügen? Enter drücken danach oder einfach nur reinkopieren?

edit: funtkioniert die Anleitung auch mit 10.11.2 sowie der neuen OZ Version?

Beitrag von „apfelnico“ vom 10. Dezember 2015, 14:06

Zeile für Zeile mit Enter bestätigen. Beim ersten "sudo" wird noch dein Passwort verlangt.

Beitrag von „daniel14513“ vom 10. Dezember 2015, 15:39

Danke für die Info`s.. es hat super geklappt und mein Mac rennt wie nie zuvor!! Tausend Dank!

Beitrag von „Adnarel“ vom 5. Januar 2016, 19:13

Der Ursprungspost könnte doch mal in die FAQs, zum bessern Wiederfinden?

Beitrag von „griven“ vom 5. Januar 2016, 23:55

Recht hast Du, habe es mal eben erledigt ist nun auch hier zu finden: [SMBIOS mit Ozmosis richtig setzen - jeder Rechner ein Unikat](https://www.hackintosh-forum.de/forum/thread/22640-smbios-mit-oz-1479-richtig-setzen-jeder-rechner-ein-unikat/)