

Booteinträge bei OpenCore

Beitrag von „pgr69“ vom 31. Oktober 2020, 11:08

Hallo Leute.

Ich benutze derzeit OpenCore 0.60 auf meinem Sage 10G Build und hätte eine Frage.

Ich habe zusätzlich zu der SSD mit macOS eine weitere mit Windows 10 und Ubuntu.

Kann ich manuell in der config.plist von OpenCore die Windows und die Ubuntu Partition auf dem Separaten Laufwerk so eintragen dass sie beim Boot mit OpenCore als Auswahlmöglichkeit zur Verfügung stehen ?

Wenn ja - wie mache ich das ?

Ig und Danke

Beitrag von „bluebyte“ vom 31. Oktober 2020, 11:40

Wenn in Misc -> Security -> [ScanPolicy](#) der Wert "0" eingetragen ist, dann werden alle startbaren Laufwerke und Volumes angezeigt. Ich rate aber davon ab, die Systeme über den OpenCore-Picker zu starten, wenn DSDT und SSDTs nicht sauber mit _OSI konfiguriert sind. Alle ACPI-Patches, die für den Betrieb von Mac OS notwendig sind, kämen sonst auch in Windows und Linux zum Einsatz. Beim Start von Windows könnte das zu einem Blue Screen führen mit der Meldung "ACPI-Error". Außerdem hätte der PC dann den wunderschönen Namen "acidanthera". Mit Linux habe ich das noch nicht probiert.

Die Werte für [ScanPolicy](#) sind in der Anleitung ausführlich dokumentiert. Den entsprechenden Wert erhält man durch das aufaddieren der einzelnen Bits und das Umwandeln in dezimal. Also braucht man dafür einen Taschenrechner mit "Programmiermodus". Hört sich jetzt wild an, ist aber halb so schlimm.

Beitrag von „pgr69“ vom 31. Oktober 2020, 11:46

Was bedeutet das mit dem _OSI oder wo kann ich das nachlesen?

Beitrag von „bluebyte“ vom 31. Oktober 2020, 11:49

Das Thema wurde hier schon x-mal diskutiert. Einfach mal die Suchfunktion bemühen.

Ich starte meine verschiedenen Systeme über die Auswahl im BIOS. Weil ich, ehrlich gesagt, keinen Bock dazu habe, diese _OSI-Weichen zu konfigurieren. Bei Gigabyte-Boards in den meisten Fällen "F12".

https://www.hackintosh-forum.de/t/1631156/?highlight=_OSI

Der apfelnico oder griven können das gut!

So lange du keine wilden ACPI-Patches gemacht hast, die Windows ins Gehege kommen, kann im **NORMALFALL** nicht viel passieren. Außer das Windows beim nächsten Start die Reparatur startet.

Ist mir auch schon passiert.

Beitrag von „pgr69“ vom 31. Oktober 2020, 12:01

Du wirst es nicht glauben - ich habe nach Open Core und booten und Partitionen gesucht und nix gefunden.

Beim Asus ist es F8 das geht ja auch aber eleganter ist es doch das direkt über Open Core zu mache .

Dein Link funktioniert nicht

Beitrag von „bluebyte“ vom 31. Oktober 2020, 12:10

Nachtrag zu vorhin: das Umrechnen geht mit der Doku und der Calculator.App ganz easy.

Erweiterte Suche oder Suche nach "_OSI" oder "OSI".

[ASUS X550L Hackintosh](#)

Beitrag von „5T33Z0“ vom 31. Oktober 2020, 12:26

OpenCore Configurator hat einen [Scanpolicy](#) Calculator eingebaut. Macht's ein bisschen einfacher. ich würde es immer auf 0 stehen lassen.

[bluebyte](#) ich habe mich auch mal an dieser _OSI-Methode versucht neulich, aber ich komme damit nicht weiter: [Benötige Hilfe: Laptop DSDT für Dualboot mit Windows anpassen \(OSI-Methode\)](#)

Beitrag von „bluebyte“ vom 31. Oktober 2020, 12:30

5T33Z0 hast Recht, geht damit einfacher. Benutze heute noch den Clover Configurator zum Erzeugen des SMBIOS. Gibt nichts Besseres.

Beitrag von „cobanramo“ vom 31. Oktober 2020, 16:15

Meine Empfehlung ist den beiliegenden SSDT-OC-XOSI.aml im Config unter Acpi einzubinden und unter patch den rename anzuwenden, damit ist das problem behoben und benötigt kein hantieren mit Bios boot Menü.

Allerdings wird auf diese weise unter Windows oder Linux die Hardware auch so erkannt wie unter SMBios definiert wurde.

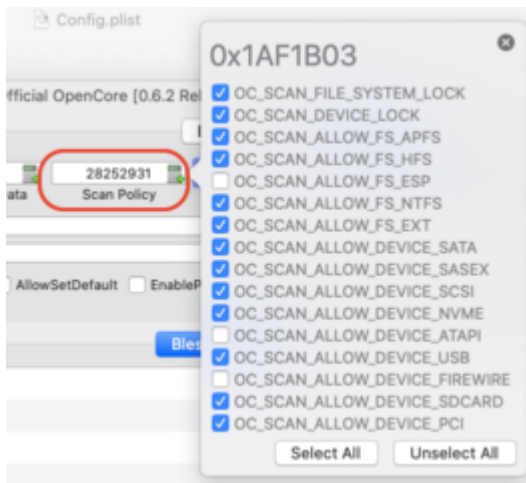
ACPI	Dictionary	3	3	Schlosser/Boot-Pass
+ Add	Array	3	4	generelle Comments
+ 0	Dictionary	3	3	Schlosser/Boot-Pass
+ 1	Dictionary	3	3	Schlosser/Boot-Pass
+ 2	Dictionary	3	3	Schlosser/Boot-Pass
+ 3	Dictionary	3	3	Schlosser/Boot-Pass
+ 4	Dictionary	3	3	Schlosser/Boot-Pass
+ 5	Dictionary	3	3	Schlosser/Boot-Pass
+ 6	Dictionary	3	3	Schlosser/Boot-Pass
+ 7	Dictionary	3	3	Schlosser/Boot-Pass
+ 8	Dictionary	3	3	Schlosser/Boot-Pass
+ 9	Dictionary	3	3	Schlosser/Boot-Pass
+ 10	Dictionary	3	3	Schlosser/Boot-Pass
+ 11	Dictionary	3	3	Schlosser/Boot-Pass
+ 12	Dictionary	3	3	Schlosser/Boot-Pass
+ 13	Dictionary	3	3	Schlosser/Boot-Pass
+ 14	Dictionary	3	3	Schlosser/Boot-Pass
+ 15	Dictionary	3	3	Schlosser/Boot-Pass
+ 16	Dictionary	3	3	Schlosser/Boot-Pass
+ 17	Dictionary	3	3	Schlosser/Boot-Pass
+ 18	Dictionary	3	3	Schlosser/Boot-Pass
+ 19	Dictionary	3	3	Schlosser/Boot-Pass
+ 20	Dictionary	3	3	Schlosser/Boot-Pass
+ 21	Dictionary	3	3	Schlosser/Boot-Pass
+ 22	Dictionary	3	3	Schlosser/Boot-Pass
+ 23	Dictionary	3	3	Schlosser/Boot-Pass
+ 24	Dictionary	3	3	Schlosser/Boot-Pass
+ 25	Dictionary	3	3	Schlosser/Boot-Pass
+ 26	Dictionary	3	3	Schlosser/Boot-Pass
+ 27	Dictionary	3	3	Schlosser/Boot-Pass
+ 28	Dictionary	3	3	Schlosser/Boot-Pass
+ 29	Dictionary	3	3	Schlosser/Boot-Pass
+ 30	Dictionary	3	3	Schlosser/Boot-Pass
+ 31	Dictionary	3	3	Schlosser/Boot-Pass
+ 32	Dictionary	3	3	Schlosser/Boot-Pass
+ 33	Dictionary	3	3	Schlosser/Boot-Pass
+ 34	Dictionary	3	3	Schlosser/Boot-Pass
+ 35	Dictionary	3	3	Schlosser/Boot-Pass
+ 36	Dictionary	3	3	Schlosser/Boot-Pass
+ 37	Dictionary	3	3	Schlosser/Boot-Pass
+ 38	Dictionary	3	3	Schlosser/Boot-Pass
+ 39	Dictionary	3	3	Schlosser/Boot-Pass
+ 40	Dictionary	3	3	Schlosser/Boot-Pass
+ 41	Dictionary	3	3	Schlosser/Boot-Pass
+ 42	Dictionary	3	3	Schlosser/Boot-Pass
+ 43	Dictionary	3	3	Schlosser/Boot-Pass
+ 44	Dictionary	3	3	Schlosser/Boot-Pass
+ 45	Dictionary	3	3	Schlosser/Boot-Pass
+ 46	Dictionary	3	3	Schlosser/Boot-Pass
+ 47	Dictionary	3	3	Schlosser/Boot-Pass
+ 48	Dictionary	3	3	Schlosser/Boot-Pass
+ 49	Dictionary	3	3	Schlosser/Boot-Pass
+ 50	Dictionary	3	3	Schlosser/Boot-Pass
+ 51	Dictionary	3	3	Schlosser/Boot-Pass
+ 52	Dictionary	3	3	Schlosser/Boot-Pass
+ 53	Dictionary	3	3	Schlosser/Boot-Pass
+ 54	Dictionary	3	3	Schlosser/Boot-Pass
+ 55	Dictionary	3	3	Schlosser/Boot-Pass
+ 56	Dictionary	3	3	Schlosser/Boot-Pass
+ 57	Dictionary	3	3	Schlosser/Boot-Pass
+ 58	Dictionary	3	3	Schlosser/Boot-Pass
+ 59	Dictionary	3	3	Schlosser/Boot-Pass
+ 60	Dictionary	3	3	Schlosser/Boot-Pass
+ 61	Dictionary	3	3	Schlosser/Boot-Pass
+ 62	Dictionary	3	3	Schlosser/Boot-Pass
+ 63	Dictionary	3	3	Schlosser/Boot-Pass
+ 64	Dictionary	3	3	Schlosser/Boot-Pass
+ 65	Dictionary	3	3	Schlosser/Boot-Pass
+ 66	Dictionary	3	3	Schlosser/Boot-Pass
+ 67	Dictionary	3	3	Schlosser/Boot-Pass
+ 68	Dictionary	3	3	Schlosser/Boot-Pass
+ 69	Dictionary	3	3	Schlosser/Boot-Pass
+ 70	Dictionary	3	3	Schlosser/Boot-Pass
+ 71	Dictionary	3	3	Schlosser/Boot-Pass
+ 72	Dictionary	3	3	Schlosser/Boot-Pass
+ 73	Dictionary	3	3	Schlosser/Boot-Pass
+ 74	Dictionary	3	3	Schlosser/Boot-Pass
+ 75	Dictionary	3	3	Schlosser/Boot-Pass
+ 76	Dictionary	3	3	Schlosser/Boot-Pass
+ 77	Dictionary	3	3	Schlosser/Boot-Pass
+ 78	Dictionary	3	3	Schlosser/Boot-Pass
+ 79	Dictionary	3	3	Schlosser/Boot-Pass
+ 80	Dictionary	3	3	Schlosser/Boot-Pass
+ 81	Dictionary	3	3	Schlosser/Boot-Pass
+ 82	Dictionary	3	3	Schlosser/Boot-Pass
+ 83	Dictionary	3	3	Schlosser/Boot-Pass
+ 84	Dictionary	3	3	Schlosser/Boot-Pass
+ 85	Dictionary	3	3	Schlosser/Boot-Pass
+ 86	Dictionary	3	3	Schlosser/Boot-Pass
+ 87	Dictionary	3	3	Schlosser/Boot-Pass
+ 88	Dictionary	3	3	Schlosser/Boot-Pass
+ 89	Dictionary	3	3	Schlosser/Boot-Pass
+ 90	Dictionary	3	3	Schlosser/Boot-Pass
+ 91	Dictionary	3	3	Schlosser/Boot-Pass
+ 92	Dictionary	3	3	Schlosser/Boot-Pass
+ 93	Dictionary	3	3	Schlosser/Boot-Pass
+ 94	Dictionary	3	3	Schlosser/Boot-Pass
+ 95	Dictionary	3	3	Schlosser/Boot-Pass
+ 96	Dictionary	3	3	Schlosser/Boot-Pass
+ 97	Dictionary	3	3	Schlosser/Boot-Pass
+ 98	Dictionary	3	3	Schlosser/Boot-Pass
+ 99	Dictionary	3	3	Schlosser/Boot-Pass
+ 100	Dictionary	3	3	Schlosser/Boot-Pass

Gruss Coban

EDit:

Mit [Scanpolicy](#) 0 gibt es wiederum andere probleme, da können auch unerwünschte Einträge auftauchen.

Mit dem Eintrag [ScanPolicy](#)=28252931 erzielte ich für meine Bedürfnisse das beste Ergebnis.



Beitrag von „bluebyte“ vom 31. Oktober 2020, 16:15

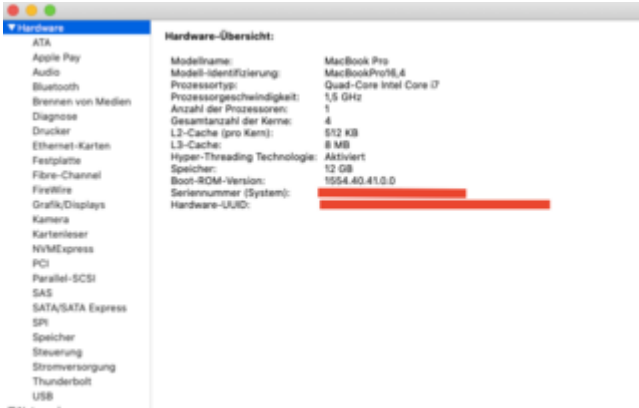
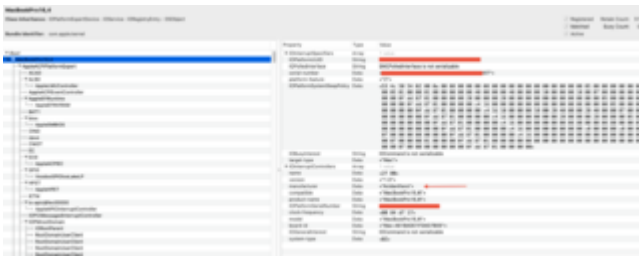
[cobanramo](#) ist es noch heute so, dass der Rechner bei Automatik noch acidanthera heißt, es sei denn man deaktiviert die Automatik/Generic und füllt die Felder selbst aus?

Beitrag von „cobanramo“ vom 31. Oktober 2020, 16:32

Als Systemhersteller wird es als acidanthera definiert ja, aber kannst ja den auch anpassen wenn es anders heissen soll. Laut Handbuch soll wird es ja auch nicht empfohlen den als Apple zu definieren.

```
1. SpoofVendor
Type: plist boolean
Fallback: false
Description: Sets SMBIOS vendor fields to Acidanthera.
It is dangerous to use Apple in SMBIOS vendor fields for reasons given in SystemManufacturer description.
However, certain firmwares may not provide valid values otherwise, which could break some software.
```

Hier bei mir sieht es so aus..



Unter Windows sieht es dann so aus.



Gruss Coban

Beitrag von „pgr69“ vom 31. Oktober 2020, 23:30

Hallo

[cobanramo](#)

Du scheinst Dich damit ja ganz gut auszukennen - darf ich ein paar Fragen stellen....

1. Der RENAME benennt `_OSI` zu `XOSI` um. Was ist `_OSI`?? Eine Funktion, eine Systemvariable oder was?
2. Wenn ich mit meinen bescheidenen Programmierkenntnissen die SSDT lese sieht das für mich so als würde die definierte Methode `XOSI` überprüfen ob in `_OSI` "Darwin" steht...falls nicht gibt sie das ihr übergebene Argument als Werte zurück. Ist es aber "Darwin" dann wird geprüft ob das übergebene Argument "Windows 2015" und entsprechend FFFFFFFF oder Zero zurückgegeben. Stimmt das so ? Aber warum Windows 2015? Woher kommt die 2015 ?
3. Bei der ScanPolicy kann ich doch SAS, SATA, SDCARD weg lassen oder ? Denn ich werde von keinem dieser Medien booten. Oder gibt es andere Gründe die drin zu lassen ?

Ig und Danke

Beitrag von „griven“ vom 1. November 2020, 00:03

Ich bin zwar nicht [cobanramo](#) aber kann vielleicht auch die eine oder andere Frage beantworten 😊

1. `_OSI` steht für "Operation System Interface" und ist eine Methode im ACPI die dazu dient zu erkennen welche Funktionen ein Betriebssystem unterstützt oder eben auch nicht. Im ACPI eines Rechners werden so zum Beispiel bestimmte Features nur für bestimmte Betriebssysteme bereitgestellt man erkennt das im AML Code dann an Konstrukten die in etwa so aussehen `if(_OSI("Darwin"))` oder `if(_OSI("Windows_2013"))`. Der Patch benennt alle Aufrufe der `_OSI` Methode in `XOSI` um und deaktiviert diese damit faktisch. Gemacht wird das um etwaige Einschränkungen die durch die `_OSI` Methode ggf. gegeben sind aufzuheben bzw. wie im gegebenen Beispiel diese Aufrufe auf die in der SSDT definierte Methode `XOSI` umzuleiten.

2. Fast richtig. Die `XOSI` Methode bekommt bei Ihrem Aufruf ein Argument mitgegeben als Argument kommt hier das mit was vor dem rename im Original mal da gestanden hat. Der

rename sorgt ja dafür das jedes vorkommen von _OSI durch XOSI ersetzt wird so würde aus einem _OSI("Windows-2015") ein XOSI("Windows-2015") werden wobei das Argument Windows-2015 eben an die XOSI Methode durchgereicht wird. Die Methode in der SSDT prüft jetzt im ersten Aufschlag ob es sich beim Inhalt von _OSI um macOS handelt (Darwin) wobei dieser Wert von macOS selbst ans ACPI übergeben wird und sofern es sich um macOS handelt wird im weiteren geprüft ob das übergebene Argument "Windows-2015" (Steht für Windows 10) ist und falls dem so ist gibt XOSI den Wert 0xFFFFFFFF zurück ansonsten halt ZERO. Sollte nicht Darwin gestartet worden sein wird von XOSI einfach das Argument zurückgegeben womit die Methode aufgerufen wurde.

3. Kannst Du ohne weiteres weglassen wenn Du diese Medien Formen nicht nutzt oder als Bootlaufwerk verwenden möchtest.

Ig und Bitteschön 😊

Beitrag von „cobanramo“ vom 1. November 2020, 00:09

Also das letzte zuerst, [ScanPolicy](#) ist natürlich für die eigene Bedürfnisse anzupassen, wenn du kein SAS oder SDCard usw. hast brauchst du diesen Bit auch nicht konfigurieren.

Andererseits sieht es so aus, [ScanPolicy](#) auf 0 zu stellen und somit **komplett auszuschalten ist nicht grad von Vorteil**, somit wird je nach installation (MultiBoot, Windows, Linux usw.) zbspl. unbenötigte Einträge auch mehrfach aufgelistet oder einfach alles in einem Haufen dargestellt.

Ich persönlich Steuer mit diesem Wert halt umgekehrt, es soll alles scannen aber das was ich nicht benötige ausschliessen.

Die _OSI zu XOSI geschichte beruht auf die [Acpi Call von Microsoft Windows](#), wurde glaub zum ersten mal von Rehabman aktiv eingesetzt für USBInject und patching.

Mit der Methode kann man eben abfragen ob "Windows" oder "Darwin" (MacOs) gestartet wurde somit eben auch mitgeben wenn "Darwin" am start wert x ansonsten halt y usw.

Das *Arg0* == "Windows 2015" basiert auch auf die Microsoft Acpi Dokumentation im obigen link, bedeutet halt das es ein Windows 10 ist.

Bspl.

Table 1. OS Strings for Windows Operating Systems

Windows operating system version	OSI string
Windows 10, version 1703	"Windows 2017"
Windows 10, version 1607	"Windows 2016"
Windows 10	"Windows 2015"
Windows Server Technical Preview	"Windows 2015"
Windows 8.1	"Windows 2013"
Windows Server 2012 R2	"Windows 2013"
Windows 8	"Windows 2012"
Windows Server 2012	"Windows 2012"
Windows 7	"Windows 2009"
Windows Server 2008 R2	"Windows 2009"
Windows Vista	"Windows 2006"
Windows Vista SP1	"Windows 2006 SP1"
Windows Server 2008	"Windows 2006.1"
Windows XP	"Windows 2001"
Windows XP SP2	"Windows 2001 SP2"
Windows Server 2003	"Windows 2001.1"
Windows Server 2003 SP1	"Windows 2001.1 SP1"

Gruss Coban

Edit: Bin nicht grad der Schreiber, [griven](#) mag das bestimmt besser rüberbringen.

Danke Bert 😊

Beitrag von „pgr69“ vom 1. November 2020, 00:14

Hallo ihr Beiden [griven](#) und [cobanramo](#)

Perfekt - danke für die tolle Erklärungen, so kann man das verstehen und es macht Spaß dazu zulernen.

Aber ein Frage dazu hätte ich jetzt noch.....wenn im Rename ALLE Aufrufe von _OSI in XOSI umbenannt werden betrifft das dann aber nicht die SSDT sonst könnte die darin definierte neue Methode XOSI ja Ihrerseits nicht die Original Methode _OSI benutzen.....richtig ?

Ig und nochmal DANKE !

Beitrag von „griven“ vom 1. November 2020, 00:18

Genau die Reihenfolge dabei ist immer wie folgt:

- ggf. definierte ACPI Tabellen dropen.
- Patches anwenden.
- ACPI Tabellen die geladen werden sollen (load) laden und ggf. vorhandene gegen die geladenen ersetzen.

Damit ist dann auch sichergestellt das sich Patches nicht auf die custom erzeugten Tabellen auswirken.

Beitrag von „cobanramo“ vom 1. November 2020, 00:19

Den rename brauchst du damit eben ein neues "zustand" im ACPI erhältst, du kannst nicht ein vorhandenes sagen wir mal "_OSI" einfach so verändern. Benennst den eben zu "XOSI" und verpasst den einen neuen zustand.

Gruss Coban

Beitrag von „bluebyte“ vom 1. November 2020, 09:51

Hier ein Link zu Microsoft mit den aktuellen _OSI-Einträgen

[_OSI Strings for Windows Operating Systems](#)

Beitrag von „cobanramo“ vom 1. November 2020, 13:37

In unserem Fall mit OC spielt bei dem vorgehen mit dem SSDT genaugenommen keine rolle welche Version vom Windows angegeben wurde, nach der Übergabe des "wertes" übernimmt das die Windows selber was es zutun hat.

Das mit aktuellem oder verschiedenen versionen vom Windows kommt zu tragen wenn du im Firmware "funktionen oder Werte" bereitstellen willst die halt bsplw. bei Windows 7 anders als bei Windows 10 oder Server Systemen behandelt werden.

Gruss Coban

Quelle: [Advanced Configuration and Power Interface \(ACPI\) Specification](#) (search _OSI)

Beitrag von „Harry69“ vom 1. November 2020, 16:20

Hallo miteinander,

ich hänge mich mal hier mit meiner Frage an diesen Thread da es im weitesten Sinne auch mit dem Booten und Windows 10 zu tun hat.

System ist aktuelles Catalina im Dual Boot mit aktuellem Windows 10 auf seperaten SSD.

Bootloader ist Open Core 0.6.2.

Ich denke beide Systeme und OpenCore sind sauber konfiguriert. Beide Betriebssysteme starten sauber und laufen ohnen Probleme oder abstürze flott und stabil.

Nun ist mir unter Windows 10 unter der Systeminformation aufgefallen das unter

Systemhersteller: Acidanthera

Systemmodell: iMacPro 1,1 usw. steht (siehe Anhang)

Frage: Ist dies nur ein tolerierbarer kosmetischer Fehler durch das Booten von Windows 10 mit OpenCore oder steckt da doch ein größerer Fehler in der Konfiguration von OC dahinter?

Wie kann man das beheben?

_OSI Einträge wären eingefügt..

Wie gesagt läuft alles sauber und stabil und ohne Probleme..., aber nerft trotzdem..

PS: Was haltet ihr von ReFind als Bootmanager vor OpenCore als Bootloader gesetzt, das würde das Problem ja auch beheben..

Beitrag von „bluebyte“ vom 1. November 2020, 16:26

[Harry69](#) siehe Kommentar von cobanramo. Alles gut, habe ich auch. Um das zu ändern müsstest du die Automatik/Generic deaktivieren und die komplette Platforminfo mit den zig Einträgen von Hand konfigurieren.

[Booteinträge bei OpenCore](#)

Beitrag von „karacho“ vom 1. November 2020, 16:33

Mal was zu [ScanPolicy=0](#)

Damit wird kein Linux angezeigt. Dazu ist ein Eintrag unter Misc->Entries nötig.

Beitrag von „bluebyte“ vom 1. November 2020, 16:39

[karacho](#) bei mir wird Windows nur angezeigt, wenn [ScanPolicy](#) auf "0" steht.

Beitrag von „karacho“ vom 1. November 2020, 16:46

Das war nicht meine Aussage Steffen. Mit [ScanPolicy](#)=0 wird Windows ja auch angezeigt. Aber ein Linux bootloader wird nicht erkannt. Ich habe [ScanPolicy](#) auf dem Standard Wert gelassen und für Windows und Linux jeweils einen Eintrag in den Entries gemacht, den ich bei Bedarf mit der Leertaste beim booten einblenden kann.

Beitrag von „bluebyte“ vom 1. November 2020, 16:47

[karacho](#) es bezog sich ja nicht auf deine Aussage, sondern es war insgeheim ein Hilferuf

Beitrag von „karacho“ vom 1. November 2020, 16:48

Alles gut. Ich wollte auch nur darlegen, dass der TE mit [ScanPolicy](#)=0 kein Linux finden wird

Beitrag von „bluebyte“ vom 1. November 2020, 16:51

War wohl mein Fehler, wenn ich ESP aus der Suche ausschließe, oder?

Dann such er ja nicht Windows auf einer anderen Platte, oder?

Beitrag von „karacho“ vom 1. November 2020, 17:00

Beim Wert 0 wird alles gescannt, auch alle ESPs und was darin zum booten gefunden wird. Nur halt kein Linux. Den Pfad zur grubx64.efi muss man manuell bei Misc->Entries eintragen.

Beitrag von „cobanramo“ vom 1. November 2020, 17:53

[Zitat von karacho](#)

Mal was zu [ScanPolicy=0](#)

Tut mir leid wenn ich mit meiner Aussage "Linux" Verwirrung gestiftet haben sollte,

OC scannt keine Linux, da hat Karacho recht, meine aussage sollte eher alles andere beinhalten.

Für den Linux braucht es einen manuellen Eintrag unter "Misc/Entries"

Die falsch aussage basiert auf meinem ungewöhnlichem Umgebung, ich persönlich binde zusätzlich zbspl. ntf.s.efi um externe Medien unter anderem Windows Installer oder WinPE zu starten, alles unter OC.

Werde mein aussage dort korrigieren.

Beispiel für einen Linux Eintrag;

```
PciRoot(0x0)/Pci(0x1D,0x4)/Pci(0x0,0x0)/NVMe(0x1,88-CF-C0-91-3B-E4-D2-5C)/HD(1,GPT,658F6EE5-AD6D-41D3-843D-BBD0F4ADC6CD,0x800,0x82000)\EFI\Ubuntu\shimx64.efi
```

Den Pfad für euren müsst Ihr selber herausfinden, Siehe Handbuch, man kann nicht einfach übernehmen.

Gruss Coban



Beitrag von „ozw00d“ vom 1. November 2020, 18:11

Hm sehr interessant, hab nicht einen _OSI eintrag hinterlegt. W10 rennt als gäbe es kein Morgen mit OC.

Beitrag von „badbrain“ vom 1. November 2020, 18:37

[Zitat von cobanramo](#)

Die falsch aussage basiert auf meinem ungewöhnlichem Umgebung, ich persönlich binde zusätzlich zbspl. ntfs.efi um externe Medien unter anderem Windows Installer oder WinPE zu starten, alles unter OC.

So ungewöhnlich ist die Umgebung gar nicht und ich frage mich, wozu man da die ntfs.efi benötigt - funktioniert bei mir auch ohne. WinPE auf FAT32 (habe dafür einen Eintrag in OC) und fertig. Windows Installer einbinden braucht man doch nicht, wenn man WinPE hat. Ich installiere Windows immer von meinem WinPE aus. Heißt ja nicht umsonst Preinstallation Environment (Vorinstallationsumgebung). Dazu kommt dann nur noch eine install.wim / -.esd und [WinNTSetup](#), mein bevorzugtes Tool für die Windows-Installation mit dem man z. B. die für Windows vorgesehene EFI-Partition bestimmen und noch einiges mehr machen kann.

Beitrag von „karacho“ vom 1. November 2020, 19:25

[cobanramo](#) Die Aussage zu Linux war doch nicht von dir. Der TE hat das in seinem ersten Posting gefragt. Daher schrieb ich überhaupt auf das, auf [ScanPolicy](#) bezogene Statement, das

man mit '0' kein Linux angezeigt bekommt. Von daher, alles gut Jong.

Beitrag von „cobanramo“ vom 1. November 2020, 21:26

[Zitat von badbrain](#)

funktioniert bei mir auch ohne.

Klar, ne lösung gibt es immer, ich hantiere da nicht gross rum. verschiebe einfach die Daten in den Partition und schon hab ich das was ich brauche, mit Fat32 müsste ich entweder den wim splitten oder den Medium verändern. Mit NTFS.efi werden eben die erstellten Ntfs Partitionen mit den Installationsmedien auch gescannt. Per Fat32 kann ich sogar Linux Live Systeme starten wen ich sie mal brauche.

Wird aber glaub hier so langsam offtopic.



Gruss Coban

Beitrag von „griven“ vom 1. November 2020, 21:43

[Zitat von Harry69](#)

Hallo miteinander,

ich hänge mich mal hier mit meiner Frage an diesen Thread da es im weitesten Sinne auch mit dem Booten und Windows 10 zu tun hat.

System ist aktuelles Catalina im Dual Boot mit aktuellem Windows 10 auf separaten SSD.

Bootloader ist Open Core 0.6.2.

Ich denke beide Systeme und OpenCore sind sauber konfiguriert. Beide Betriebssysteme starten sauber und laufen ohne Probleme oder Abstürze flott und stabil.

Nun ist mir unter Windows 10 unter der Systeminformation aufgefallen das unter

Systemhersteller: Acidanthera

Systemmodell: iMacPro 1,1 usw. steht (siehe Anhang)

Frage: Ist dies nur ein tolerierbarer kosmetischer Fehler durch das Booten von Windows 10 mit OpenCore oder steckt da doch ein größerer Fehler in der Konfiguration von OC dahinter?

Wie kann man das beheben?

_OSI Einträge wären eingefügt..

Wie gesagt läuft alles sauber und stabil und ohne Probleme..., aber nerft trotzdem..

PS: Was haltet ihr von ReFind als Bootmanager vor OpenCore als Bootloader gesetzt, das würde das Problem ja auch beheben..

Alles anzeigen

Ich zitiere entgegen meiner üblichen Gepflogenheiten ausnahmsweise mal den gesamten Beitrag einfach weil er in der Diskussion zu dem anderen Thema in diesem Thread untergegangen ist und nun irgendwo gefühlte zwei Seiten weiter vorne verwaist dasteht und vermutlich niemand mehr einen Zusammenhang zusammenbringt wenn ich jetzt auf die eigentliche Fragestellung eingehe 😊

[Harry69](#) das was Du beobachtet hat nichts mit dem ACPI bzw. der _OSI Methode zu tun sondern hängt mit dem SMBIOS (System Management Bios Spezifikation) zusammen und der Art und Weise wie OpenCore diese Spezifikation interpretiert. Das Betriebssystem bezieht die

Informationen darüber auf welcher Plattform es läuft via DMI (Desktop Management Interface) aus dem SMBIOS. Im SMBIOS sind vom Hersteller des Mainboards bzw. vom Zulieferer der Firmware unter anderem Informationen wie der Name des Mainboards, die Versionsnummer des eingesetzten Chipsatzes und die Hardware UUID abgelegt wobei diese Werte in der Regel nicht durch den Benutzer änderbar sind und somit zumindest in der Theorie jedem Mainboard einen einmaligen Fingerabdruck verpassen. Open Core sitzt als Mittler zwischen der Firmware des Rechners auf der einen und dem Betriebssystem auf der anderen Seite und nimmt in dieser Position einige zum Teil tiefgreifende Änderungen an bestimmten Daten vor um eine relative macOS Kompatibilität zu erreichen. Eine dieser Änderungen betrifft auch das SMBIOS bzw. eben die Informationen welche über die Hardware an das Betriebssystem via DMI übergeben werden. Je nach Einstellung überschreibt OpenCore die vom Hersteller bereitgestellten Informationen mit solchen die macOS kompatibel sind um einen Start von macOS überhaupt erst zu ermöglichen. Auch wenn es eigentlich kein Problem ist das Windows hier ein Mac untergeschoben wird mag es den einen oder andern vielleicht stören bzw. mag es sogar sein das Windows seine Aktivierung vergisst (passiert gerne wenn Windows via SLIC aus der Firmware aktiviert wurde und nun die Informationen nicht mehr zu der SLIC Aktivierung passen). Für diese Fälle kann man OpenCore so definieren das sich die Änderungen am SMBIOS nicht global auswirken sondern sich auf macOS beschränken. Erreicht wird das wie folgt:

1. Im Bereich Kernel -> Quirks wird der Quirk `CustomSMBIOSGuid` auf `enabled` (Boolean Wert `true`) gesetzt
2. Im Bereich PlattformInfo wird für den Key `UpdateSMBIOSMode` der Wert auf `Custom` gesetzt

Ist beides erledigt wird die `config.plist` gespeichert und nach einem reboot steht das Apple SMBIOS nur noch für macOS zur Verfügung alle anderen Betriebssysteme werden wieder mit den Werten versorgt die sie über das DMI aus dem SMBIOS des Boards auslesen können. Einmal richtig eingestellt verhält sich die Kiste auch so wie man es erwarten würde und das ganz ohne abstruse Chainloader Konstrukte über `reFind` oder ähnliche Stunts 😊

Beitrag von „Harry69“ vom 1. November 2020, 21:43

[cobanramo](#) warum sollte ich unter Misc->Entries-> 0 den Pfad auf EFI->Microsoft.....angeben.

Ist das wegen der externen Installations Mediensammlung?

Bei mir steht dort kein einziger Eintrag, und ich kann trotzdem ganz normal Windows über den OpenCore Bootloader starten.

Was bringt es also wenn dort ein Eintrag zu Windows gemacht wird?

[griven](#) Selten so einen kompetenten auf den Punkt gebrachten Lösungsvorschlag mit kurzer Erklärung gesehen.

Was soll ich sagen. Zwei Tage Zeit vergeudet um nach OC und Apple kompatiblen Bootmanagern zu suchen weil mich das extrem genervt hatte.

Nun zwei Variablen innerhalb von einer Minute geändert und es funktioniert alles wie es soll.

Ein ganz dickes Daumen hoch..

Beitrag von „cobanramo“ vom 1. November 2020, 21:59

Ohne diesen Eintrag wird Windows gescannt und "links" vor den MacOSs dargestellt, mit dem Eintrag sind es "rechts" nach dem MacOSs. Ist für mich nur was visuelles, mit dem Eintrag kann ich so wie ich es gerne hätte den Position sowie Label bestimmen. Mehr nicht.

Externe & gescannte Einträge füllen die "linke" Seite des Bildschirms. Siehe Bild oben.

Gruss Coban

Beitrag von „bluebyte“ vom 2. November 2020, 09:55

[griven](#) vielen Dank für die Tipps. Da muss man erst mal drauf kommen.

Habe eben die Dokumentation gelesen.

Bei **CustomSMBIOSGuid** steht folgender Hinweis für **Enabled**

Normalerweise relevant für Dell-Laptops"

Bei **UpdateSMBIOSMode** steht folgender Hinweis für **Custom**

Hinweis: Ein Nebeneffekt bei der Verwendung des benutzerdefinierten Ansatzes besteht darin, dass SMBIOS-Updates exklusiv für macOS verfügbar sind, wodurch eine Kollision mit der vorhandenen Windows-Aktivierung und der benutzerdefinierten OEM-Software vermieden wird, möglicherweise jedoch Apple-spezifische Tools beschädigt werden.

Der letzte Hinweis mit den Apple-Tools gibt zu denken. Was genau kann da beschädigt werden?

Beitrag von „griven“ vom 2. November 2020, 10:01

Mir ist da bisher keines untergekommen und gerade auf dem Hackintosh hat das meiner Meinung nach auch keine wirkliche Relevanz. Der Kernel Quirk sorgt dafür das der Aufruf für über das DMI auf eine andere Adresse gebogen wird als die an der diese Informationen üblicherweise zur Verfügung stehen der Eintrag unter PlattformInfo sorgt dafür das OpenCore die Informationen auch an der Stelle zur Verfügung stellt. Wenn also irgendwelche Tools davon beeinflusst werden sollten dann nur solche die am Kernel vorbei auf die Informationen zugreifen und von denen wird es nicht besonders viele geben die für uns eine Relevanz hätten. Anders sieht das freilich aus wenn man OpenCore auf echter Apple Hardware verwendet hier sollte man dann von solchen Stunts vermutlich eher Abstand halten (mir fällt da zum Beispiel der EFI Firmware Updater ein der auf die Idee kommen könnte anhand der „falschen“ Informationen eine falsche Firmware auf die Kiste zu basteln).

Beitrag von „bluebyte“ vom 2. November 2020, 10:03

[griven](#) vielen Dank für die schnelle Antwort

Beitrag von „apfelnico“ vom 2. November 2020, 10:03

Denke da geht es um Treiber aus BootCamp, setzt ja einen echten Mac normalerweise voraus.

Beitrag von „griven“ vom 2. November 2020, 10:31

Stimmt die BootCamp Geschichte unter Windows wird dann natürlich auch nicht mehr funktionieren aber die wird am Hackintosh auch kaum jemand ernsthaft nutzen wollen immerhin sind ja die PC Kisten nativ für den Betrieb mit Windows gemacht *gg*

Beitrag von „bluebyte“ vom 2. November 2020, 11:25

Habe jetzt, dank der Infos hier, mein Windows ohne Probleme starten können.

Beitrag von „karacho“ vom 2. November 2020, 12:04

[Zitat von bluebyte](#)

Einziger Schönheitsfehler ist, dass ich die ESP unter [ScanPolicy](#) mit einbinden muss, damit Windows angezeigt wird.

Musst du doch nicht. Mach es doch einfach so wie ich schon schrieb, Steffen. [PCIROOT Pfade für Custom Entries für OpenCore heraus finden.](#)



Name	Value
PCIROOT	0
PCIROOT0	0
PCIROOT1	1
PCIROOT2	2
PCIROOT3	3
PCIROOT4	4
PCIROOT5	5
PCIROOT6	6
PCIROOT7	7
PCIROOT8	8
PCIROOT9	9
PCIROOT10	10
PCIROOT11	11
PCIROOT12	12
PCIROOT13	13
PCIROOT14	14
PCIROOT15	15
PCIROOT16	16
PCIROOT17	17
PCIROOT18	18
PCIROOT19	19
PCIROOT20	20
PCIROOT21	21
PCIROOT22	22
PCIROOT23	23
PCIROOT24	24
PCIROOT25	25
PCIROOT26	26
PCIROOT27	27
PCIROOT28	28
PCIROOT29	29
PCIROOT30	30
PCIROOT31	31

Beitrag von „bluebyte“ vom 2. November 2020, 12:18

[karacho](#) ich schätze deinen Ehrgeiz und ich weiß auch wie man das macht, aber im Moment habe ich keinen Bock solche Zahlen- und Buchstabenkolonnen einzugeben.

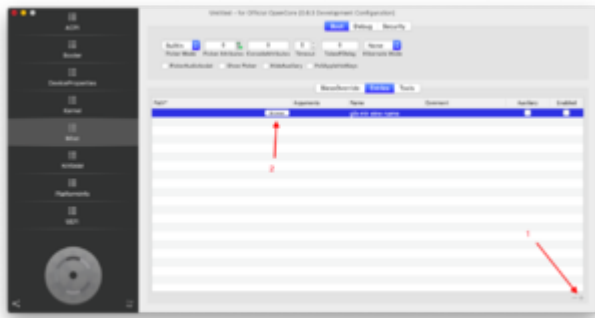
Maddeen hat da ein tolles Tutorial geschrieben. Natürlich wurdest du dort als Erfinder namentlich erwähnt.

Beitrag von „karacho“ vom 2. November 2020, 12:22

Ist doch nur Copy&Paste. 😊

Beitrag von „cobanramo“ vom 2. November 2020, 12:55

[bluebyte](#)



geht auch einfacher wenn schnell-schnell gehen soll 😊

EDIT: Die funktion scheint gebrickt zu sein. Gestern ging es bei mir noch, heute Nada..
Bin auch nicht dahintergekommen was es auf sich hat, also fehl information mit dem Tool.

Beitrag von „bluebyte“ vom 2. November 2020, 12:56

Vielleicht sollte ich auf englische Tastatur umstellen? 🤔

Beitrag von „badbrain“ vom 2. November 2020, 14:10

[cobanramo](#) Ok, ich dachte, dass du ein WinPE mit eingebunden hättest, aber oben sehe ich jetzt keins. Mit WinPE hättest du doch Zugriff auf alle Laufwerke (standardmäßig natürlich nur auf die Dateisysteme, die Windows kann, aber mit den Paragon Treibern habe ich z. B. auch Zugriff auf APFS, HFS und extFS) und müsstest keine .wim splitten oder einen Windows-Installer mit ntfs.efi im Bootmenu einbinden, weil du diesen ja aus dem PE heraus starten könntest.

Ich habe bei meinem System ein, ich nenne es mal Haupt-OpenCore mit Custom Entries für das, was ich öfter im Einsatz habe, was da macOS, Windows und WinPE wären, wobei WinPE erst nach drücken der SPACE-Taste zur Auswahl steht (Linux ist gerade weg - kommt aber wieder). Standardmäßig wird hier macOS meist ohne Anzeige des Pickers gebootet. Die anderen Systeme werden dann bei Bedarf mit Aufruf des Pickers mit ESC gestartet. Dazu habe ich für etwaige Notfälle in der EFI meiner Time Machine SSD noch ein zweites OpenCore mit [ScanPolicy](#) 0, aktiviertem Verbose, angepasste Werte in Misc/Debug sowie die .efi Dateien der entsprechenden OC Debug Version, welche dann, wenn nötig, zum Einsatz kommen. So handhabe ich es auch mit der EFI meines Install-Sticks, der natürlich immer die aktuell eingesetzte macOS Version enthält. Zu all dem hat man dann noch das Bios-Bootmenu.

Hier mal mein Picker, den ich eigentlich fast nicht zu Gesicht bekomme - kann ja, wenn zu sehr OT, in einen 'Show me your canopy' Thread verschoben werden. 😄



Picker:



Nach gedrückter SPACE-Taste:

Dies soll keinesfalls belehrend wirken, da ja jeder seine eigene Herangehensweise hat und auch haben soll. Hier wollte ich nur mal meine darlegen.

[karacho](#) [bluebyte](#) [cobanramo](#) Die Pfade für die Custom Entries kann man übrigens auch im Debug-Log finden.

Beitrag von „karacho“ vom 2. November 2020, 20:55

[Zitat von badbrain](#)

Die Pfade für die Custom Entries kann man übrigens auch im Debug-Log finden.

Klaro, aber nur wenn man mit der Debug Version startet und 0x67 bei Misc->Debug->Target drin hat. Und meinen Picker sehe ich auch nur, wenn ich das will 😊 Ansonsten...guter Beitrag 👍

Aber, verrate uns doch mal, wo deine Icons herkommen, oder noch besser, würdest du die mit der Community teilen? Die sehen echt geil aus.

Beitrag von „badbrain“ vom 3. November 2020, 09:01

[Zitat von karacho](#)

Klaro, aber nur wenn man mit der Debug Version startet und 0x67 bei Misc->Debug-

>Target drin hat.

Genau das (Target und DisplayLevel) meinte ich in diesem Satz:

Zitat

... ein zweites OpenCore mit [ScanPolicy](#) 0, aktiviertem Verbose, angepasste Werte in Misc/Debug sowie die .efi Dateien der entsprechenden OC Debug Version, ...

Alles andere ist kein richtiges Debug-Log.

Die Icons stammen von einem [User \(blackosx\) aus dem InsanelyMac-Forum](#). Man kann sie sich [bei GitHub herunterladen](#).

Beitrag von „kexterhack“ vom 2. Januar 2021, 19:57

[cobanramo](#) Nutze immer [ScanPolicy](#)=0 außer meinem Win/Hackintosh und USB Stick im Falle und NVME Reset kommen da keine großen weiteren Einträge bei mir.

Also wenn mich das nicht stört, kann ich das so lassen, oder hat das noch andere Probleme/Gründe?

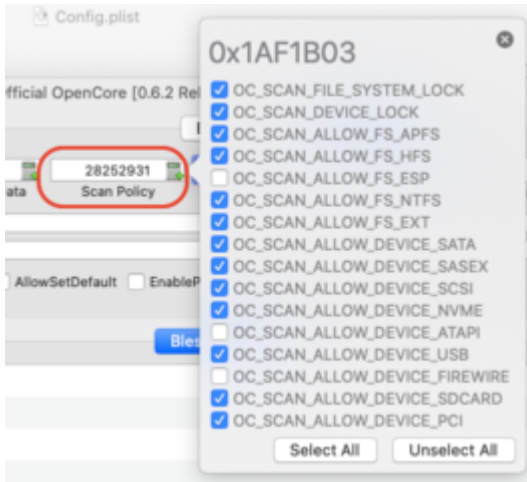
Habe gerade den ganzen Thread gelesen, aber sonst nichts gefunden, was daraufhin deutet.

Die Systeminfo ist bei mir geändert, sodass mein MB unter Win korrekt ausgelesen wird.

Beitrag von „cobanramo“ vom 2. Januar 2021, 20:11

Musste grad nochmals gucken warum ich das so mache 😄

Also lass mich so erklären wie ich auf so ein Wert komme,



ich lasse praktisch alles (modernere Möglichkeiten) Scannen was möglich ist, (grund: Ich nehme das meistens als Vorlage für alle meine Rechner) schließe aber gezielt den ESP/EFI aus.

Grund hierfür ist wiederum das ich den Windows & Linux zbspl. unter /Misc/Entries manuell eingetragen habe.

Somit werden bei mir immer zuerst alle MacOS Installationen angezeigt, danach Windows & Linux.

Ich steuere so direkt auch die Reihenfolge von den Betriebssystemen.

Klar muss man für sich und seine vorhanden Hardware Möglichkeiten selber was vernünftiges einrichten.

Bei [Scanpolicy](#) = 0 wirst du auch bemerkt haben das Windows "links" vom Boot Menü und MacOS rechts erscheint oder?

Genau das steuere ich damit auch noch 😊

Gruss Coban

Beitrag von „kexterhack“ vom 2. Januar 2021, 20:20

[cobanramo](#)

Danke für die schnelle und ausführliche Antwort.

Also nur visuelle Sachen weil du auch viele Optionen hast?!

Bluebyte verwirrte mich in dem Thread ein wenig wo er schrieb das es nicht gut wäre, da Crashes etc. kommen können.

MacOS und Win laufen bei mir aber top - mit [ScanPolicy](#)=0 keine Probleme

Übers Bios gehen zu müssen fände ich auch sehr umständlich.

Zu dem Boot Menü muss ich sagen, dass ich kein visuelles habe wie bei dir zu sehen ist, sondern nur ein einfaches Drop Down Menü mit Text. Windows ist immer an erster Stelle und der Hacki an zweiter.

Das ist so perfekt/ i.O für mich. Auch das Windows als Standardauswahl hinterlegt ist, da ich Win am meisten verwende.

Beitrag von „cobanramo“ vom 2. Januar 2021, 20:31

Genau so ja, bei mir ist eben auf verschiedenen Volume´s & Partitionen verschiedene Betriebssysteme.

Ist halt praktischer für mich wenn ich ne linie drinnen habe und das auf verschiedenen Rechnern immer gleich ist.

Ich blende auch die Recovery´s & Tool´s per "Hide Auxillary" aus.

Gruss Coban