

Lilu & Plugins mit Bootflags und Beispielen

Beitrag von „Noir0SX“ vom 11. Juni 2017, 12:32

Code

1. `<p>Beschreibung zu Lilu und seinen Plugins samt Hilfsklasse mit einzelnen Beispielen und Anwendungen seiner Bootflags</p>`
2. `<p></p>`
3. `<p>New Innie</p>`
4. `<p></p>`
5. `<ul style="list-style-type: disc" class="nativeList">`
6. `Übersicht`
7. `Features und Bootflags`
8. `Anwendung & Debug`
9. `Framebuffer-Patch`
10. `Beispiele`
11. `VirtualSMC`
12. `Plugin-Kext erstellen`
13. ``
14. `<table style="background-color:#F2F2F2" cellspacing="3" cellpadding="5"><caption>Grundvoraussetzung</caption>`
15. `<thead><tr bgcolor="#87cefa"><th>Name *</th><th>Kurze Beschreibung</th><th>Letzte Version</th><th>Kategorie</th><th>Bootflags</th></tr></thead>`
16. `<tbody>`
17. `<tr bgcolor="#dcdcdc"><!-- Lilu -->`
18. `<td bgcolor="#87cefa"> Lilu</td>`
19. `<td>Open-Source-Kernel-Erweiterung die eine Plattform für beliebige Kext, Bibliotheken und Programm-Patching im gesamten System für MacOS darstellt.</td>`
20. `<td align="center"></td><td>Kernel</td>`
21. `<td bgcolor="#87cefa"><a href="https://www.hackintosh-forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-`

```

Beispielen/#Lilu">hier</a></td>
22. </tr>
23. </tbody>
24. </table>
25. <p></p>
26. <table style="background-color:#F2F2F2" cellpadding="5"><caption><b>Plugins</b></caption>
27. <thead><tr bgcolor="#87cefa"><th>Name <span class="badge green">
* </span></th><th>Kurze Beschreibung</th><th>Letzte
Version</th><th>Kategorie</th><th>Pulls</th><th>Issues</th><th>Bootflags</th></tr></thead>
28. <tbody>
29. <tr><!-- AirportBrcmFixup -->
30. <td bgcolor="#87cefa"><a href="https://github.com/lvs1974/AirportBrcmFixup">
AirportBrcmFixup</a></td>
31. <td>Patches für Broadcom Airport Wi-Fi Karten</td>
32. <td align="center"></td>
33. <td>WiFi</td>
34. <td></td>
35. <td></td>
36. <td bgcolor="#87cefa"><a href="https://www.hackintosh-
forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-
Beispielen/#AirportBrcmFixup">hier</a></td>
37. </tr>
38. <tr bgcolor="#dcdcdc"><!-- AppleALC -->
39. <td bgcolor="#87cefa"><a href="https://github.com/acidanthera/AppleALC">
AppleALC</a></td>
40. <td>Native MacOS HD Audio für nicht offiziell unterstützte Codecs</td>
41. <td align="center"></td><td>Audio</td>
42. <td></td>

```

```

43. <td></td>
44. <td bgcolor="#87cefa"><a href="https://www.hackintosh-forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-Beispielen/#AppleALC">hier</a></td>
45. </tr>
46. <tr><!-- AMFIExemption -->
47. <td bgcolor="#87cefa"><a href="https://github.com/osy86/AMFIExemption">AMFIExemption </a></td><td> Fügt zulässige Berechtigungen zu nicht von Apple signierten Apps hinzu, wenn SIP aktiviert ist</td>
48. <td align="center"></td><td>Kernel</td>
49. <td></td> <td></td><td bgcolor="#87cefa"><a href="https://www.hackintosh-forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-Beispielen/#AMFIExemption">hier</a></td>
50. </tr>
51. <tr bgcolor="#dcdcdc"><!-- ATH9KFixup -->
52. <td bgcolor="#87cefa"><a href="https://github.com/chunnann/ATH9KFixup">ATH9KFixup </a> <span class="badge blue"> *</span></td><td>Patches für nicht unterstützte Atheros Wi-Fi Karten.</td>
53. <td align="center" style="color: red">v1.0.2 <i style="color:yellow"> *</i></td><td>Wifi</td>
54. <td></td> <td></td><td bgcolor="#87cefa"><a href="https://www.hackintosh-forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-Beispielen/#ATH9KFixup">hier</a></td>
55. </tr>
56. <tr><!-- BT4LEContiunityFixup -->
57. <td bgcolor="#87cefa"><a href="https://github.com/lvs1974/BT4LEContiunityFixup">BT4LEContiunityFixup</a></td>
58. <td>Aktivieren BT4LE-Handoff-Hotspot-Funktionen</td>
59. <td align="center"></td><td>Bluetooth</td>
60. <td></td> <td></td><td bgcolor="#87cefa"><a href="https://www.hackintosh-forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-Beispielen/#BT4LEContiunityFixup">hier</a></td>
61. </tr>
62. <tr bgcolor="#dcdcdc"><!-- CPUFriend -->

```

```

63. <td bgcolor="#87cefa"><a href="https://github.com/acidanthera/CPUFriend">
CPUFriend</a></td>
64. <td>Dynamische Energiemanagement für CPUs</td>
65. <td align="center"></td><td>CPU</td>
66. <td></td>
67. <td></td>
68. <td bgcolor="#87cefa"><a href="https://www.hackintosh-
forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-
Beispielen/#CPUFriend">hier</a></td>
69. </tr>
70. <tr><!-- CpuTscSync -->
71. <td bgcolor="#87cefa"><a href="https://github.com/lvs1974/CpuTscSync">
CpuTscSync</a></td>
72. <td>Kombiniert VoodooTSCSync und Deaktivieren von xcpm_urgency, wenn TSC nicht
synchronisiert ist</td>
73. <td align="center"></td><td>Kernel</td>
74. <td></td>
75. <td></td>
76. <td bgcolor="#87cefa"><a href="https://www.hackintosh-
forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-
Beispielen/#CpuTscSync">hier</a></td>
77. </tr>
78. <tr bgcolor="#dcdcdc"><!-- DebugEnhancer -->
79. <td bgcolor="#87cefa"><a href="https://github.com/acidanthera/DebugEnhancer">
DebugEnhancer</a></td>
80. <td>Aktiviert die Debug-Ausgabe im macOS-Kernel</td>
81. <td align="center"></td><td>Kernel</td>
82. <td></td>
83. <td></td>
84. <td bgcolor="#87cefa"><a href="https://www.hackintosh-forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-Beispielen/#DebugEnhancer">hier</a></td>
85. </tr>
86. <tr><!-- DiskArbitrationFixup -->
87. <td bgcolor="#87cefa"><a href="https://github.com/Goldfish64/DiskArbitrationFixup">DiskArbitrationFixup</a></td>
88. <td>Deaktiviert die Nachricht von nicht initialisierten Datenträgern beim Einlegen</td>
89. <td align="center"></td><td>CPU</td>
90. <td></td> <td></td><td bgcolor="#87cefa"><a href="https://www.hackintosh-forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-Beispielen/#DiskArbitrationFixup">hier</a></td>
91. </tr>
92. <tr bgcolor="#dcdcdc"><!-- HibernationFixup -->
93. <td bgcolor="#87cefa"><a href="https://github.com/acidanthera/HibernationFixup">HibernationFixup</a></td>
94. <td>Aktiviert 3 & 25-Modus Ruhezustand auf bestimmte Hardware</td>
95. <td align="center"></td><td> Grafik</td>
96. <td></td>
97. <td></td>
98. <td bgcolor="#87cefa"><a href="https://www.hackintosh-forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-Beispielen/#HibernationFixup">hier</a></td>
99. </tr>
100. <tr><!-- Innie -->
101. <td bgcolor="#87cefa"><a href="https://github.com/cdf/Innie"> Innie</a></td>
102. <td>PCIe-Laufwerke werden als intern angezeigt</td>
103. <td align="center"></td><td>Kernel</td>

```

```

104. <td></td> <td></td><td bgcolor="#87cefa"><a href="https://www.hackintosh-
forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-
Beispielen/#Innie">hier</a></td>
105. </tr>
106. <tr bgcolor="#dcdcdc"><!-- MacProMemoryNotificationDisabler -->
107. <td align="center" bgcolor="#87cefa"><a href="https://github.com/IOIIIIO/MacProMemoryNotificationDisabler">
MacProMemoryNotificationDisabler</a></td>
108. <td>Deaktivieren des Popups "Mehr als maximale Speichermenge" auf MacPro7,1</td>
109. <td align="center"></td><td> Grafik</td>
110. <td></td> <td></td><td bgcolor="#87cefa"><a href="https://www.hackintosh-
forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-
Beispielen/#MacProMemoryNotificationDisabler">hier</a></td>
111. </tr>
112. <tr><!-- NightShiftUnlocker -->
113. <td bgcolor="#87cefa"><a href="https://github.com/Austere-J/NightShiftUnlocker">
NightShiftUnlocker</a></td>
114. <td>Ermöglicht Night Shift für alle Modelle</td>
115. <td align="center"></td><td> Grafik</td>
116. <td></td> <td></td><td bgcolor="#87cefa"><a href="https://www.hackintosh-
forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-
Beispielen/#NightShiftUnlocker">hier</a></td>
117. </tr>
118. <tr bgcolor="#dcdcdc"><!-- NoTouchID -->
119. <td bgcolor="#87cefa"><a href="https://github.com/al3xtjames/NoTouchID">
NoTouchID</a></td>
120. <td>Deaktiviert Touch-ID-Prüfungen, die zu Abstürzen führen</td>
121. <td align="center"></td><td> Grafik</td>
122. <td></td> <td></td><td bgcolor="#87cefa"><a href="https://www.hackintosh-
forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-
Beispielen/#NoTouchID">hier</a></td>
123. </tr>
124. <tr><!-- NVMeFix -->
125. <td bgcolor="#87cefa"><a href="https://github.com/acidanthera/NVMeFix">
NVMeFix</a></td>

```

```

126. <td>Verbesserte Energieverwaltung für generische NVMe-SSDs</td>
127. <td align="center"></td><td> SSD</td>
128. <td></td> <td></td><td bgcolor="#87cefa"><a href="https://www.hackintosh-
    forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-
    Beispielen/#NVMeFix">hier</a></td>
129. </tr>
130. <tr bgcolor="#dcdcdc"><!-- OpcodeEmulator -->
131. <td bgcolor="#87cefa"><a href="https://www.insanelymac.com/forum/topic/329704-
    opcode-emulator-opemu-plugin-in-project/"> OpcodeEmulator</a></td>
132. <td>Intel Haswell Pentium / Celeron-Serie oder ältere CPU zur Prozessorerweiterung
    Emulation</td>
133. <td align="center"></td><td> CPU</td>
134. <td></td> <td></td><td bgcolor="#87cefa"><a href="https://www.hackintosh-
    forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-
    Beispielen/#OpcodeEmulator">hier</a></td>
135. </tr>
136. <tr><!-- RTCMemoryFixup -->
137. <td bgcolor="#87cefa"><a href="https://github.com/acidanthera/RTCMemoryFixup">
    RTCMemoryFixup</a></td>
138. <td>Offsets in der CMOS (RTC) -Speicheremulation</td>
139. <td align="center"></td><td> Grafik</td>
140. <td></td> <td></td><td bgcolor="#87cefa"><a href="https://www.hackintosh-
    forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-
    Beispielen/#RTCMemoryFixup">hier</a></td>
141. </tr>
142. <tr bgcolor="#dcdcdc"><!-- SystemProfilerMemoryFixup -->
143. <td align="center" bgcolor="#87cefa"><a
    href="https://github.com/Goldfish64/SystemProfilerMemoryFixup">
    SystemProfilerMemoryFixup</a></td>
144. <td>Zeigt den RAM in der Speicher Karte bei MacBook-Modellen mit gelötetem RAM
    an</td>
145. <td align="center"></td><td> Kernel</td>
146. <td></td> <td></td><td bgcolor="#87cefa"><a href="https://www.hackintosh-
    forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-
    Beispielen/#SystemProfilerMemoryFixup">hier</a></td>
147. </tr>
148. <tr><!-- ThunderboltReset -->

```

```

149. <td bgcolor="#87cefa"><a href="https://github.com/osy86/ThunderboltReset">
ThunderboltReset</a></td>
150. <td>Deaktiviert das ICM bei Alpine Ridge</td>
151. <td align="center"></td><td> Grafik</td>
152. <td></td> <td></td><td bgcolor="#87cefa"><a href="https://www.hackintosh-
forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-
Beispielen/#ThunderboltReset">hier</a></td>
153. </tr>
154. <tr bgcolor="#dcdcdc"><!-- VirtualSMC -->
155. <td bgcolor="#87cefa"><a href="https://github.com/acidanthera/VirtualSMC">
VirtualSMC</a></td>
156. <td>SMC emulator layer</td>
157. <td align="center"></td><td> Kernel</td>
158. <td></td> <td></td><td bgcolor="#87cefa"><a href="https://www.hackintosh-
forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-
Beispielen/#VirtualSMC">hier</a></td>
159. </tr>
160. <tr><!-- WhateverGreen -->
161. <td bgcolor="#87cefa"><a href="https://github.com/acidanthera/WhateverGreen">
WhateverGreen</a></td>
162. <td>Patches für ATI/AMD/Intel/Nvidia GPUs</td>
163. <td align="center"></td><td> Grafik</td>
164. <td></td> <td></td><td bgcolor="#87cefa"><a href="https://www.hackintosh-
forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-
Beispielen/#WhateverGreen">hier</a></td>
165. </tr>
166. </tbody>
167. </table>
168. <p>
169. </p>
170. <table style="background-color:#F2F2F2" cellspacing="3"
cellpadding="5"><caption><b>Plugins die in neue Plugins eingeflossen
sind</b></caption>
171. <thead>

```



```

172. <tr bgcolor="#87cefa"><th>Name <span class="badge green">
    *</span></th><th>Kurze Beschreibung</th><th>Letzte Version <span class="badge
    red"> *</span></th><th>Kategorie</th><th>Bootflags</th></tr>
173. </thead>
174. <tbody>
175. <tr bgcolor="#dcdcdc"><!-- AzulPatcher4600 -->
176. <td bgcolor="#87cefa"><a href="https://github.com/coderobe/AzulPatcher4600">
    AzulPatcher4600</a></td>
177. <td>Zusätzliche Intel HD 4600 Patches für einige Laptops am 10.12.4+ <span
    class="badge orange"> *WhateverGreen</span></td>
178. <td align="center" style="">v 1.2.0</td><td>Grafik</td>
179. <td
    bgcolor="#87cefa"><a href="https://www.hackintosh-
    forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-
    Beispielen/#AzulPatcher4600">hier</a></td>
180. </tr>
181. <tr><!-- AppleBacklightFixup -->
182. <td
    bgcolor="#87cefa"><a href="https://bitbucket.org/RehabMan/applebacklightfixup">
    AppleBacklightFixup</a></td>
183. <td>Patch fü Clover-Beschränkung AppleBacklight.kext in OS-Installer und
    Wiederherstellung, Original-Repository von hieplpvip <span class="badge orange">
    *WhateverGreen</span></td>
184. <td align="center" style="">v 1.0.2</td><td>Grafik</td>
185. <td
    bgcolor="#87cefa"><a href="https://www.hackintosh-
    forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-
    Beispielen/#AppleBacklightFixup">hier</a></td>
186. </tr>
187. <tr bgcolor="#dcdcdc"><!-- EnableLidWake -->
188. <td bgcolor="#87cefa"><a href="https://github.com/syscl/EnableLidWake">
    EnableLidWake</a></td>
189. <td>Aufwachen für Intel Grafikkarten <span class="badge orange">
    *WhateverGreen</span></td>
190. <td align="center" style="">v 2.1</td><td>Grafik</td>
191. <td
    bgcolor="#87cefa"><a href="https://www.hackintosh-
    forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-
    Beispielen/#EnableLidWake">hier</a></td>
192. </tr>
193. <tr><!-- BrcmWLFixup -->
194. <td bgcolor="#87cefa"><a href="https://github.com/PMheart/BrcmWLFixup">
    BrcmWLFixup</a></td>

```



```

werden <span class="badge orange"> *WhateverGreen</span></td>
220. <td align="center" style="">v 1.0.0</td><td> Grafik</td>
221. <td
      bgcolor="#87cefa"><a
      href="https://www.hackintosh-
forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-
Beispielen/#NoVPAJpeg">hier</a></td>
222. </tr>
223. <tr bgcolor="#dcdcdc"><!-- NvidiaGraphicsFixup -->
224. <td bgcolor="#87cefa"><a href="https://github.com/lvs1974/NvidiaGraphicsFixup">
NvidiaGraphicsFixup</a></td>
225. <td>Patches für NVIDIA-GPUs <span class="badge orange">
*WhateverGreen</span></td>
226. <td align="center" style="">v 1.2.7</td><td> Grafik</td>
227. <td
      bgcolor="#87cefa"><a
      href="https://www.hackintosh-
forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-
Beispielen/#NvidiaGraphicsFixup">hier</a></td>
228. </tr>
229. <tr><!-- Shiki -->
230. <td bgcolor="#87cefa"><a href="https://github.com/vit9696/Shiki"> Shiki</a></td>
231. <td>Dynamischer Userpace-Patcher für MacOS <span class="badge orange">
*WhateverGreen</span></td>
232. <td align="center" style="">v 2.2.7</td><td> Grafik</td>
233. <td
      bgcolor="#87cefa"><a
      href="https://www.hackintosh-
forum.de/forum/thread/32411-Bootflags-zu-Lilu-Plugins-mit-
Beispielen/#Shiki">hier</a></td>
234. </tr>
235. </tbody>
236. </table>
237. <p>
238. </p>
239. <table
      style="background-color:#F2F2F2"
      cellspacing="3"
      cellpadding="5"><caption><b>Sensoren und Plugins von VirtualSMC</b></caption>
240. <thead>
241. <tr
      bgcolor="#87cefa"><th>Name <span class="badge green">
* </span></th><th>Kurze Beschreibung</th><th>Letzte Version <span class="badge
red"> * </span></th><th>Kategorie</th></tr>
242. </thead>
243. <tbody>
244. <tr bgcolor="#dcdcdc"><!-- SMCBatteryManager -->
245. <td bgcolor="#87cefa"><a href="">SMCBatteryManager</a></td>
246. <td></td>
247. <td align="center"></td><td></td>

```

```

248. </tr>
249. <tr><!-- SMCDellSensors -->
250. <td bgcolor="#87cefa"><a href="">SMCDellSensors</a></td>
251. <td></td>
252. <td align="center"></td><td></td>
253. </tr>
254. <tr bgcolor="#dcdcdc"><!-- SMCLightSensor -->
255. <td bgcolor="#87cefa"><a href="">SMCLightSensor</a></td>
256. <td></td>
257. <td align="center"></td><td></td>
258. </tr>
259. <tr><!-- SMCPProcessor -->
260. <td bgcolor="#87cefa"><a href="">SMCPProcessor</a></td>
261. <td></td>
262. <td align="center"></td><td></td>
263. </tr>
264. <tr bgcolor="#dcdcdc"><!-- SMCSuperIO -->
265. <td bgcolor="#87cefa"><a href="">SMCSuperIO.kext</a></td>
266. <td></td>
267. <td align="center"></td><td></td>
268. </tr>
269. <tr><!-- AsusSMC -->
270. <td
                                bgcolor="#87cefa"><a
                                href="https://github.com/hieplvpip/AsusSMC">AsusSMC</a></td>
271. <td>Improved ASUS ATK support (e.g. keyboard backlight)</td>
272. <td
                                align="center"></td><td></td>
273. </tr>
274. <tr bgcolor="#dcdcdc"><!-- SMCAMDProcessor -->
275. <td
                                bgcolor="#87cefa"><a
                                href="https://github.com/trulyspinach/SMCAMDProcessor">SMCAMDProcessor</a></td>
276. <td>AMD Processor power management plugin</td>
277. <td
                                align="center"></td><td></td>
278. </tr>
279. </tbody>
280. </table>
281. <p>
282. </p>
283. <table
                                style="background-color:#F2F2F2"
                                cellspacing="3"
                                cellpadding="5"><caption><b><i style="color:red">Achtung</i> LiluFriend ist kein

```

```

Plugin von Lilu!</b></caption>
284. <thead>
285. <tr bgcolor="#87cefa"><th>Name <span class="badge green">
    *</span></th><th>Kurze Beschreibung</th><th>Lilu 1.2.0</th><th>Letzte Version
    <span class="badge red"> *</span></th><th>Kategorie</th></tr>
286. </thead>
287. <tbody>
288. <tr bgcolor="#dcdcdc"><!-- LiluFriend -->
289. <td bgcolor="#87cefa"><a href="https://github.com/PMheart/LiluFriend/releases">
    LiluFriend</a></td>
290. <td>Hilfsklasse für Lilu</td><td bgcolor="ff4500"></td>
291. <td align="center" style="">v 1.1.0</td><td> WiFi</td>
292. </tr>
293. </tbody>
294. </table>
295.
296.
297.
298. <p><span class="badge green"> *</span> Lilu & Plugins sind verlinkt zu den einzelnen
    Repo's der Entwickler</p>
299. <p><span class="badge orange"> *</span> Neues Plugin übernommen</p>
300. <p><span class="badge red"> * v X.X.X</span> Kexte die in einer neuen Version seit
    der letzten Tabelle vorliegen</p>
301. <p><span class="badge blue"> *</span> <a href="https://github.com/black-
    dragon74/ATH9KFixup">ATH9KFixup Fork</a> ist Lilu 1.2.0 tauglich</p>

```

Alles anzeigen

Beitrag von „Noir0SX“ vom 12. Oktober 2017, 12:42

Features und Bootflags

Code

1. `<h3>Lilu</h3>`
2. `<div class="warning notice">`
3. Achtung:`

`
4. Die meisten Plugins funktionieren nicht mehr im sicheren Modus (-x). Standardmäßig funktioniert Lilu selbst nicht im Einzelbenutzermodus (-s), es sei denn, -liluforce ist vorhanden.`</div>`

- -liludbg : aktiviert Debug Ausgaben
- -liludbgall : aktiviert Debug Ausgaben aller geladenen Plugins
- -liluoff : deaktiviert das Laden
- -liluuseroff : deaktiviert Lilu Plugin Patch (für z. B. dyld_shared_cache-Manipulationen) **v1.2.6**
- -liluslow : aktiviert Legacy User Patcher
- -lilulowmem : deaktiviert Kernel Unpack (nur zum Debug)
- -liluforce : aktiviert sicheren Modus und Recovery ab **v1.1.5**
- -lilubeta : aktiviert das Laden in noch nicht unterstützten macOS Versionen
- -lilubetaall : aktiviert das Laden für alle Plugins in noch nicht unterstützten macOS Versionen **v1.1.7**
- liludelay=1000 : verzögert den Debug Print um 1 Sekunde **v1.2.4**
- -lilucpu=N : übernimmt die CPU Erkennung
- -liludump=N : Lilu DEBUG-Version nach N Sekunden das Protokoll in /var/log/Lilu_VERSION_KERN_MAJOR.KERN_MINOR.txt ablegt

[Zum Seitenanfang](#)

Plugins (alle hier aufgeführten KEXT brauchen zum Ausführen als Voraussetzung Lilu.kext)

Code

1. `<h3> AirportBrcmFixup * Neu ab Lilu 1.1.6 ersetzt BrcmWlFixup`
2. `</h3>`

Features

- -brcmfxoff : deaktiviert das Laden

- `-brcmfxdbg` : aktiviert Debug Ausgaben
- `-brcmfxbeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen
- `brcmfx-country=XX` : Ändert die Länderkennung zu XX (US, DE, CN, ...)
- `-wl_msg_level=XXXXX` : v1.0.7
- `-wl_msg_level2=XXXXX` : v1.0.7
- `-brcmfxwow` : aktiviert WOWL (WoWLAN) - deaktiviert bei Default v1.0.10

[Zum Seitenanfang](#)

Code

1. `<h3> AppleALC </h3>`

Features

- `-alcoff` : deaktiviert das Laden
- `-alcdbg` : aktiviert Debug Ausgaben
- `-alcbeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen
- `alcid=X` : überschreibt die layout-id und wird als apple-layout-id wiedergegeben ab v1.2.8

[Zum Seitenanfang](#)

Code

1. `<h3> AMFIExemption </h3>`

Features

- `-amfiexemptoff` : deaktiviert das Laden
- `-amfiexemptdbg` : aktiviert Debug Ausgaben
- `-amfiexemptbeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen

[Zum Seitenanfang](#)

Code

1. [ATH9KFixup](#) **ATH9KFixup**

Features

- `-ath9koff` : deaktiviert das Laden
- `-ath9kdbg` : aktiviert Debug Ausgaben
- `-ath9kbeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen

[Zum Seitenanfang](#)

Code

1. [BT4LEContiunityFixup](#) **BT4LEContiunityFixup**

Features

- `-bt4lexoff` : deaktiviert das Laden
- `-bt4lexdbg` : aktiviert Debug Ausgaben
- `-bt4lexbeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen

[Zum Seitenanfang](#)

Code

1. [CPUFriend](#) **CPUFriend**

- `-cpufoff` : deaktiviert das Laden
- `-cpufdbg` : aktiviert Debug Ausgaben
- `-cpufbeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen

[Zum Seitenanfang](#)

Code

1. [CpuTscSync](#) **CpuTscSync**

Features

- `-cpusoff` : deaktiviert das Laden
- `-cpusdbg` : aktiviert Debug Ausgaben

- `-cputsbeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen

[Zum Seitenanfang](#)

Code

1. `<h3> DebugEnhancer </h3>`

- `-dbgenhxooff` : deaktiviert das Laden
- `-dbgenhdbg` : aktiviert Debug Ausgaben
- `-dbgenhbeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen

[Zum Seitenanfang](#)

Code

1. `<h3> DiskArbitrationFixup </h3>`

Features

- `-dafxooff` : deaktiviert das Laden
- `-dafxdbg` : aktiviert Debug Ausgaben
- `-dafxbeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen

[Zum Seitenanfang](#)

Code

1. `<h3> HibernationFixup </h3>`

Features

- `-hbfxooff` : deaktiviert das Laden
- `-hbfxdbg` : aktiviert Debug Ausgaben
- `-hbfxbeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen
- `-hbfx-dump-nvram` : speichert NVRAM im File `nvram.plist` vor Hibernation und nach Kernel Panic
- `-hbfx-patch-pci` : aktiviert IOPCIFamily um hängendes & schwarzes Bild nach restart zu vermeiden
- `-hbfx-patch-pci=` : Durch Komma getrennte Liste der ignorierten Geräte ab v1.1.6

[Zum Seitenanfang](#)

Code

1. `<h3> Innie </h3>`

Features

- `-innieoff` : deaktiviert das Laden
- `-inniedbg` : aktiviert Debug Ausgaben
- `-inniebeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen

[Zum Seitenanfang](#)

Code

1. `<h3> MacProMemoryNotificationDisabler </h3>`

Features

- `-mpmndoff` : deaktiviert das Laden
- `-mpmnddbg` : aktiviert Debug Ausgaben
- `-mpmndbeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen

[Zum Seitenanfang](#)

Code

1. `<h3> NightShiftUnlocker </h3>`

- `-nsuoff` : deaktiviert das Laden
- `-nsudbg` : aktiviert Debug Ausgaben
- `-nsubeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen

[Zum Seitenanfang](#)

Code

1. [NoTouchID](#)

Features

- `-nobiiooff` : deaktiviert das Laden
- `-nobiobeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen

[Zum Seitenanfang](#)

Code

1. [NVMeFix](#)

Features

- `-nvmefoff` : deaktiviert das Laden
- `-nvmefdbg` : aktiviert Debug Ausgaben

[Zum Seitenanfang](#)

Code

1. [RTCMemoryFixup](#)

Features

- `-rtcfxdbg` : aktiviert Debug Ausgaben
- `-rtcfx_exclude=offset1,offset2,start_offset-end_offset` : Liste der Offsets oder Bereiche von Offsets, in denen das Schreiben nicht zulässig ist

[Zum Seitenanfang](#)

Code

1. [SystemProfilerMemoryFixup](#)

Features

- `-spmemfxoff` : deaktiviert das Laden
- `-spmemfxdbg` : aktiviert Debug Ausgaben
- `-spmemfxbeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen

[Zum Seitenanfang](#)

Code

1. `<h3> Thunderbolt Reset </h3>`

Features

- `-tbresetoff` : deaktiviert das Laden
- `-tbresetdbg` : aktiviert Debug Ausgaben
- `-tbresetbeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen

[Zum Seitenanfang](#)

Code

1. `<h3> VirtualSMC </h3>`

Features

- `-vsmccoeff` deaktiviert Lilu
- `-vsmcbeta` aktiviert Lilu Plugins für nicht unterstütztes macOS
- `-vsmcrrpt` SMC-Schlüssel im Systemprotokoll
- `-vsmccomp` sucht unterstützte SMC Hardware
- `-vsmcgen=X` to force exposing X-gen SMC device (1 and 2 are supported).
- `-vsmchbkrp=X` to set HBKP dumping mode (0 - off, 1 - normal, 2 - without encryption).
- `-vsmcslvl=X` to set value serialisation level (0 - off, 1 - normal, 2 - with sensitive data (default)).
- `-smcdebug=0xff` to enable AppleSMC debug information printing.
- `-watchdog=0` to disable WatchDog timer (if you get accidental reboots).

[Zum Seitenanfang](#)

Code

1. `<h3> WhateverGreen * Neu ab Lilu 1.1.6(alternativ dazu`
2. ` AMD-RADEON-RX-Grafikkarten-ohne-LILU-WhatEverGreen-nutzen)`
3. `</h3>`

Features

- `-wegdbg` : aktiviert Debug Ausgaben
- `-wegoff`: deaktiviert das Laden
- `-wegbeta` : aktiviert das Laden in noch nicht unterstützten macOS
- `-wegnoegpu` : deaktiviert externe GPU (oder `disable-external-gpu` für IGPU)
- `-wegtree` : erzwingt das Umbenennen von Geräten **v1.3.5**
- `-igfxvesa` : Intel-Grafik ohne Hardware-Beschleunigung (VESA-Modus) zu booten
- `-radvesa` : deaktiviert ATI-AMD [Video Beschleunigung](#) Komplet
- `-rad24`: Erzwingt den 24Bit Anzeigemodus
- `-raddvi` : deaktiviert DVI standardmäßig (Multimonitor Support) v1.0.1
- `radpg=15` : deaktiviert Several Power-Gating Modus (Details in der FAQ).
- `-radgl` : deaktiviert den Metal Support (v.1.1.7)
- `-radcfg` : Korrigiert den Namen des Beschleunigers
- `agdpmo` : Komma getrennte List von Patches: `cfgmap`, `Pikera`, `Vit9696` (by default)
 - `agdpmo=cfgmap` : Erzwingt none in der ConfigMap für Board-ID
 - `agdpmo=vit9696` : deaktiviert die Prüfung auf Board-ID, standardmäßig aktiviert
 - `agdpmo=pikera` : ersetzt Board-ID durch Board-IX
- `-applbkl` : AppleBacklight Kext Patch**v1.3.5**
- `-applbkl-name` : **v1.3.5**
- `-applbkl-data` : **v1.3.5**
- `ngfxgl=1` : deaktiviert den Metal Support
- `ngfxcompat=1` : deaktiviert Prüfung NVDAStartupWeb
- `ngfxsubmit=0` : deaktiviert Interface-Stottern auf 10.13 NVDAStartupWeb
- `igfxrst=1` : Zeichnen des Apple-Logos in der zweiten Boot-Phase anstelle des Kopierens des Framebuffers
- `igfxgl=0` : deaktiviert den Metal Support
- `igfxsnb=0` : deaktiviert IntelAccelerator Fix für Sandy Bridge CPUs
- `igfxframe=frame` : um eine dedizierte Framebuffer-ID in IGPU zu injizieren (nur für Testzwecke)
- `-igfxnohdmi` : deaktiviert DP zu HDMI Conversion Patches für Digitalen Sound
- `igfxcflbkl` : Komma getrennte List von Patches: `none`, `Opcode`, `wrap`, `Freq` (v.1.2.4)
 - `igfxcflbkl = none` : deaktiviert CFL-Backlight-Patch
 - `igfxcflbkl = Opcode` : aktiviert Fraxul's CFL Backlight Patch

- `igfxcflbklt = wrap` : aktiviert Headkaze's CFL Backlight Patch (und `enable-cfl-backlight-fix` property)
- `igfxcflbklt = Freq` : aktiviert CFL-Backlight-Patch von RehabMan
- `-cdfon` : aktiviert HDMI 2.0 Patches `-enable-hdmi20` (v.1.2.1)

Obsolet

- `-radlogo` : Patch gegen ein verzerrtes Boot Logo *v1.1.4 nicht mehr notwendig und entfernt
- `-radnoaudio` : deaktiviert die automatische Verwendung Audio DP/HDMI (alternativ `-no-audio-autofix` für GPU Controller)
- `-rad4200` : aktiviert den 4200-Treiber für 0x67DF, 0x67EF und 0x67FF

[Zum Seitenanfang](#)

Code

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

<h2>Plugins die in neue Plugins eingeflossen sind</h2></div>`
-

Code

1. `<h3> AzulPatcher4600 </h3>`

Features

- `-cdrpatchoff` : deaktiviert das Laden
- `-cdrpatchdbg` : aktiviert Debug Ausgaben
- `-cdrpatchbeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen

[Zum Seitenanfang](#)

Code

1. `<h3> AppleBacklightFixup </h3>`
 - `-applbkloff` : deaktiviert das Laden

- `-applbklbg` : aktiviert Debug Ausgaben
- `-applbklbeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen

[Zum Seitenanfang](#)

Code

1. `<h3> EnableLidWake </h3>`

- `-elwoff` : deaktiviert das Laden
- `-elwdbg` : aktiviert Debug Ausgaben
- `-elwbeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen

[Zum Seitenanfang](#)

Code

1. `<h3>BrcmWLFixup * bis Lilu v1.1.5 Neu AirportBrcmFixup</h3>`

- `-brcmoff` : deaktiviert das Laden
- `-brcmdbg` : aktiviert Debug Ausgaben
- `-brcmbeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen

[Zum Seitenanfang](#)

Code

1. `<h3> CoreDisplayFixup </h3>`

- `-cdfdbg` : aktiviert Debug Ausgaben
- `-cdfoff` : deaktiviert das Laden
- `-cdfbeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen
- `-cdfinteloff` : deaktiviert IntelPatcher
- `-cdfnvoff` : deaktiviert NVPatcher

Code

1. `<h3> IntelGraphicsDVMTFixup </h3>`

Features

- `-igdvmtoff` : deaktiviert das Laden
- `-igdmvtdbg` : aktiviert Debug Ausgaben
- `-igdvmtbeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen

Code

1. `<h3> IntelGraphicsFixup </h3>`

Features

- `-igfxoff` : deaktiviert das Laden
- `-igfxdbg` : aktiviert Debug Ausgaben
- `-igfxbeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen
- `-igfxvesa` : Intel-Grafik ohne Hardware-Beschleunigung (VESA-Modus) zu booten
- `igfxrst=1` : Zeichnen des Apple-Logos in der zweiten Boot-Phase anstelle des Kopierens des Framebuffers
- `igfxfw=1` : aktiviert das Laden von GuC-Mikrocode in 10.13 oder neuer zu aktivieren
- `igfxgl=0` : deaktiviert den Metal Support (v1.2.7)
- `igfxsnb=0` : deaktiviert IntelAccelerator Fix für Sandy Bridge CPUs (v1.2.7)
- `igfxframe=frame` : um eine dedizierte Framebuffer-ID in IGPU zu injizieren (nur für Testzwecke)

Code

1. `<h3> NvidiaGraphicsFixup </h3>`

Features

- `-ngfxoff` : deaktiviert das Laden
- `-ngfxdbg` : aktiviert Debug Ausgaben
- `-ngfxbeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen
- `-ngfxnoaudio` : deaktiviert alle Audio Fixes
- `-ngfxnoaudiocon` : Fügt `@0,connector-type - @5,connector-type` hinzu
- `-ngfxlibvalfix` : deaktiviert `NVWebDriverLibValFix` fix
- `-ngfxnovarenderer` : deaktiviert `IOVAREnderer-Injektion`
- `ngfxpatch` : Komma getrennte List von Patches: `cfgmap, Pikera,Vit9696` (by default)
 - `-ngfxpatch=cfgmap` : Erzwingt `none` in der `ConfigMap` für Board-ID
 - `-ngfxpatch=vit9696` : deaktiviert die Prüfung auf Board-ID, standardmäßig aktiviert
 - `-ngfxpatch=pikera` : ersetzt Board-ID durch Board-IX
- `ngfxgl=1` : deaktiviert den Metal Support (v.1.2.4)
- `-ngfxcompat=1` : deaktiviert Prüfung `NVDAStartupWeb` (v.1.2.4)
- `-ngfxsubmit=0` : deaktiviert Interface-Stottern auf 10.13 `NVDAStartupWeb` (v.1.2.5)

[Zum Seitenanfang](#)

Code

1. `<h3> NoVPAJpeg </h3>`

- `-novpaoff` : deaktiviert das Laden
- `-novpabeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen

[Zum Seitenanfang](#)

Code

1. [**Shiki**](#)

- `-shikioff` : deaktiviert das Laden
- `-shikibg` : aktiviert Debug Ausgaben
- `-shikibeta` : aktiviert das Laden in noch nicht unterstützten macOS Versionen
- `shikigva=x` :
 - `shikigva=1` : (wie bis v2.0.4)
 - `shikigva=2` : deaktiviert hweBGRA (nützlich für einige AMD) (ab v2.0.4)
 - `shikigva=4` : SKL/KBL & NVIDIA
 - `shikigva=8` : muss dies verwendet werden, wenn eines der folgenden Bits verwendet wird:// - ForceOnlineRenderer // - AllowNonBGRA // - ForceCompatibleRenderer // - ReplaceBoardID // - FixSandyBridgeClassName
 - `shikigva=16` : für iTunes Crashes in 10.13 (ab v2.1.1)
 - `shikigva=32` : Überschreiben des Mac-Modells
 - `shikigva=64` : Versuch, fps.1_0 (FairPlay 1.0) in Safari zu unterstützen, alternativ `-shikifps` (ab v2.2.0)
 - `shikigva=128` : Sandy Bridge (ab v2.2.4)
- `shiki-id = "board-id"` : ist standardmäßig iMac14,2

[Zum Seitenanfang](#)

Beitrag von „Noir0SX“ vom 4. November 2017, 14:55

Anwendung

`-alcoff` und `-shikioff` sind in der Auswahl Boot im CloverConfigurator enthalten und können dort an- oder abgewählt werden.

Die anderen Bootflags können als Custom Flags unter Boot gesetzt werden. Einige Beispiele sind in den folgenden Zeilen zu lesen.

Beispiel 1:

Laden von derzeit nicht unterstützten Versionen

z.B. macOS 10.1x Beta mit Lilu & AppleALC `-lilubeta -alcbeta`

ab Lilu v1.1.7

z.B. macOS 10.1x Beta mit Lilu & AppleALC `-lilubetaall`

Debug

Das Debuggen und Erzeugen von Logeinträgen erfolgt wie in den nächsten Zeilen zulesen ist, durch setzen des entsprechenden BootFlags von Lilu & des entsprechenden Plugins. Seit der aktuellen Version kann auch einfach `-liludbgall` dazu verwendet werden.

Bei KernelPanik, wie zuletzt bei 10.12.13 Beta2, kann auch der Flag `-liluforce` zum kurzzeitigen Erfolg führen

Beispiel 1:

Debug Ausgabe von AppleALC (Audio)

Nach dem setzen der Bootflags `-alcdbg -liludbg` und dem Booten folgende Befehle im Terminal absetzen.

10.12 und Neuer :

Code

1. `log show --predicate 'process == "kernel" AND (eventMessage CONTAINS "AppleALC" OR eventMessage CONTAINS "Lilu")' --style syslog --source`

10.11 und älter :

Code

1. `cat /var/log/system.log | egrep '(AppleALC|Lilu)'`

Beispiel 2:

DEBUG Ausgabe von WhateverGreen (Grafik) Framebuffer

Wenn Sie einen Dump Ihrer Framebuffer in hex benötigen, verwenden Sie den Boot-Flag `-igfxdump` Die Ausgabe wird in `/AppleIntelFrambuffer_GEN_KERNX_Y` sein.

Beispiel 3:

DEBUG Ausgabe von WhateverGreen (Grafik) Um Debug-Ausgaben von WhateverGreen zu erhalten, benutzen Sie bitte das Boot-Flag `-wegdbg`. Sie müssen WhateverGreen als Debug für beide kompilieren um mit diesen Flaggen zu arbeiten.

10.14 und Neuer :

Code

1. `log show --predicate 'process == "kernel" AND (eventMessage CONTAINS "WhateverGreen" OR eventMessage CONTAINS "Lilu")' --style syslog --source --last boot`

Sollte das Problem GPU spezifisch sein, kann man versuchen das Protokoll über SSH zu erhalten.

Beispiel 4:

Um herauszufinden welches Gerät mit HibernationFixup verboten werden soll

Debug Ausgabe von HibernationFixup

Nach dem setzen der Bootflags `-hbfxdbg -liludbg -hbfx-patch-pci` und dem Booten in den Hibernation Modus gehen und nach dem Aufwachen folgende Befehle im Terminal absetzen.

10.12 und Neuer :

Code

1. `log show --predicate 'process == "kernel" AND eventMessage CONTAINS "HBFX"' --style syslog --source`

In der Ausgabe des Befehls gibt es Zeilen wie:

HBFX @ extendedConfigWrite16 wird nicht für das Gerät IGPU aufgerufen

HBFX @ extendedConfigWrite16 wird nicht für das Gerät IMEI aufgerufen

HBFX @ extendedConfigWrite16 wird nicht für das Gerät XHC aufgerufen

Beitrag von „Noir0SX“ vom 4. November 2017, 14:55

Framebuffer Patch

In WhatevenGreen hat thx headkatze und vit9696 einen Framebuffer Patch eingebunden. Dieser besteht aus dem Framebuffer und den Connectoren

Gründe für einen Framebuffer-Patch könnten sein

- Einstellung [DVMT](#), wo die [Bios Einstellungen](#) höher 32MB es nicht zulassen
- Ein höhere VRAM
- Deaktivieren von EGPU
- Deaktivieren von Connectors und Aktivieren des Ruhezustands
- Entfernen des CNConnectorAlwaysConnected-Flags für eDP-Laptop-Bildschirme
- Ändern der Konnektortypen zu Ihren Systemanschlüssen

Tools

Zum bestimmen des Gerätepfades wird [gfxutil](#) benötigt

Aufruf im Terminal `./gfxutil -f IGPU`

Code

1. `DevicePath = PciRoot(0x0)/Pci(0x2,0x0)`

Oder

Aufruf im Terminal `bdmesg | grep -y "intel hd"`

Code

1. `0:115 0:000 - GFX: Model=Intel HD Graphics 515 (Intel)51:502 0:000 Intel HD Graphics 515 [8086:191E] :: PciRoot(0x0)\Pci(0x2,0x0)`

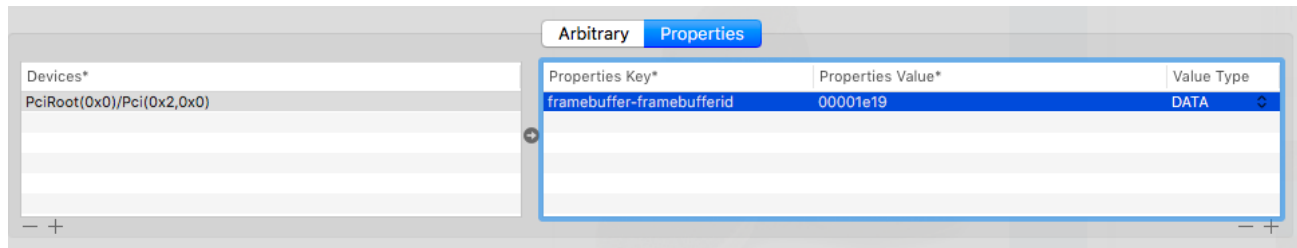
Den Gerätepfad `PciRoot(0x0)\Pci(0x2,0x0)` merken wir uns für später

Zum Auslesen und dem Erstellen, wie in den folgenden Zeilen beschrieben kann der [Intel-FB-Patcher](#) eine große Hilfe sein

ig-platform-id

Die ID sollte fürs System bekannt sein oder eine Systemverwandte genommen werden.

Diese zwei Werte trägt man nun in CloverConfigurator ein.



Sollte ein anderer Weg gewählt werden muss die ID 191e0000 erst in umgekehrt 00001e19 und dann mit Base64 `AAAEgQ==` umgewandelt werden

Code

1. DevicesProperties
2. PciRoot(0x0)/Pci(0x2,0x0)
3. framebuffer-framebufferid `AAAEgQ==`

In vielen Fällen langt dies schon und WhateverGreen macht den Rest.

Framebuffer-Patchtypen

Es gibt zwei verschieden Arten von Patches

Hart codiert (empfohlen)

Code

1. framebuffer-patch-enable (muss aktiviert werden)
2. framebuffer-framebufferid (extra aktuelle Platform-id)(extra)
3. framebuffer-stolenmem
4. framebuffer-fbmem
5. framebuffer-unifiedmem
6. framebuffer-mobile
7. framebuffer-pipecount
8. framebuffer-portcount

9. framebuffer-memorycount
10. framebuffer-conX-enable (muss aktiviert werden)
11. framebuffer-conX-index
12. framebuffer-conX-busid
13. framebuffer-conX-pipe
14. framebuffer-conX-type
15. framebuffer-conX-flags

Alles anzeigen

Für das Patches des [DVMT](#) sind dann wichtig `framebuffer-patch-enable` `framebuffer-stolenmem` `framebuffer-fbmem` `framebuffer-unifiedmem`

Beispiel 32MB BIOS, 19MB stolen (framebuffer) 9MB fbmem (cursor) 1536MB unifiedmem (vram)

Das sind die Werte meiner HD 515 von oben und das Prinzip wie vorher im Clover Configurator eingefügt

Properties Key*	Properties Value*	Value Type
framebuffer-stolenmem	00003001	DATA ↕
framebuffer-patch-enable	01000000	DATA ↕
framebuffer-unifiedmem	00000060	DATA ↕
AAPL,ig-platform-id	00001E19	DATA ↕
framebuffer-fbmem	00009000	DATA ↕

Oder auf eine andere Art

Code

1. DevicesProperties
2. PciRoot(0x0)/Pci(0x2,0x0)
- 3.
4. AAPL,ig-platform-id AAAeGQ==
5. framebuffer-patch-enable AQAAAA==
6. framebuffer-stolenmem AAaWAQ==
7. framebuffer-fbmem AACQAA==
8. framebuffer-unifiedmem AAAAYA==

Connector

Code

1. ID: 191E0000, STOLEN: 34 MB, FBMEM: 21 MB, VRAM: 1536 MB, Flags: 0x0000050FTOTAL STOLEN: 1 MB (1048579 bytes), TOTAL CURSOR: 1 MB (1572864 bytes), OVERALL: 2 MB (2633731 bytes)
- 2.
3. Model name: Intel HD Graphics 515
- 4.
5. Camelia: Disabled, Freq: 1388 Hz, FreqMax: 1388 HzMobile: 1, PipeCount: 3, PortCount: 3, FBMemoryCount: 3
- 6.
7. [0] busId: 0x00, pipe: 8, type: 0x00000002, flags: 0x00000098 - LVDS
8. [1] busId: 0x05, pipe: 9, type: 0x00000400, flags: 0x00000187 - DP
9. [2] busId: 0x04, pipe: 10, type: 0x00000400, flags: 0x00000187 - DP
- 10.
11. 00000800 02000000 9800000001050900 00040000 8701000002040A00 00040000 87010000

Alles anzeigen

Beispiel

```
00 00 08 00 02000000 98000000
```

- framebuffer-conX-enable01000000
- 00 framebuffer-conX-index 00000000
- 00 framebuffer-conX-busid 00000000
- 08 framebuffer-conX-pipe 08000000
- 02000000 framebuffer-conX-type 02000000
- 98000000 framebuffer-conX-flags 98000000

Komplett sieht es dann so aus. Das conX ist fortlaufend von Anschluss und der Letzte (3) ist in diesem Fall ein Dummy

Code

1. DevicesProperties
- 2.
3. PciRoot(0x0)/Pci(0x2,0x0)
- 4.
5. AAPL,ig-platform-idAAAGQ==

6. device-idHhkAAA==
7. framebuffer-patch-enable AQAAAA==
8. framebuffer-con0-enable AQAAAA==
9. framebuffer-con0-index AAAAAA==
10. framebuffer-con0-busid AAAAAA==
11. framebuffer-con0-pipeC AAAAAA==
12. framebuffer-con0-type AgAAAA==
13. framebuffer-con0-flagsm. AAAAAA==
14. framebuffer-con1-enable. AQAAAA==
15. framebuffer-con1-index. AQAAAA==
16. framebuffer-con1-busid. BQAAAA==
17. framebuffer-con1-pipe. CQAAAA==
18. framebuffer-con1-type AAQAAA==
19. framebuffer-con1-flag. shwEAAA==
20. framebuffer-con2-enable AQAAAA==
21. framebuffer-con2-index AgAAAA==
22. framebuffer-con2-busid BAAAAA==
23. framebuffer-con2-pipe CgAAAA==
24. framebuffer-con2-type AAQAAA==
25. framebuffer-con2-flags hwEAAA==
26. framebuffer-con3-enable AQAAAA==
27. framebuffer-con3-index ////w==
28. framebuffer-con3-busid AAAAAA==
29. framebuffer-con3-pipe AAAAAA==
30. framebuffer-con3-type AQAAAA==
31. framebuffer-con3-flags IAAAAA=

Alles anzeigen

Beitrag von „Noir0SX“ vom 4. November 2017, 14:56

VirtualSMC

...Wildcard ...

... ..

Beitrag von „Noir0SX“ vom 25. Juli 2018, 20:40

LiluFriend

... ..

Beitrag von „Noir0SX“ vom 17. Oktober 2018, 16:55

Helferlein Kext Update & Erstellen

Um die (seine) Kexte aktuell zuhalten, kann man das [Tool \(Kext-Updater\)](#) von [@Sascha 77](#) benutzen

Einen Kext selber erstellen, brauchen hier die wenigsten. Meist sind die Versionen unter Releases zu finden und gut.

Interessant wird es für Leute:

- die Probleme haben, solange in der Repo ein Fix vorliegt
- die nicht warten können
- wie ich das mal probierten
- oder selber den Code verbessern (obwohl der kann auch mit Xcode umgehen)
- usw. ...

Beispiel 1

Am Beispiel von [PMheart](#) Lilu-Plugin [CPUFriend](#), kann man schon sehen, das man einen Kext ganz schnell mal über das Terminal bauen kann.

Als erstes braucht man den Code (Dateien) vom jeweiligen Lilu-Plugin, geht also nicht nur mit dem CPUFriend-Kext.

In das heruntergeladene Verzeichniss laden wir den fertigen Debug-Kext von Lilu.

Das Beispiel CPUFriend zum Ausführen liegt dabei. Zu finden unter Tools mit Namen *build.command*. Diese Datei in das Terminal ziehen und ausgeführt werden dann die beiden Varianten des Plugins im gleichen Verzeichnis erstellt.

Klar auch hier braucht man XCode im Hintergrund, sonst wird das nix.

Der Weg bei den anderen Plugins ist gleich, nur das die Datei unter Tools an den entsprechenden Stellen mit dem jeweiligen Namen des Plugins angepasst und das Verzeichnis Tools kopiert werden muss.

Beispiel 2

Noch einfacher geht das ganze mit den in Python geschriebenen Script von [CorpNewt](#).

[Lilu-and-Friends](#) ist eigentlich selbst erklärend und braucht nach den Download nur ausgeführt werden.

Hier findet man nicht nur Lilu & Plugins, sondern auch viele andere Kexte.

```
#####
# Lilu and Friends #
#####
C 3. AppleBacklight - uses ACPI methods to control laptop backlight - pre-OS.12.4
C 3. AppleBatteryManager - implements an ACPI based battery manager kext
C 3. AirportFixup - for Airport (AirMac) & Airport Extreme cards
C 3. AirportMacFixup - for non-native Airport Broadcom Wi-Fi cards
C 3. AppleSMB - dynamic SMB patching
C 3. AppleSMBControl - for killing and bypass LAN
C 3. ApplePatchMacOS - for Huawei H3400 mobile GPUs
C 3. AppleSMBControl - for Broadcom's BCM5722 hardware and Mellinix Family LAN
C 3. AppleSMBControlFixup - for Apple's hardware network features
C 3. AppleSMBControl - applies patches updates for Broadcom BCM5722 based devices
C 3. CPUFriend - for dynamic power management data injection
C 3. DeviceCommander - aids in non-functioning audio after sleep
C 3. DeviceSleepFixup - enables it on greater set on certain graphics
C 3. DeviceLockdown
C 3. FakePCIID - spoof iGPU devices to allow drivers to load
C 3. FakeSMC - only FakeSMC
C 3. FakeSMC (Legacy) - only FakeSMC (Legacy)
C 3. GenericUSBInject - generic, non-Intel USB support
C 3. Hibernation (FakeSMC + Plugins) - contains FakeSMC + Hibernation plugins
C 3. Hibernation (Legacy) - contains legacy FakeSMC + Hibernation plugins
C 3. HibernationFixup - saves DRAM contents to RAM
C 3. IntelBacklight - replaces AppleBacklight for Intel graphics - pre-OS.12.4
C 3. IntelGraphicsFixup - alternative to the HD-panels patch
C 3. IntelGraphicsFixup - provides patches for Intel GPUs
C 3. IntelGraphicsFixup - for most Intel LAN
C 3. Lilu - for arbitrary kext, library, and program patching
C 3. NightShiftLockdown - unlocks night shift on unsupported OEMs
C 3. NvidiaGraphicsFixup - provides patches for Nvidia GPUs
C 3. RealtekRTL8100 - for RTL8100 LAN
C 3. RealtekRTL8100 - for RTL8100/8108 LAN
C 3. Video - for extended video playback
C 3. USBInjectAll - injects all USB ports
C 3. VendorPreference - sets device for desktop/laptop
C 3. VendorPreference - change DSDT number to your case count - 5 in Info.plist
C 3. LiluFriend - Lilu helper for those who keep kexts in /L/E and /S/L/E

N. Build Selected
N. Select All
N. Select None
N. ShowBuild Options
F. Profiles
B. Quit

Please make a selection: █
```

Beitrag von „Noir0SX“ vom 30. Juni 2020, 09:35

Um diesen Thread erweitern zu können wurden die Kommentare und Diskussionen hier her verschoben: [Diskussionen und Kommentare zu Bootflags zu Lilu & Plugins mit Beispielen](#) bitte hier nicht mehr antworten, danke