

```
1424     "@0,compatible",
1425     Buffer (0x08)
1426     {
1427         "NVDA,NVMac"
1428     },
1429
1430     "@0,device_type",
1431     Buffer (0x08)
1432     {
1433         "display"
1434     },
1435
1436     "@0,name",
1437     Buffer (0x0F)
1438     {
1439         "NVDA,Display-A"
1440     },
1441
1442     "@1,compatible",
1443     Buffer (0x08)
1444     {
1445         "NVDA,NVMac"
1446     },
1447
1448     "@1,device_type",
1449     Buffer (0x08)
1450     {
```

Line: 2293 Column: 41 C++ Tab Size: 4 Defi...

Um Drittanbieter Kexte

zu vermeiden und möglichst viele OS X Funktionen zu nutzen ist das Bearbeiten einer DSDT eine sinnvolle Maßnahme.

DSDT Was ist das?

Ist eine von mehreren im Bios integrierten Tabellen die das System über Hardware spezifische Funktion etc. informiert.

Zum bearbeiten muss diese vom komprimierten ACPI Machine Language (AML) mit Hilfe eines Tools in die ACPI Source Language (ASL) übersetzt werden. Da die meisten Mainboards für Windows ausgelegt werden und entsprechend nur damit getestet werden, kann es bei anderen Betriebssystemen, wie Mac OS X (Unix) und Linux, zu Problemen kommen. Eine weitere Möglichkeit die für uns Hackintosh Benutzer interessant ist das ändern von Geräte Adressen

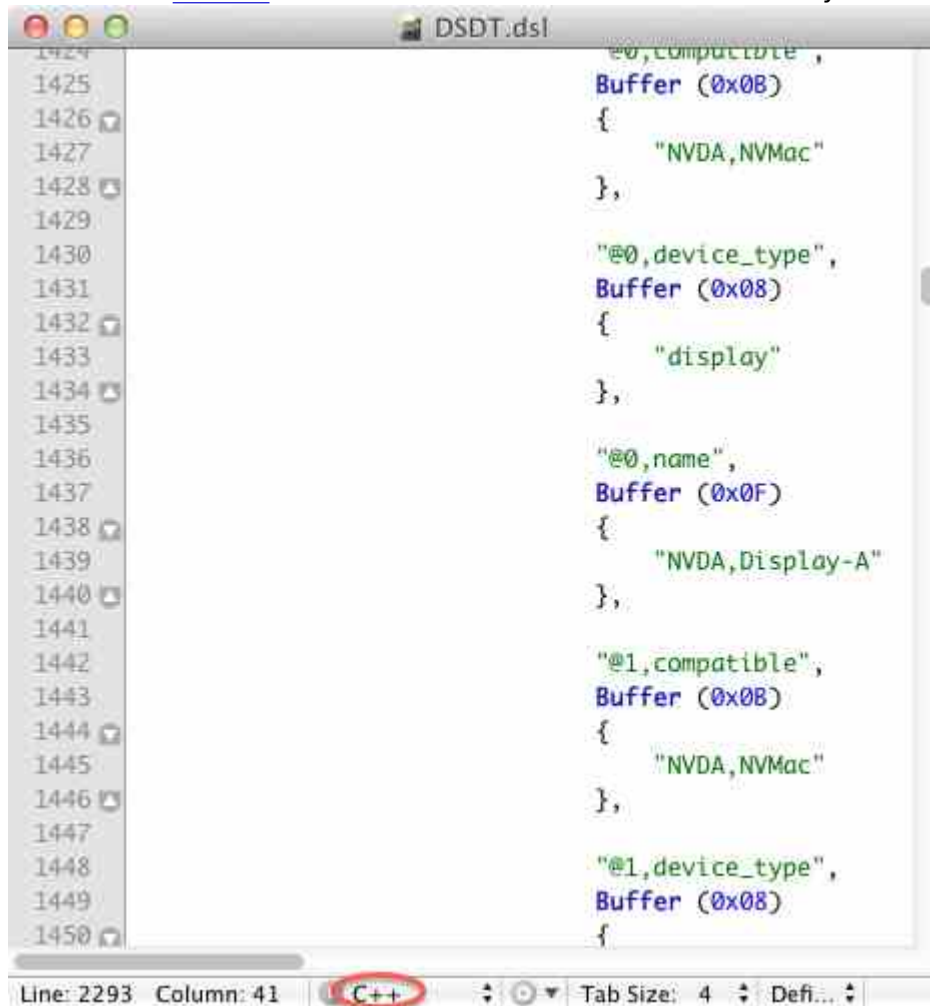
um dem System kompatible Hardware vorzutauschen (geht natürlich nur bei ähnlicher Hardware).

[Wikipedia Artikel](#)

Tools

Wenn man eine komplett neue DSDT patchen will, z. B nach einem [Bios Update](#) ist der [DSDTFixer](#) ein gutes Tool, er behebt gleich lästige Fehler, um die man sich später nicht mehr kümmern muss.

Mir persönlich gefällt DSDTSE nicht, weil man auf ein sehr kleines Fenster begrenzt ist. Deshalb benutze ich [iASLMe](#) und einen Text Editor mit Source-Code Syntax-Hervorhebung z. B. [TextMate](#)



The image shows a screenshot of a text editor window titled "DSDT.dsl". The editor displays DSDT code with syntax highlighting. The code is organized into sections, each starting with a label like "@0,compatible", "@0,device_type", "@0,name", "@1,compatible", and "@1,device_type". Each section is followed by a "Buffer" declaration and a block of code enclosed in curly braces. The code includes string literals such as "NVDA,NVMac" and "NVDA,Display-A". The editor's status bar at the bottom shows "Line: 2293 Column: 41" and a "C++" icon circled in red. The window title bar includes standard macOS window controls (red, yellow, green buttons) and the filename "DSDT.dsl".

```
1424 @0,compatible",
1425 Buffer (0x08)
1426 {
1427     "NVDA,NVMac"
1428 },
1429
1430 "@0,device_type",
1431 Buffer (0x08)
1432 {
1433     "display"
1434 },
1435
1436 "@0,name",
1437 Buffer (0x0F)
1438 {
1439     "NVDA,Display-A"
1440 },
1441
1442 "@1,compatible",
1443 Buffer (0x08)
1444 {
1445     "NVDA,NVMac"
1446 },
1447
1448 "@1,device_type",
1449 Buffer (0x08)
1450 {
```

Als Quellcode Art C++ auswählen.

Als Grundlage für Geräte Adressen und anderes sind die [hier](#) erwähnten tools Pflicht.

Tutorials:

1. Grundlagen und compiling Errors

2. [Nvidia-Injection.pdf](#)

3. [BUILT-IN Ethernet.pdf](#)

Da die meisten anderen Sachen wie Sleep (PTS, WAK) schon in den vorgefertigten DSDT's angepasst oder zu spezifisch sind, werde ich darauf zunächst nicht eingehen.

Es kann helfen in DSDT's und IOREG Dumps von original Mac's zu schauen.

Antworten auf diesen Beitrag bleibt zunächst geschlossen, da weitere Informationen Folgen