SOURCEFORGE

Clover EFI bootloader

Boot macOS, Windows, and Linux on Mac or PC with UEFI or BIOS firmware

Brought to you by: apianti, slice2009

This project can now be found here.

Devices

<u>Configuration</u> / Devices

```
<key>Devices</key>
<dict>
...
</dict>
```

Parameter group for tweaking setting affecting PCI devices.

Table of Contents

- <u>Inject</u>
- <u>Properties</u>
- <u>FakeID</u>
- <u>Audio</u>
  - <u>Audio / Inject</u>
  - <u>Audio / ResetHDA</u>
  - <u>Audio / AFGLowPowerState</u>

- <u>USB</u>
  - <u>USB / Inject</u>
  - <u>USB / FixOwnership</u>
  - <u>USB / AddClockID</u>
  - <u>USB / HighCurrent</u>

- <u>UseIntelHDMI</u>
- <u>AddProperties</u>
- <u>NoDefaultProperties</u>
- <u>SetIntelMaxBacklight</u>

Inject

```
<key>Inject</key>
<false/>
```

Enabling it will cause the automatic injection to turn off and allow the injection of a custom string from `Properties`.

Properties

```
<key>Properties</key>
<string>0207364862FA54HG345</string>
```

To create your own string you need the tool *gfxutil* that, for example, is bundled with *DarwinDumper*. You will need to create an xml file using a device path in standard notation as header and list properties, which will be injected. The example shows a file used to enable a video card:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>PciRoot(0x0)/Pci(0x2,0x0)</key>
    <dict>
        <key>AAPL,HasPanel</key>
        <data>AQAAAA== </data>
        <key>built-in</key>
        <data>AA==</data>
        <key>class-code</key>
        <data>AAADAA==</data>
        <key>device_type</key>
        <data>ZGlzcGxheQA=</data>
        <key>model</key>
        <data>R01BIDk1MAA=</data>
    </dict>
</dict>
</plist>
```

It is possible to obtain the currently used plist with *DarwinDumper*, to modify it and convert it into string form with the command:

```
./gfxutil -i xml -o hex devprop.plist devprop.hex
```

.. resulting in:

d300000001000000010000000c700000005000000002010c00d041030a00000000101060000027fff0400100000006d006f00640065006c0000000c0000004

The result needs to be used as a value for DeviceProperties. The same effect can be achieved by adding a *_DMS* method into the according section of your DSDT. It may be practical to use the string method when no DSDT is available yet.

Rev 4466

Now Devices > Properties can be written into config.plist in format as DarwinDumper do

For example:

▶ Details

More detailed instructions followed.
This way we can deprecate Arbitrary section.

FakeID

```xml
<key>FakeID</key>
<dict>
  <key>ATI</key>
  <string>0x68181002</string>
  <key>IntelGFX</key>
  <string>0x01268086</string>
  <key>NVidia</key>
  <string>0x0</string>
  <key>LAN</key>
  <string>0x436311ab</string>
  <key>SATA</key>
  <string>0x25628086</string>
  <key>WIFI</key>
  <string>0x431214E4</string>
  <key>XHCI</key>
  <string>0x0</string>
  <key>IMEI</key>
  <string>0x1E3A8086</string>
</dict>
```

This is a method to change PCI properties DeviceID and VendorID for the device will work with native drivers. In the example, AMDRadeonHD7850 has unsupported DeviceID=0x6819. Change to 0x6818 - Dell Wireless 1595, DeviceID=0x4315 is not supported. Change to 0x4312 - Marvell Yukon 8056, DeviceID=0x4353. Change to 0x4363.

There are other known substitutions for unsupported devices.

This substitution will work if InjectATI (Nvidia, Intel) is set. Or if FixDsdtMask set for the device.

Likewise, the IMEI fix will only work if the DSDT Patch `AddMCHC_0008` is enabled.

Audio

```
<key>Audio</key>
<dict>
...
</dict>
```

Parameter group for tweaking setting affecting audio devices.

Audio / Inject

```
<key>Inject</key>
<string>Detect</string>
```

Property injection for the sound chip. Only works when the DSDT defines `Device (HDEF)`. If you renamed it, you can also inject the other properties differently. Do not use this parameter with VoodooHDA. Possible options are:

- `No` - Injection is turned off
- `Detect` - Automatic detection of the sound chip and injection of its ID as layout ID. Actually this is nonsense but still very popular. Does not do any harm and affects the display of sound properties in System Profiler.
- `883` - Decimal number representing. Refers to Realtek ALC883 in this case.
- `0x373` - Same as above in hexadecimal.

These numbers are incorrect, you will need to find the correct value and possibly replace the layout file bundled with AppleHDA to get the chip working.

Audio / ResetHDA

```
<key>ResetHDA</key>
<true/>
```

Audio controller initialization. Some users have non-working sound after cold boot but works after restart or wake (even on Windows!). This is a workaround that works at early boot, so it will affect Windows as well.

Audio / AFGLowPowerState

```
<key>AFGLowPowerState</key>
<false/>
```

This helps remove cracking sounds at audio output after idle mode, so sound card is always on.

USB

```
<key>USB</key>
<dict>
...
</dict>
```

Parameter group for tweaking setting affecting PCI devices.

USB / Inject

```
<key>Inject</key>
<true/>
```

Injects USB properties. You can turn it off for whatever reason, if you need. It is also disabled if the *DSDT patch mask* matches `0x1000` to prevent data duplication.

USB / FixOwnership

```
<key>FixOwnership</key>
<true/>
```

It is possible to leave USB injection enabled and only turn off the ownership fix.

This fix is not relevant for UEFI booting.

USB / AddClockID

```
<key>AddClockID</key>
<true/>
```

- `<true/>` - Enables a good, deep sleep, which cannot be exited by keyboard or mouse input.
- `<false/>` - The PC will possibly sleep and it can be woken up by keyboard or mouse; or it will be automatically woken up by some attached device

Injects the property "AAPL,clock-id" with a unique identifier for each device. Set it to your liking.

Requires USBInjection to be enabled.

Default value is set to disabled.

USB / HighCurrent

```
<key>HighCurrent</key>
<true/>
```

More power needed to charge iPad from USB ports.

UseIntelHDMI

```
<key>UseIntelHDMI</key>
<false/>
```

If TRUE, *hda-gfx=onboard-1* will be injected into the GFX0 and HDEF devices. Also, if an ATI or Nvidia HDMI device is present, they'll be assigned to *onboard-2*. If FALSE, then ATI or Nvidia devices will get *onboard-1* as well as the HDAU device if present.

AddProperties

```
<key>AddProperties</key>
<array>
  <dict>
    <key>Device</key>
    <string>NVidia</string>
    <key>Key</key>
    <string>AAPL,HasPanel</string>
    <key>Value</key>
    <data>AQAAAA==</data>
  </dict>
  <dict>
    <key>Device</key>
    <string>NVidia</string>
    <key>Key</key>
    <string>AAPL,Haslid</string>
    <key>Value</key>
    <data>AQAAAA==</data>
  </dict>
</array>
```

Using device properties injection may be insufficient for some cases (new device, new standard, new OS), so this function lets you customize these injected properties.

You may write as many properties as you want for these devices: * ATI * NVidia * IntelGFX * LAN * WIFI * Firewire * SATA * IDE * HDA * HDMI * LPC * SmBUS * USB

NoDefaultProperties

```
<key>NoDefaultProperties</key>
<false/>
```

This key will affect DSDT fixes and force them to generate an empty _DSM. For example, if you enable `FIX_DISPLAY` Clover will create a device for the graphics card but with an empty _DSM. `AddProperties` and `FakeID` values will still be injected.

This works only for Display, Sound, LAN and WiFi.

SetIntelMaxBacklight

Fix MaxBrightness for Intel HD3000/4000 proposed by Dr.Hurt here
New setting in config plist

```
<key>SetIntelMaxBacklight</key>
```