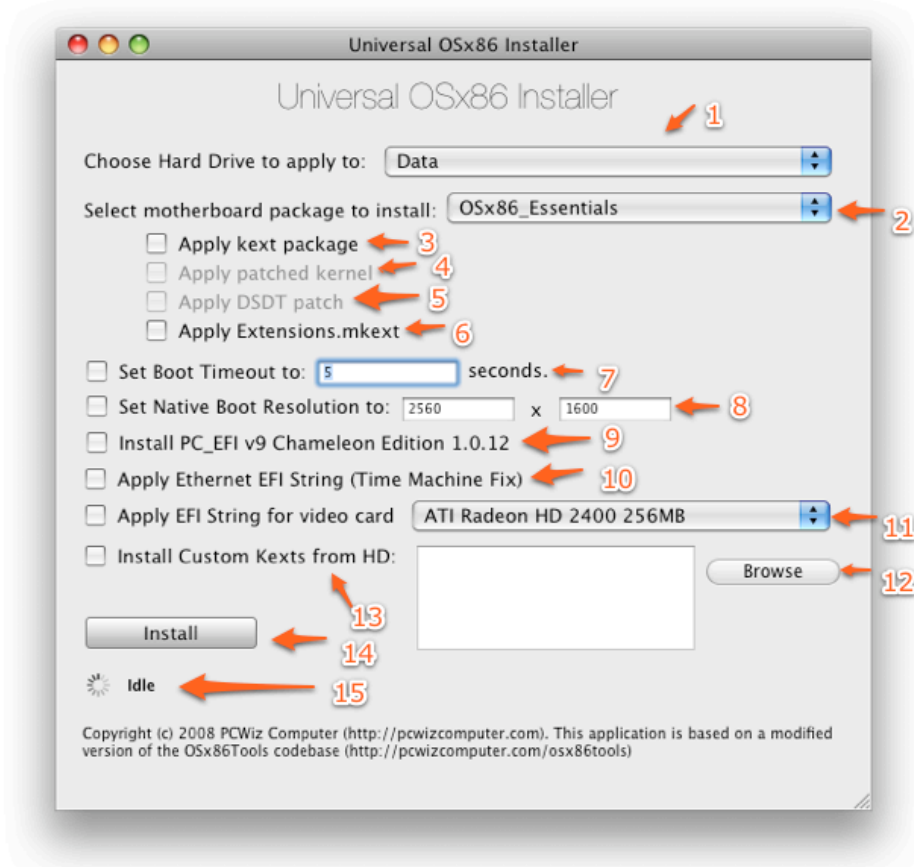


## Universal OSx86 Installer Documentation

Application developed and maintained by <http://pcwizcomputer.com>

Based on the OSX86Tools Codebase (<http://pcwizcomputer.com/osx86tools/>)



Welcome to Universal OSx86 Installer and thanks for downloading. This application may begin your OSx86 adventures, and we hope it will assist to smooth the process. UOI is an application designed to give you working hardware support and a good OSx86 experience in just a couple easy clicks after the initial installation. Above is the main screen of UOI and here are its main components:

- (1) Use this drop down menu to choose a hard drive to apply all your changes to
- (2) This is the drop down where you select the plugin to use (more on plugins later)
- (3) This checkbox allows you to select whether or not to install the kext package that comes with the plugin. This checkbox will be greyed out if your plugin does not support this function
- (4) This checkbox allows you to select whether or not to install the kernel that comes with the plugin. This checkbox will be greyed out if your plugin does not support this function
- (5) This checkbox allows you to select whether or not to install the DSDT file that comes with the plugin. This checkbox will be greyed out if your plugin does not support this function. More information on DSDT is available later in this user guide.

- (6) This checkbox allows you to select whether or not to install the Extensions.mkext (installed to /Extra) that comes with the plugin. This checkbox will be greyed out if your plugin does not support this function
- (7) Allows you to set a boot timeout by entering the number of seconds you want before the OS X bootloader starts loading OS X (allows you time to set kernel flags and boot options)
- (8) Enter a resolution here and select this option to set the correct resolution for the Apple boot logo
- (9) Check this box to have the app install the best current EFI Bootloader, PC\_EFI v9 (Chameleon 1.0.12)
- (10) This option applies an Ethernet device property string to fix UUID errors and issues with Time Machine on OSx86
- (11) Allows you to choose from a list of supported cards (or use your own) to apply an EFI string for. In most cases this will give you full video and graphics support without the need for patched kexts
- (12) Use the Browse button to browse your computer for kexts you want to install
- (13) Check this checkbox to install your selected custom kexts
- (14) The all important “Install” button. This applies everything you have selected to the chosen hard drive
- (15) Progress indicator will tell you what the application is doing during the Install process

## **The Basics**

As this application is meant for newcomers who are new to OSx86, it is important to cover some of the basics of OSx86 as these terms will appear repeatedly throughout the course of this guide.

### **What are kexts?**

Kexts, or kernel extensions are drivers that are installed to /System/Library/Extensions/. Kext files often have the “.kext” extension and allow for extra hardware support and in the case of OSx86, to replace Apple’s original drivers with ones that are optimized for use on PCs.

### **What is a kernel?**

A kernel is in short terms the “core” of the operating system. It controls basically all low level operating functions. Kernels exist in all Linux and UNIX based systems, including Mac OS X. In Mac OS X the kernel is located in the root of your hard drive (/) and is named “mach\_kernel” by default. If you have a vanilla based system (see below) then replacing the kernel is most likely not necessary, however if you have an AMD or SSE3 **incapable** processor then a patched kernel will likely be required.

### **What is a vanilla compatible system?**

A vanilla compatible system is a computer capable of running OSx86 with minimal modifications (no patched kernel, compatible with Apple software updates). You have a vanilla capable system if: you have a Core 2 based processor.

## What is EFI emulation?

EFI is the Extensible Firmware interface found in real macs. EFI is basically the “BIOS” of a Mac. For a computer to be properly recognized as a mac and to have the most compatibility it must have EFI. The problem here is that PCs do not have EFI.

Developers have counteracted this problem by using EFI emulation which enables basic EFI function calls through a specially modified bootloader. EFI distributions for OSx86 include PC\_EFI and Chameleon. **Nearly all OSx86 installs have some form of EFI emulation installed, so this is not necessarily something to worry about.** EFI emulation is required to use vanilla (unpatched) kernels and kexts, and to use GUID partition maps and EFI strings (more on that below)

## What is an EFI string?

EFI strings AKA device-properties are property list files (plists) converted to hexadecimal format that provide hardware support (in most cases this hardware support is for video cards). These hexadecimal strings are implemented in the com.apple.Boot.plist file which controls all boot parameters. EFI strings require an EFI emulation of some sort, and it is found in nearly all OSx86 installs today, so chances are that you have it.

## What is DSDT patching?

This is an area very rarely delved into, and is something that some of even the most experienced users dare not venture. In reality, its not dangerous, it just can be painfully hard at times. DSDT is a table found in your computer's BIOS that controls ACPI (power, time, etc.) functions. Starting at OS X 10.5.6 Apple decided to start checking for faulty DSDTs when it boots. Obviously the PCs DSDT comes back as faulty so it will not boot. The only way to counteract this is to make a dump of the DSDT in the BIOS and patch it properly for Darwin. First of all, you need a modified bootloader that will support DSDT override. This installer includes just that, using the “Install PC\_EFI v9 Chameleon Edition 1.0.12” checkbox. Now you need a patched DSDT file that will be copied to /dsdt.aml. To create DSDT dumps you can use the DSDT Patcher (also included in the Extras folder) but this is an advanced method, and often UOI plugins will include a DSDT file that you can install easily with this installer. However beware, even if you have the same motherboard DSDT's can vary by BIOS version so try to make sure that you have the same BIOS version as what is specified in the plugin.

That basically covers all of it! No worries if you don't understand all of this, you don't need to understand, its just there to teach a basic concept.

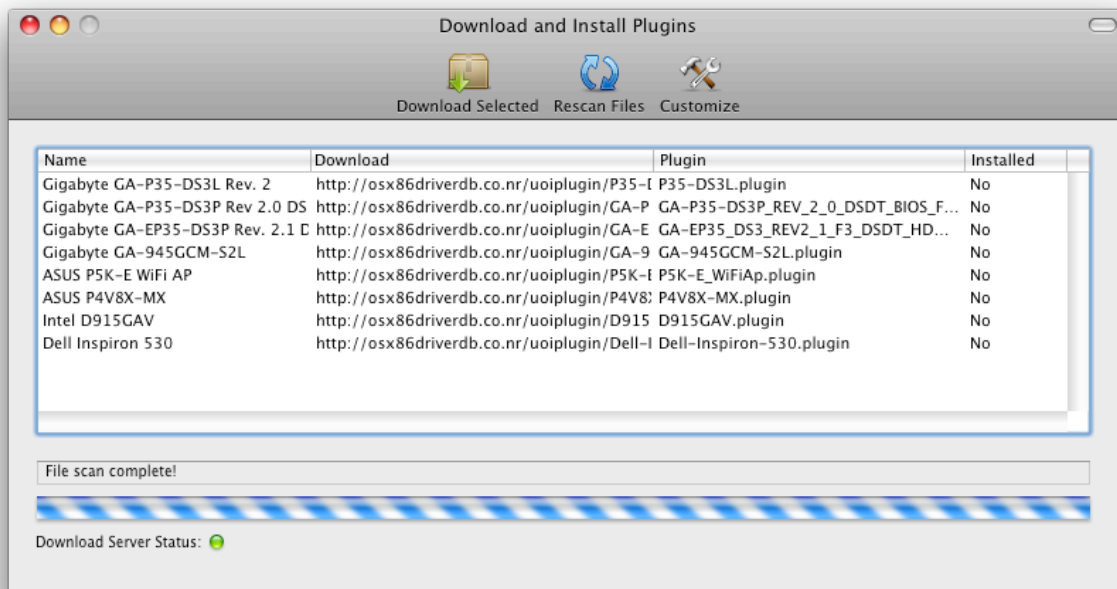
## Universal OSx86 Installer Plugins

Plugins form the foundation of Universal OSx86 Installer. Plugins are little bundles (with a .plugin extension) that are placed into the “Motherboards” folder of this application. Plugins can consist of a kext package, DSDT file, and a patched kernel. None of the components have to be there, so some plugins may include a DSDT, some may not, some may include a kernel, some may not, some may just have a DSDT dump and nothing else, its all interchangeable. Each of the 3 checkboxes below the motherboard plugin selection drop down menu may be greyed out if that option (dsdt, kexts, kernel) is not available in the plugin. **If all 3 checkboxes are greyed out, then this is a sign that: 1) The plugin is not for Universal OSx86 Installer, 2) The plugin is empty**

Universal OSx86 Installer comes preinstalled with one plugin (OSx86\_Essentials) that just contains 3 kexts that are useful and/or essential to running OSx86 and that are compatible with any computer in general.

## Downloading plugins

Plugins may be freely available on the internet, all you do is download the plugin and drop it in the Motherboards folder. Make sure you are downloading from trusted sources. The easy method, however is a build in plugin downloader built into the app. This plugin repository is kept up to date it is full of PCWiz Computer approved plugins that can be downloaded and installed with one click. To access this feature, click on the UInstaller menu and choose “Download more plugins...”. You will be presented with a screen like this:

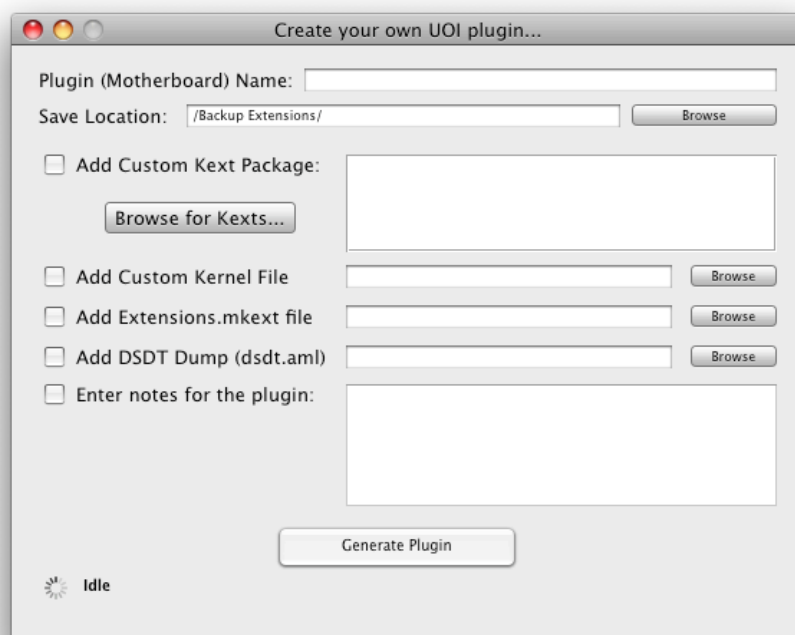


Each time you open the plugin downloader, it will download the latest XML file database from our server so you are always presented with the latest plugins in the database.

The download server (osx86tools.osx86.hu) must be online for you to download and view plugins. The plugin downloader is easy to use. In the table you see the name, plugin name, download path, and the installer indicator (tells you whether the plugin is already installed) for each plugin. To download a plugin simply select a plugin from the list and click the Download Selected button. Within a matter of seconds the plugin will be downloaded and installed, ready for use. **Each time you download a plugin you must restart the app for the plugin to appear in the list**

## Developing your own UOI plugins

So what if your motherboard doesn't have a plugin (and the chances of that are more than likely)? Create your own and send it to us for addition to the main repository, or distribute it yourself! Its very easy to create a plugin and here is how. Please note that all components are optional, so you could have just kexts, or maybe kexts ,dsdt, and kernel. Or maybe you only want to include DSDT. Its your choice! To start, go to the "UInstaller" plugin and click "Open UOI Plugin Generator...". You will be presented with this window:



Creating a plugin is easy. Choose a name for your plugin, browse for a save path. Then tick the checkboxes for what you want to include in the plugin (kexts, kernel, extensions.mkext, and dsdt) and also some notes. Then use the Browse buttons to select the respective files. Once everything is in check, click the Generate Plugin button at the bottom, enter your password when prompted, and your plugin will be saved in your desired save location.

To submit the plugin to our repository just go to the UInstaller menu and click Submit UOI plugin to be taken to our plugin submission form

## Looking inside a plugin

Wary of what's inside a plugin? To see the contents of a plugin go to the Motherboards folder, right click the plugin you want and choose "Show Package Contents" to see the inside of the plugin. This view can also be used to modify and add to a plugin.

## Adding your own EFI strings to the card

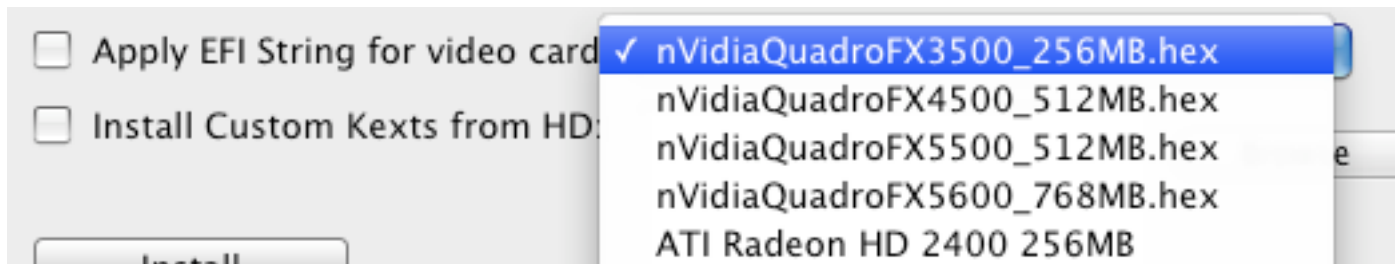
Have a Video card that is not supported by default but you found a string for it? UOI is designed to accept EFI string additions to the app in .hex or .plist form. A .hex file is just a text file containing a hex string. If you are cutting and pasting a hex string from the web into a .hex file MAKE SURE that the entire string is on one line and that there are no line breaks. Here is an example hex string for the GeForce 7900GS:

```
6c0200000100000001000000600200000d00000002010c00d041030a01000000
01010600000090101060000007ffff04000e0000004e00560050004d0000002000
0000010000000000000000000000000000000000000000000000000000000002200
00005600520041004d002c0074006f00740061006c00730069007a0065000000
080000000000000101c0000006400650076006900630065005f00740079007000
6500000000f0000004e5644412c506172656e740e0000006e0061006d00650000
000b000000646973706c617922000000400030002c0064006500760069006300
65005f00740079007000650000000b000000646973706c617922000000400031
002c006400650076006900630065005f00740079007000650000000b00000064
6973706c6179100000006d006f00640065006c0000001a0000006e5669646961
204765466f726365203739303020475320000000400031002c0063006f006d00
700061007400690062006c006500000000e0000004e5644412c4e564d61631400
0000400030002c006e0061006d0065000000120000004e5644412c446973706c
61792d4114000000400031002c006e0061006d0065000000120000004e564441
2c446973706c61792d4220000000400030002c0063006f006d00700061007400
690062006c006500000000e0000004e5644412c4e564d6163100000004e005600
43004100500000001800000004000000000003000c0000000000000700000000
1e00000072006f006d002d007200650076006900730069006f006e0000002e00
00006e5669646961204765466f7263652037393030204753204f70656e474c20
456e67696e65205b4546495d
```

Another format the app accepts is the .plist format which contains the same contents as a hex string but in readable property list form. Here is an example of a plist file for the GeForce 7900GS:

```
<?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(0x1)/Pci(0x1,0x0)/Pci(0x0,0x0)</key>
    <dict>
        <key>@0,compatible</key>
        <string>NVDA,NVMac</string>
        <key>@0,device_type</key>
        <string>display</string>
        <key>@0,name</key>
        <string>NVDA,Display-A</string>
        <key>@1,compatible</key>
        <string>NVDA,NVMac</string>
        <key>@1,device_type</key>
        <string>display</string>
        <key>@1,name</key>
        <string>NVDA,Display-B</string>
        <key>NVCAP</key>
        <data>
            BAAAAAAAAAwAMAAAAAAAAABwAAAAA=
        </data>
        <key>NVPM</key>
        <data>
            AQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==
        </data>
        <key>VRAM,totalsize</key>
        <string>0x10000000</string>
        <key>device_type</key>
        <string>NVDA,Parent</string>
        <key>model</key>
        <string>nVidia GeForce 7900 GS</string>
        <key>name</key>
        <string>display</string>
        <key>rom-revision</key>
        <string>nVidia GeForce 7900 GS OpenGL Engine [EFI]</
string>
    </dict>
</dict>
</plist>
```

Just take your GFX .plist or .hex file, drop it into the “Video” folder and it will appear in the Video card EFI string list as seen here:



**Please note: An incorrect or invalid hex or plist file used during installation will cause the EFI string creation to fail and abort**

### List of supported video cards:

Here is the list of supported cards for which strings are already included:

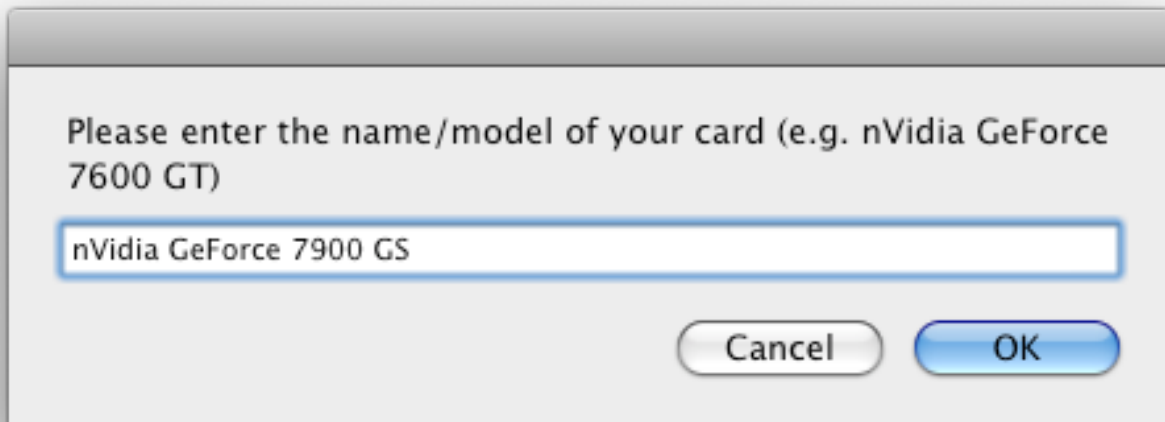
```
ATI Radeon HD 2400 256MB
ATI Radeon HD 2900 512MB
ATI Radeon HD 3400 256MB
ATI Radeon HD 3600 256MB
ATI Radeon HD 3800 512MB
ATI Radeon HD 3800 X2 1GB
ATI Radeon X1900XT 512MB
Intel GMA 950 (Laptop)
NVIDIA GeForce 6100 32MB
NVIDIA GeForce 6100 64MB
NVIDIA GeForce 6100 128MB
NVIDIA GeForce 6100 256MB
NVIDIA GeForce 6150 32MB
NVIDIA GeForce 6150 64MB
NVIDIA GeForce 6150 128MB
NVIDIA GeForce 6150 256MB
NVIDIA GeForce 6200 32MB
NVIDIA GeForce 6200 64MB
NVIDIA GeForce 6200 128MB
NVIDIA GeForce 6200 256MB
NVIDIA GeForce 6200 512MB
NVIDIA GeForce 6500 128MB
NVIDIA GeForce 6500 256MB
NVIDIA GeForce 6600 128MB
NVIDIA GeForce 6600 256MB
NVIDIA GeForce 6600 512MB
NVIDIA GeForce 6600GT 128MB
NVIDIA GeForce 6600GT 256MB
NVIDIA GeForce 6600GT 512MB
NVIDIA GeForce 6600LE 128MB
NVIDIA GeForce 6600LE 256MB
NVIDIA GeForce 6800 128MB
NVIDIA GeForce 6800 256MB
NVIDIA GeForce 6800 512MB
```

NVIDIA GeForce 6800GS 128MB  
NVIDIA GeForce 6800GS 256MB  
NVIDIA GeForce 6800GS 512MB  
NVIDIA GeForce 6800GT 128MB  
NVIDIA GeForce 6800GT 256MB  
NVIDIA GeForce 6800GT 512MB  
NVIDIA GeForce 6800 Ultra 256MB  
NVIDIA GeForce 6800 Ultra 512MB  
NVIDIA GeForce 6800XT 128MB  
NVIDIA GeForce 6800XT 256MB  
NVIDIA GeForce 6800XT 512MB  
NVIDIA GeForce 7025 32MB  
NVIDIA GeForce 7025 64MB  
NVIDIA GeForce 7025 128MB  
NVIDIA GeForce 7025 256MB  
NVIDIA GeForce 7050 32MB  
NVIDIA GeForce 7050 64MB  
NVIDIA GeForce 7050 128MB  
NVIDIA GeForce 7050 256MB  
NVIDIA GeForce 7100 32MB  
NVIDIA GeForce 7100 64MB  
NVIDIA GeForce 7100 128MB  
NVIDIA GeForce 7100 256MB  
NVIDIA GeForce 7100GS 64MB  
NVIDIA GeForce 7100GS 128MB  
NVIDIA GeForce 7150 32MB  
NVIDIA GeForce 7150 64MB  
NVIDIA GeForce 7150 128MB  
NVIDIA GeForce 7150 256MB  
NVIDIA GeForce 7200GS 128MB  
NVIDIA GeForce 7200GS 256MB  
NVIDIA GeForce 7300GS 64MB  
NVIDIA GeForce 7300GS 128MB  
NVIDIA GeForce 7300GS 256MB  
NVIDIA GeForce 7300GS 512MB  
NVIDIA GeForce 7300GT 64MB  
NVIDIA GeForce 7300GT 128MB  
NVIDIA GeForce 7300GT 256MB  
NVIDIA GeForce 7300GT 512MB  
NVIDIA GeForce 7300LE 64MB  
NVIDIA GeForce 7300LE 128MB  
NVIDIA GeForce 7300LE 256MB  
NVIDIA GeForce 7300LE 512MB  
NVIDIA GeForce 7300SE 64MB  
NVIDIA GeForce 7300SE 128MB  
NVIDIA GeForce 7300SE 256MB  
NVIDIA GeForce 7300SE 512MB  
NVIDIA GeForce 7500 64MB  
NVIDIA GeForce 7500 128MB  
NVIDIA GeForce 7500 256MB  
NVIDIA GeForce 7500 512MB  
NVIDIA GeForce 7600GS 256MB  
NVIDIA GeForce 7600GS 512MB  
NVIDIA GeForce 7600GT 256MB  
NVIDIA GeForce 7600GT 512MB  
NVIDIA GeForce 7800GS 256MB  
NVIDIA GeForce 7800GS 512MB  
NVIDIA GeForce 7800GT 256MB

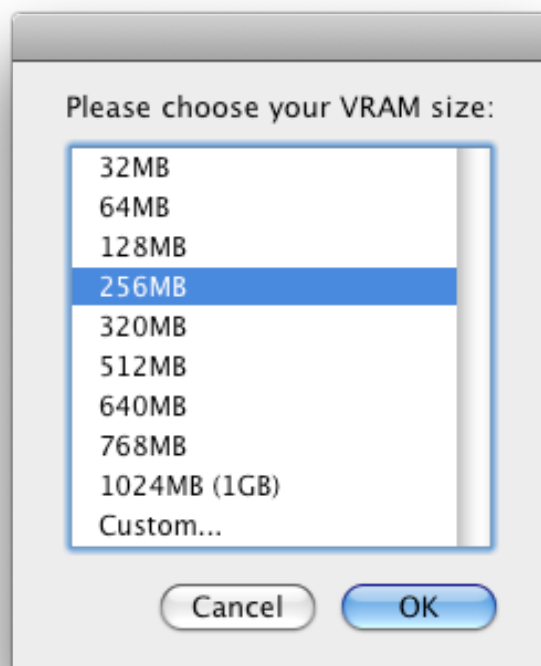
NVIDIA GeForce 7800GT 512MB  
NVIDIA GeForce 7800GTX 256MB  
NVIDIA GeForce 7800GTX 512MB  
NVIDIA GeForce 7900GS 256MB  
NVIDIA GeForce 7900GS 512MB  
NVIDIA GeForce 7900GT 256MB  
NVIDIA GeForce 7900GT 512MB  
NVIDIA GeForce 7900GTX 256MB  
NVIDIA GeForce 7900GTX 512MB  
NVIDIA GeForce 8200 32MB  
NVIDIA GeForce 8200 64MB  
NVIDIA GeForce 8200 128MB  
NVIDIA GeForce 8200 256MB  
NVIDIA GeForce 8300 32MB  
NVIDIA GeForce 8300 64MB  
NVIDIA GeForce 8300 128MB  
NVIDIA GeForce 8300 256MB  
NVIDIA GeForce 8300GS 128MB  
NVIDIA GeForce 8300GS 256MB  
NVIDIA GeForce 8300GT 128MB  
NVIDIA GeForce 8300GT 256MB  
NVIDIA GeForce 8300GT 512MB  
NVIDIA GeForce 8400GS 256MB  
NVIDIA GeForce 8400GS 512MB  
NVIDIA GeForce 8500GT 256MB  
NVIDIA GeForce 8500GT 512MB  
NVIDIA GeForce 8500GT 1GB  
NVIDIA GeForce 8600GT 256MB  
NVIDIA GeForce 8600GT 512MB  
NVIDIA GeForce 8600GT 1GB  
NVIDIA GeForce 8600GTS 256MB  
NVIDIA GeForce 8600GTS 512MB  
NVIDIA GeForce 8800GS 384MB  
NVIDIA GeForce 8800GT 256MB  
NVIDIA GeForce 8800GT 512MB  
NVIDIA GeForce 8800GT 1GB  
NVIDIA GeForce 8800GTS 320MB  
NVIDIA GeForce 8800GTS 512MB  
NVIDIA GeForce 8800GTS 640MB  
NVIDIA GeForce 8800GTX 768MB  
NVIDIA GeForce 8800 Ultra 768MB  
NVIDIA GeForce 9800GTX 512MB  
NVIDIA Quadro FX 3500 256MB  
NVIDIA Quadro FX 4500 512MB  
NVIDIA Quadro FX 5500 512MB  
NVIDIA Quadro FX 5600 768MB

## Creating your own NVIDIA Video Strings:

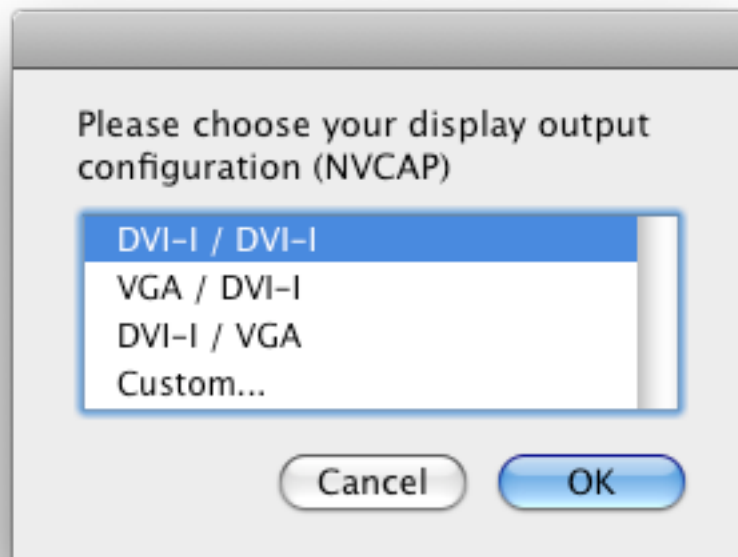
Say you have an NVIDIA video card that doesn't have a string? Well there's an easy way now. This application can dynamically create working strings for basically any NVIDIA card. To do so, tick the Apply EFI string option, and in the list go all the way down and choose "Custom NVIDIA...". When the installation procedure comes to this part, it will guide you through a series of steps to create the custom string. Step 1:



Here you just enter a name for your card. We are using nVidia GeForce 7900 GS as an example. Click OK to proceed.



All you need to do here is choose the amount of VRAM your card has. If you want to go advanced you can choose Custom and input your own VRAM data value. Click OK to proceed.



In this part you need to choose your display output configuration. Choose DVI-I / DVI-I if you have 2 DVI ports on your card, and in most cases this setting will work even if you have 1 DVI and 1 VGA. There are 2 other options available for your convenience, and if you want you can go Custom and input your own NVCAP string. Click OK to proceed. And that's it! Your custom string is generated and applied.

### **Bug reports/Contact Us**

To report bugs and to contact us please visit <http://pcwizcomputer.com> and use the Contact Us option to send in your bug reports, comments, and suggestions.

### **Credits**

PCWiz Computer - Application programming and development  
weaksauce12 and Hara Taiki - For the idea of motherboard plugins  
fassi - For the DSDT Patcher (<http://forum.insanelymac.com/index.php?showtopic=133683>)  
mackerintel - For Chameleon with DSDT Override (<http://forum.insanelymac.com/index.php?showtopic=132757>)  
netkas - PC\_EFI v9 (<http://netkas.org/?p=74>)  
Chameleon Team - Chameleon (<http://chameleon.osx86.hu>)

For more cool applications and more computer stuff in general visit <http://pcwizcomputer.com>!