

The Wayback Machine - <http://web.archive.org/web/20060906210619/http://vision.nyu.e...>

VideoToolbox

7300/7500/7600/8500/8600 120 Hz Video Driver

[VideoToolbox](#) [PsychToolbox](#) [Tips](#)

Search[about](#)[advice](#)[bugs](#)[changes](#)[contents](#)[download](#)

October 2, 1997

hi nano

i just discovered another unexpected behavior in your Resolutions Control Strip, which may be just another manifestation of the problem that Brian Stankiewicz reported. I just bought a \$4,500 grayscale monitor (the [BrightView](#)) that will synch to almost anything. Supposedly it supports the new DDC "plug'n play" digital communication standard. It has the fancy new DDC "plug'n play" 15-pin HD connector. From Griffin, I bought an HD-15 to HD-15 cable and a ["plug and play" adapter](#). This works, but seems to identify the monitor as an old-fashioned 480x640 fixed-frequency monitor.

when i boot up my 7500/100 with your new video driver and this monitor (with the Griffin adapter) and Mac OS 7.6.1, your new modified Resolutions control strip offers only 480x640 67 Hz in roman type (recommended) and all the others (including 640x480 120 Hz) in italic (at user's discretion). Apparently the boot-time identification fails to identify my monitor as DDC-compatible.

If, instead, I boot up with my Apple 17" multisynch then I get offered the wide range of resolutions I know and love. Since the Mac OS identifies the monitor type only at boot time, I can, after booting, manually transfer the connection to my new monitor (with the Griffin adapter) and everything works fine. Your control strip offers me all the options, and they all work.

best

denis

September 16, 1997

Nano's new driver worked fine for me on my PowerMac 7500 running Mac OS 7.6.1 driving an Apple 17" Multiscan. (The image was about 0.5" to the left of center; about 10 pixels were cut off at the left. This can be corrected by using the monitor's horizontal offset adjustment.) [Frans Cornelissen](#) reports that the new driver and control panel work fine on PowerMac 8600 running Mac OS 7.5.5 driving an Apple 20" Multiscan. [Brian Stankiewicz](#) says "I tested the new graphics driver on our 8500/150, and it worked fine with our Apple Multiscan monitor."

[Brian Stankiewicz](#) says "Under Mac OS 7.6, when I select the new Monitor Resolution panel with my AppleVision 1710AV monitor, I don't get the 120 Hz mode." [Nano](#) replies, "Sorry, I should've foreseen this. You need to disable the AppleVision software (or disconnect the monitor's ADB cable) in order to use the control strip "trick". The AV software overrides everything else." [Brian Stankiewicz](#) says, "I tried this, and it did change the Monitor Resolution strip selections, but it still did not provide the 120 Hz option, while I was running Mac OS 7.6. However, since I upgraded to Mac OS 8, your Monitor Resolution strip now does offer 120 Hz option." [If for some reason you don't want to upgrade to Mac OS 8, you could instead buy a [multisynch adapter](#) to set the video sense pins in a way that identifies the monitor as a multiscan (e.g. Apple 17" Multiscan).]

September 10, 1997

Our prayers have been answered! Nano Urbina, Apple Engineer extraordinaire, has updated his customized version of 7500/7600/8500/8600 graphics driver, which supports 640x480 120 Hz, to work with Mac OS 7 through 8. It's available for [download](#).

Here are the Sept. 10 release notes from Nano:

The enclosed driver should work under System 7.6 and greater, including MacOS 8. Please, test it and let me know if you have success with it or not. The included files consist of:

1. **Graphics Driver for the PowerMac [7300,] 7500, 7600, 8500, and 8600**, with all the different CPU cards (e.g. 8500/120, 8600/200, etc). This contains a new 640 x480 120 Hz resolution with better timing so it's more centered on the screen of an Apple Multiscan 17" display. [To use it, just drop it into your System Folder:Extensions folder and reboot. It offers all the features of the standard driver, plus a few extra resolutions. To remove it, take it out of the System Folder and reboot. Nano didn't mention the PowerMac 7300, but that computer's release notes indicated that it inherited the 7500's video circuitry, and [Andrew Derrington](#) has confirmed (9/18) that the new driver does work on the 7300. - denis]

2. **Monitor Resolution Control Strip Module** (This goes into the Control Strip Modules folder inside the System Folder). This new version, especially for VideoToolbox fans, allows the display of ALL the resolutions that a particular driver supports, without regard to the particular monitor that is connected. Just hold down the control key while clicking on the resolution module and you'll see the complete list of resolutions. [If you're using an AV monitor, you must **disable** the **AppleVision** software in order to use this control strip "trick".] All these resolutions are in italics, which means that a user will have to confirm the resolution change through a dialog on the screen. Do note that you can burn your display if you feed it an unsupported timing, so, **USE AT YOUR OWN RISK**.

[10/2/97. As noted in the subsequent correspondence above, the control strip won't offer any options if the computer thinks your display is fixed-frequency, which the Mac OS determines at boot time by looking at the sense pins. Unfortunately all the latest multisynch displays using the

"plug&play" digital communication standard (including the 1710AV) seem to identify themselves, via the sense pins, as fixed-frequency. You can fool your computer by booting with an old-fashioned multisynch attached and then switching to your plug&play monitor, or you can insert a "[multisynch adapter](#)" to program the sense pins to identify your monitor as multisynch. Nano hopes to fix the control panel, so that these shenanigans won't be necessary. - denis]

3. PDF file containing the **timing parameters** for the new 640 x480 120 Hz resolution.

As always, the driver and module are not supported by Apple. **Please don't call the Apple Helpline with any issues regarding this software.** If you do, the support group will track me down and will force me to stop supporting the VideoToolbox. This has already happened to one of my co-workers with another driver, so PLEASE, don't do it.

Do let me know if you have any troubles with the drivers. [Please send **bug reports** to both Nano, who might fix them, and to Denis, who will add them to this page to inform other users. [Nano & Denis](#)]

A note on driver versioning. [At boot time, the System loads the latest version of the driver that it finds, looking both inside the System File and for loose 'ndvr' files in the System Folder.] However, the 'vers' resources in the file [which the Finder can show and ResEdit can change] do not have anything to do with whether the driver loads or not. Native drivers have an internal version number that is used by the system software to determine the version of a particular driver (through an API call). The current shipping version is 1.4f4 (or so). The driver I'm enclosing has a version of 10.0f1, which means that it should not be superceded any time soon.

Denis, once you are comfortable that the driver works, you can add it to your releases after removing the old one. Do stress the situation with calling for support with this driver.

Regards,

Nano

???REQUIREMENTS (10/6/97, not yet confirmed): To recognize DDC monitors (e.g. Apple's 1710AV) the driver needs the new Display Manager, which is built into Mac OS 8, but can be installed on earlier systems by installing AppleVision 1.6.1. And the "Monitor Resolution Control Strip Module" needs Mac OS 8 in order to detect that you've pressed the Control key.

Several people have asked for higher frame rates, e.g. 150 Hz and 200 Hz. Nano replies, "All the monitors that I have, including the 20" Multiscan, max out at 120 Hz vertical refresh rate. That's why there isn't anything higher." Perhaps if we all chipped in and bought Nano a higher frame rate monitor, we might convince him to add driver support for it.

March 6, 1997

Flip Phillips reports that this 7500/7600/8500 Graphics Driver won't load in System 7.6. This makes sense. System 7.6 has a new video driver for 7300/7500/7600/8500/8600 computers. This driver is asynchronous (i.e. cscSetEntries returns immediately, the CLUT update happens later, at VBL time). As I recall, the Mac OS uses the highest version number driver that it finds. [I thought one might get this 7500/7600/8500 Graphics Driver driver to load by hacking it to give it a fictitious version number, higher than the one built into System 7.6, but the version number is buried too deep. The Mac OS uses something else (Driver Gestalt?), not the version resource, to determine the version.]

June 15, 1996

I've edited the note below slightly, renaming the driver from "7500/8500 ..." to "7500/7600/8500 Graphics Driver", to make it clear that the subsequently released 7600 is supported too. - denis.

Sent: 10/24/95

From: Nano Urbina, nano@apple.com

To: Denis Pelli, denis@cns.nyu.edu

Enclosure: 7500/7600/8500 Graphics Driver

Denis,

I'm enclosing a file that contains a revised version of the Graphics Driver for the Power Macintosh 7500, 7600, and 8500. Its version is 1.3f1. This has the following new resolutions:

1. For the Apple Multiscan 17" and 20", a new resolution of 640x480 at 120 Hz is available through the Sounds & Displays Control Panel. It will not show up on the Control Strip.
2. For the Multiscan 17" and 20", different refresh rates are available to be used while filming the display. Those rates are 59.94 Hz (NTSC), 50 Hz (PAL), and 48 Hz (film). Of course, a display that syncs to those refresh rates is required. The Apple Multiscan 17" and 20" will NOT sync to 48 Hz.

The "[7500/7600/8500 Graphics Driver](#)" file should be placed in the extensions folder.

It will NOT be supported by Apple. I will support it as long as time permits.

- Nano

1. (Added by dgp) Any application can control the display resolution by using Apple's new Display Manager: DMCheckDisplayMode and DMSetDisplayMode. You may also want to use GDVideo.c:GDGetDisplayMode to record what the resolution was before you change it, so you can restore it when you're done.

Try the TimeVideo demo, asking it to test all the resolutions.

2. (Added by dgp) Denis asks, "The 640x480x120 Hz mode on my Apple 17" multiscan is great. Rock solid. But the image isn't centered; about 10 pixels worth fall off the left edge of the screen. Can this be fixed?" Nano replies, "Can you just adjust the controls on the monitor to move the image to the right? This would be the easiest solution and is what I had to do. The better solution (requiring a new version of the driver) is to adjust the horizontal parameters so that the display puts it close to the middle, but that'll take some doing."
3. (Added by dgp, [Ben Singer](#) and [Peter Bex](#)) COMPATIBILITY: The driver works only on PowerMac 7500, 7600, and 8500 computers. The 120 Hz mode works on the Apple Multiscan 17 and 20, and the 1705 monitors, which are all rated to run up to 120 Hz (vertical). Surprisingly, the 120 Hz mode is *not* available when running the 1710 and 1710AV monitors, even though they are rated up to 120 Hz. Not surprisingly, the mode is not available when using an Apple Multiscan 15, since it's only rated up to 75 Hz vertical.

p.s. Nano also documented a pair of private status and control video driver calls for the 7500/8500 driver that can disable waiting for VBL's and control how long the driver waits for the CLUT to settle after writing each entry in the CLUT. Nano's documentation of the calls appears next to my implementation of access routines, `GDSetsDelay` and `GDGetDelay`, in `GDVideo.c`. Documentation of how to use `GDSetsDelay` and `GDGetDelay` appears at the top of `GDVideo.c`. (I've pasted a copy below.) Try running the TimeVideo demo. It allows you to enter any values you like, and they persist until reboot.

```
OSErr GDSetsDelay(GDHandle device, Boolean dontWaitForVBL, double nanoseconds);
OSErr GDGetDelay(GDHandle device, Boolean *dontWaitForVBLPtr, double
*nanosecondsPtr);
```

These two routines (in `VideoToolbox:GDVideo.c`) support features, i.e.

`cscSetTimeDelays` and `cscGetTimeDelays`, unique to the built-in video driver of the PowerMac 7500 and 8500. Both routines will merely return innocuous errors (-17 and -18) if used with any other video driver. Like most video drivers, the 7500/8500 video driver normally waits for VBL when loading the CLUT, but this can be overridden by the Boolean parameter `dontWaitForVBL`, in which case, from then on (until it's changed again or the machine is rebooted) the driver will load the CLUT immediately when it's called, without waiting for VBL. The second feature is that the video driver normally waits 800 ns after loading each RGB triplet to the CLUT, "to allow for the CLUT to settle and increment its address", and that time can be set by the parameter "nanoseconds". `GDGetDelay()` allows you to read the current setting of both parameters. `GDSetsDelay()` allows you to set both parameters. If the supplied value of "nanoseconds" is infinite or NAN then it's ignored and only "dontWaitForVBL" is set. The 7500/8500 video driver was written by Apple Engineer Nano Urbina, nano@apple.com. `GDSetsDelay()` and `GDGetDelay()` were written by Denis Pelli, to allow convenient access to Nano's custom control and status driver calls. Note that Nano Urbina has also supplied a newer version of this driver (included in the `VideoToolbox`) that supports several extra resolutions, of which the most noteworthy is 640x480x120 Hz, which works well on the Apple 17" Multiscan.

Thus far, using visual inspection in TimeVideo, I've been unable to notice any bad effects of not waiting for VBL and reducing the waiting interval from 800 ns to zero. I asked Nano Urbina, nano@apple.com, about it, since he put in the delays because he "does indeed get artifacts when updating the CLUT if we don't suppress interrupts while updating it." What were the artifacts? Nano replies, "The delays are there because of hardware requirements: When writing to the CLUT in a self-incrementing mode the hardware requires 800ns to update the address counter. Of course, this is worst case. (We write the CLUT starting address and then the R, G, and B. At the end of the B, the address counter for the CLUT will automatically increment to the next address.) I noticed some artifacts when doing palette animation -- one of the After Dark modules showed this very well."

Received: 10/24

From: Nano Urbina, nano@apple.com

To: Denis Pelli, denis@cns.nyu.edu

The following is the information on the private call for the 7500 and the 8500 to disable waiting for VBL's and to change the time that we wait for the CLUT to settle after writing each entry in the CLUT. As shipped, the driver waits for 800 nanoseconds after writing each RGB triplet, to allow the hardware to increment the CLUT address.

cscSetTimeDelays is used to set some flags and time delays used to write the CLUT. Use PBControl to issue the call, with the csParam[0] containing a pointer to a VDTimeDelay structure. Set the "validMask" field with the bits indicating which parameters you want to set.

cscGetTimeDelays is used to examine the current values of the flags and time delays. Use PBStatus instead of PBControl. Set the "validMask" field with the bits indicating which parameters you want to get.

The following is the definition of the fields of the VDTimeDelay struct:

flags, bit 0: DontWaitForVBL flag: When false, it means that the driver should wait for a VBL before writing the CLUT. When true, the driver should NOT wait for a VBL.

flags, bit 1: SetCLUTAddrRegTiming flag: When false, use the default timing. When true, use paramOne and paramTwo to specify the delay in nanoseconds.

validMask: Specifies which bits in flags are valid. Bits are valid when corresponding bit position is 1.

paramOne,paramTwo: When the SetCLUTAddrRegTiming bit of the validMask is set, then these fields specify the delay, in nanoseconds: nanoseconds.hi in paramOne and nanoseconds.lo in paramTwo. See Designing PCI Cards & Drivers for the Nanoseconds data structure.

```
enum{cscSetTimeDelays = 141}; // Used to set the different time delays
enum{cscGetTimeDelays = 141};
```

```
enum{gDontWaitForVBLMask=1,gSetCLUTTimingMask=2};
#if PRAGMA_ALIGN_SUPPORTED || __MWERKS__
#pragma options align=mac68k
#endif
struct VDTimeDelays{
  UInt32 flags;
  UInt32 validMask;
  UInt32 paramOne;
  UInt32 paramTwo;
};
typedef struct VDTimeDelays VDTimeDelays;
#if PRAGMA_ALIGN_SUPPORTED || __MWERKS__
#pragma options align=reset
#endif
```