New Technical Notes
Macintosh

®

Developer Support

Video Hardware Q&As
Hardware

Revised by: Developer Support Center May 1993
Written by: Developer Support Center October 1990

This Technical Note contains a collection of Q&As relating to a specific topic—questions you've sent the Developer Support Center (DSC) along with answers from the DSC engineers. While DSC engineers have checked the Q&A content for accuracy, the Q&A Technical Notes don't have the editing and organization of other Technical Notes. The Q&A function is to get new technical information and updates to you quickly, saving the polish for when the information migrates into reference manuals.

Q&As are now included with Technical Notes to make access to technical updates easier for you. If you have comments or suggestions about Q&A content or distribution, please let us know by sending an AppleLink to DEVFEEDBACK. Apple Partners may send technical questions about Q&A content to DEVSUPPORT for resolution.

New Q&As this month:
Removing VBLs from queue when application exits unexpectedly

Removing VBLs from queue when application exits unexpectedly
Date Written: 11/3/92
Last reviewed: 3/1/93

My application has VBL tasks that must run while my application is running. The problem I'm having is that when the application doesn't take the normal exit path, my VBL code continues to be called because it's still in the Macintosh Operating System's VBL run list, resulting in a crash. How can I get control back just before the application loses its context so that I can be guaranteed a chance to remove all installed VBLs?

___

The best way to do this is probably to patch the ExitToShell trap, which is called every time your application quits, regardless of the method. It's called not only for a traditional exit—that is, when you call the trap yourself or just fall out the end of your program's main routine—but also when a system error occurs or when MacsBug is used to exit. At that point, you can dispose of your VBL task safely. Your context will still be valid, but you might want to make sure A5 is valid by restoring it from the CurrentA5 global before using any global variables. A little extra sanity checking might also be in order, because by definition you’re arriving here because there’s a problem; you obviously can’t protect against every case, but you should try to take some precaution against making things worse.

When you've removed your VBL and anything else you need to, just jump through to the old ExitToShell address; you don't need to worry about removing your trap patch, because it will go away when your context is disposed of by the Process Manager.

PowerBook and screen savers

Date Written: 5/8/92
Last reviewed: 9/15/92

I'm looking for a Macintosh Toolbox routine that will allow me to turn down the backlight on a Macintosh PowerBook from within a screen saver to prevent screen burn and save battery life. Is there such a thing?

___

Turning down the backlight won't prevent screen burn. Screen burn can be prevented only by either shutting the system off or letting the PowerBook enter its native sleep mode.

In an RGB monitor the phosphor that illuminates each pixel is what causes screen burn. By setting the pixels to black (the phosphor isn't active) or rapidly changing the colors of an RGB screen (as with a screen saver), you can prevent screen burn. While effective on an RGB display, setting the pixels to black may actually cause screen burn on a PowerBook. The reason is that all the PowerBooks have a liquid crystal display (LCD), which can be burned by white pixels, black pixels, or repeating patterns on the screen over a period of time. For this type of display the only good way to save the screen is to power it off.

Only the Power Manager has access to the chip that shuts the screen off. After a certain amount of time, the Power Manager makes the function calls to put the system to sleep. (These calls are documented in Chapter 31 of *Inside Macintosh* Volume VI.) At this time the Power Manager signals the chip to turn the screen off. There's no direct interface between the user and the chip to achieve this. It's best to let the PowerBook's native screen-saving mechanism (sleep mode, which shuts off the screen) work as is. This also has the benefit of saving the precious battery power that would be used by the screen saver.

By the way, if your PowerBook screen has ghost images because you've left it on too long without going into sleep mode, letting the screen sleep or shutting down your computer for at least 24 hours will probably make the ghost images go away. Although there's no hard and fast rule, usually ghost images caused by your system being on for less than 24 hours won't be permanent if the screen is rested for an equal amount of time. Any ghost images caused by the system being on for greater than 24 hours may be permanent.

Monitor/video camera synchronization extension

Date Written: 3/9/92
Last reviewed: 5/21/92

Didn’t I see a system extension somewhere that removes the video flicker that often happens when you try to shoot video from a monitor?

___

Yes, there is a system extension for synchronizing your monitor with your video camera. It is called VideoSync and is available from APDA (800-282-2732). VideoSync works with System 6.0.4 or later, an Apple 12 or 13" color monitor, and one of the following video cards:

Macintosh II Video Card
Apple High Resolution Video Card
4•8 Display Card
8•24 Display Card
8•24GC Display Card

VideoSync does not work with built in video (i.e. it cannot be used with the LC, LCII, IIsi, or IIci)

Determining Macintosh screen resolution
Date Written: 1/21/92
Last reviewed: 4/22/92

What’s the best way to determine the dots-per-inch (dpi) resolution of a Macintosh screen? Can we calculate it based on the pixel map fields?

___

QuickDraw doesn’t provide any method for determining the screen’s dpi; the resolution is assumed always to be 72 dpi. Though the pixel map contains hres and vres fields, their values are ignored by the system and are used only locally within an application. By default these resolution fields are set to the fixed type value 0x00480000.

Code for finding video slot before using SlotVinstall for VBLs
Date Written: 12/17/91
Last reviewed: 2/17/91

I want to synchronize my routine with a Macintosh Quadra 900 built-in video interrupt, but this interrupt seems to always be disabled. I call the interrupt if it uses the same SIntInstall call, replacing the internal video slot number ($9) by the slot number of an external card ($D). Is SIntInstall the correct method for video synchronism with a Macintosh Quadra?

___

Your routine should be calling SlotVInstall, which is used for vertical blanking tasks on Color QuickDraw monitors, instead of SlotIntInsall, which is used by NuBus card drivers to install handlers for interrupts generated by their cards. SlotVInstall is documented on page 567 of *Inside Macintosh* Volume V.

You need to determine the slot of the video in question by examining the DCE of the GDevice. Get the GDHandle of the GDevice in question and then determine the slot with the following C code:

```
void
GetSlot(GDHandle gDev,short *slot)
```

```
{  short    refNum;

 refNum = (**gDev).gdRefNum;   // video driver refNum for this GDevice
 *slot = (**(AuxDCEHandle)GetDCtlEntry(refNum)).dCtlSlot;
                  // slot in which this video card sits
}
```

You can then use the returned slot number to install slot VBLs with the SlotVInstall function. This code should work regardless of the CPU (provided it has Color QuickDraw), and whether or not the screen in question is internal or external video.

[Note that this is the standard technique for finding out which slot a video card is in; the above code can be used by anyone who wants to know which slot a specific video card is in.]

Changing screen size for Mac server with no display connected

Date Written: 12/11/91
Last reviewed: 12/19/91

How can I make the screen size bigger than 9 inches for our Macintosh II server running System 7 without a monitor? Changing the dummy 'scrn' resource rect in the System Folder to 0 0 480 640 has no effect.

___

Macintosh video cards and circuitry detect the kind of monitor attached by checking three sense lines in the display connector. The monitor considered to be attached (and the memory allocated to accommodate it) is dependent on which lines are pulled low during card initialization. You'll find this scheme documented in Chapter 12 of the Guide to Macintosh Family Hardware, 2nd Edition, and in the article on the 8•24 GC card in *develop* issue #3. (Back issues of *develop* are available on the Developer CD.) A summary of monitor connections is available in the Q&A Stack or in the Q&A folder in the Dev Tech Answers library on AppleLink.

The simplest way to fool the Macintosh into believing that a larger monitor is attached is to build a terminator for the video port that sets the sense lines appropriately. For example, to simulate a 640 x 480 pixel monitor, sense lines 2, 1, and 0 (pins 10, 7, and 4) should be 1, 1, and 0 (where 0 is grounded and 1 is free). We're not aware of any commercially available video terminators. Building one should be simple; as always, Apple isn't responsible for problems caused by non-Apple hardware attachments.

Adding video device to slotless Macintosh desktop

Date Written: 10/22/91
Last reviewed: 10/22/91

My video driver for Macintosh 68000 SE-type models installs itself as a driver, but since there's no Graphics Device Manager in the ROM, it can't add itself to the desktop by creating a GDevice using NewGDevice and putting itself into the DeviceList. Do I need to add my region to the region whose handle is in GrayRgn and patch the QuickDraw bottlenecks to divert drawing to my VRAM, or is there an easier way?

___

For the Macintosh Plus, SE, Classic, and Portable, your idea of how to add your video device to the system is right on: Add your region to the region whose handle is in GrayRgn and patch the QuickDraw bottlenecks to divert drawing to your VRAM. You didn't ask for implementation details, so this is just a confirmation that your approach is good.

QuickDraw 32-bit video RAM access and boundaries

Date Written: 10/15/91

Last reviewed: 10/15/91

Is there a way to let QuickDraw access video RAM in 32-bit mode, but on a word boundary ? Usually QuickDraw reads/writes on word boundaries, but sometimes it takes advantage of the 68020/30 and reads not a word boundary.

___

QuickDraw optimizes whenever it can to make efficient frame buffer accesses, but there isn’t a way for you to control how it makes accesses itself. That is, video hardware MUST be able to handle all flavors of accesses. This is because the user can position a window or other object on the screen that is not on a long/word boundry. However, some applications—HyperCard being a prime example—create their windows on aligned boundries and allow movement only to boundaries, so you’ll see a “snap”-to-grid kind of effect. Your application can do so as well, but that’s as far as you should go.

Installing a Macintosh VBL task for on-board video

Date Written: 9/4/91
Last reviewed: 9/16/91

Installing a VBL task doesn’t seem to work on Macintosh systems with on-board video. I get the slot number for a video card by extracting the second nibble of the base address of the screen’s pixMap; however, on-board video computers report that the video card is in slot $B but the video card doesn’t seem to exist in any slot. Is there a better way to determine a video card slot number? Is there any way to attach a VBL task to on-board video? How can I determine whether the main monitor is using on-board video?

___

You should be getting the slot number for the video from the AuxDCE entry, not from the base address of the pixMap *(Inside Macintosh* Volume V, page 424). You’ll need to get the gdRefNum for the desired monitor from the device list *(Inside Macintosh* Volume V, Chapter 5). For example, to install a VBL task for the main screen, do something like this:

```
GDevHand := GetMainDevice;
IF GDevHand = NIL THEN HandleError
ELSE
 BEGIN
   mainGDRefNum := GDevHand^^.gdRefNum;
   DCEHand := AuxDCEHandle(GetDctlEntry(mainGDRefNum));
   IF DCEHand = NIL THEN HandleError
   ELSE retCode := SlotVInstall(@myVBLTask,DCEHand^^.dCtlSlot);
 END;
```

This should work regardless of the kind of video used.

Main versus application screens with multiple displays

Date Written: 8/20/91
Last reviewed: 9/16/91

On a multiple-display system, most commercial software comes up on the primary screen, even though it may need the color screen next door. Is there a utility for changing the default screen to a different one AFTER startup so I can switch to a color screen for certain applications without having to reboot?

___

A user can use Monitors to change the main screen (the one with the menu bar).If an application or utility were to change the main screen whenever it wants, then every time you run a new application, the menu bar would be moved on you.This would get reeeeeeally ugly. If commercial applications need (or run better on) one screen over another, they should do the right thing and look for the right screen instead of the main screen. The code to do this is very easy. This is not the job for a utility package.

You need to reboot the machine each time you change the main screen, because the Macintosh does a lot of caching of information and building internal structures early in the boot process, which need to be changed each time the screen information is changed. It might be possible to do this, but it would require a lot of patches to the system, as well as a deep knowledge of the internals of the ROM and OS. You'll also run into a lot of compatibility problems.

SetEntries and setting CLUT asynchronously
Date Written: 9/16/91
Last reviewed: 9/16/91

I want to call the device driver Control function on my screen device to set the CLUT asynchronously, so that it won't wait until vertical retrace time like SetEntries does, causing an average delay of half the screen refresh rate, which is too much for my application. Does the firmware in the Apple video boards support asynchronous operation, or do I need to write a VBL task to handle the setting of the CLUT?

___

The control call (SetEntries) in the video card is designed to turn off interrupts in the machine until it can sync to the VBL and then change the CLUT. It does this because otherwise the update to the CLUT will look really bad. You can call PBControl to make the SetEntries call (basically what the Color Manager is doing). There is a flag for making the control call Async, but it will not take care of the problem with waiting for the VBL. This limit is due to the way the video cards are designed. There may be some video cards which don't do this, but you probably don't want to be hardware dependent. If you are interested in using the PBControl call, then you may want to take a look at the Cards and Drivers manual.

In general, DTS doesn't recommend using SetEntries to animate the CLUT. It is not a very clean way to change the CLUT, and you don't really gain anything by using it over the Palette Manager. For more information about the two ways, take a look at the "Palette Manager Animation" article in issue #5 of *develop*.

Where to put video board gamma table & custom option menu code

Date Written: 7/30/91
Last reviewed: 8/1/91

Can I put both my custom option menu and my custom gamma table in the firmware of my color video board, or must I have an 'INIT'?

___

The gamma table information can be stored in the ROM of the video card; in fact, in most cases that is where you will want it. The monitor extension, however will need to be in the system folder. You can find a fair amount of information about making a monitor extension, including sample code, in Chapter 10 of *Inside Macintosh* Volume VI.

Graphics Devices Manager call support
Date Written: 9/24/91
Last reviewed: 9/24/91

Is SysEnvirons' hasColorQD a good way to determine whether I can make Graphic Device calls?

___

SysEnvirons' hasColorQD field is a good way to check whether the Graphics Device calls are available. Another good way is to see if Gestalt returns something ≥ gestalt8BitQD. You can tell if you have GWorlds (see the Graphics Devices chapter of *Inside Macintosh* Volume VI) if Gestalt returns something (> gestaltOriginalQD and < gestalt8BitQD) or ≥ gestalt32BitQD.

Recognizing a 32-bit address mode video card
Date Written: 2/13/91
Last reviewed: 2/13/91

How do I get the Macintosh system to recognize that I have a video card installed that only supports 32-bit addressing?

___

Two bits need to be set in the sRsrcFlags field: fOpenAtStart (bit 1) and f32BitMode (bit 2). If you do not have these bits set, SecondaryInit will not get called and QuickDraw will not realize that your card is a 32-bit address only card. These bits are documented in the second edition of Designing Cards and Drivers for the Macintosh Family on page 123.

Why Macintosh IIci and IIsi don't support direct video
Date Written: 1/28/92
Last reviewed: 2/13/92

What prevents the RBV (Ram Based Video) chip in the Macintosh IIsi from displaying 16-bit color on the new 12" Macintosh RGB monitor similar to the Macintosh LC?

___

The RBV chip is not able to supply data directly to its D/A converters; thus it bypasses the color lookup step completely. The Macintosh IIci and IIsi hardware is not capable of supporting direct video.

CLUT should be black for video card startup

Date Written: 1/25/91
Last reviewed: 2/8/91

On a system restart, our PrimaryInit routine clears our video card's memory before displaying grayscreen, but garbage appears on the Macintosh LC screen. What should we do to fix this problem?

___

The problem you are having with your video card on the Macintosh LC sounds like a CLUT problem. We recommend that you start by setting your entire CLUT to black then do what you are already doing. Basically the problem is caused by the CLUT DAC coming in early. If the CLUT is set then it should not cause you any problems.

XRefs:
Designing Cards and Drivers for the Macintosh Family, page 127

Video depth on Macintosh IIsi and LC Systems
Date Written: 1/8/91
Last reviewed: 1/30/91

Is there any way to get 24-bit color with a Macintosh IIsi or LC? For the Macintosh LC, is it possible to substitute a 1-MB Video RAM SIMM for the 512K VRAM SIMM, which substitutes for the 256K stock VRAM?

___

Neither the Macintosh IIsi nor the Macintosh LC is designed to support 24-bit video.

The on-board video hardware and the video driver on the Macintosh IIsi are only designed to support 1, 2, 4, and 8-bit video. Putting more memory in the SIMMs or any hacks you might come up with will not get 24-bit video on the on- board video hardware.

The on-board video hardware on the Macintosh LC is capable of supporting 16-bit video on the 12" monitor, but it is still limited to 8-bit video on the 13" monitor. If you try to put 1-MB of VRAM into the video RAM SIMM, you probably will not be able to boot the machine. At best the machine will only see 512K of the VRAM. So, on the Macintosh LC the video hardware and the video driver only support 1,2,4,8 bits on Apple's 13" monitor and 1,2,4,8,16 bits on Apple's 12" monitor.

Upgrade old Macintosh II ROMs to use with 8•24 GC card
Date Written: 12/13/90
Last reviewed: 1/16/91

Why does our Macintosh II with old ROMs not boot up with the 8•24 GC card, but it works fine on a Macintosh IIfx?

___

There is a problem with the really old Macintosh II ROMs that prevents them from working correctly with cards that have more than 1 MB of memory. The upgrade program is still available so take your machine to your friendly dealer and ask for a new board.

X-Ref:
“Test & board exchange for early Macintosh II/NuBus problem” Q&A

Macintosh 8•24 GC card 4-MB SIMM testing and pinouts
Date Written: 1/20/91
Last reviewed: 2/1/91

Does the current Macintosh 8•24 GC card work with 4-MB SIMMs? What are the pinouts for the higher density SIMM?

___

We have made 4-MB SIMMs for the 8•24 GC card, tested them, and they work just fine. The 'cdev' sees them and they work. One way to test your SIMMs is to do a memory test over NuBus on addresses $sD000000 to $sD3FFFFF (Bank A) and $sD400000 to $sD7FFFFF (Bank B).

The pinouts for the DRAM SIMMs are as follows:

D0-2	d16-33	a0-10	CAS0-8
D1-3	d17-34	a1-28	CAS1-26
D2-5	d18-36	a2-6	CAS2-39
D3-4	d19-35	a3-15	CAS3-57
D4-14	d20-45	a4-16	Vcc-18,37,47,56
D5-13	d21-44	a5-50	GND-1,19,25,38,48,55,59,64
D6-11	d22-42	a6-9	
d7-12	d23-43	a7-7	
d8-20	d24-51	a8-40	
d9-21	d25-52	a9-41	
d10-23	d26-54	RAS-46	
d11-22	d27-53	OE-27	
d12-32	d28-63	WE0-58	
d13-31	d29-62	WE1-49	
d14-29	d30-60	WE2-24	
d15-30	d31-61	WE3-17	

Macintosh video connector pinouts J26 and J2
Date Written: 11/29/90
Last reviewed: 2/8/91

What are the pinout and signal descriptions of the tiny little connector marked J26, located on the main logic board near the external video connector. Also, what is the part number and manufacturer of the mating connector?

___

You seem to be asking about the external connector (J26), but my guess is that you really want to know about the internal connector (J2), so I will tell you about both. J26 is the external connector and is documented in the Developer Notes for the Macintosh LC on page 13. The pin assignments are as follows:

1	Red video ground
2	Red video
3	Composite horizontal and video synch
4	Monitor ID bit 1

5	Green video
6	green video ground
7	monitor ID bit 2

8 not used
9 blue video
10 monitor ID bit 3
11 CSYNC ground
12 Vertical sync
13 blue video ground
14 not used
15 not user
16 shield ground

This connector is a standard DB-15 external connector.

Connector J2 is the internal connector that is documented in the Developer Notes on pages 37 and 38. The pinouts are as follows.:

1 ground
2 green video
3 ground
4 red video
5 ground
6 blue video
7 ground
8 composite blanking
9 ground
10 overlay signal
11 ground
12 composite horizontal and video sync signal

One source for the mating connector is Amp Incorporated, Harrisburg, PA 17105. The part number of the connector is 4-174904-2.

All modes must have same starting address for a frame buffer

Date Written: 11/27/90
Last reviewed: 1/30/91

Does Macintosh QuickDraw require that all modes have the same starting address for a frame buffer? In other words, I can’t have 1-bit mode start at offset 1024 and 8-bit mode start at offset 2048, right?

___

Yes, QuickDraw requires that all modes have the same starting address for a frame buffer.

When to put Mac configuration ROM at top of 16-MB slot space

Date Written: 11/27/90
Last reviewed: 2/8/91

Is it OK to put the configuration ROM on my board at the top of the 16-MB slot space?

___

As long as the card is installed on all Macintosh systems after the Rev A Macintosh II (the very first Macintosh II systems that were shipped), you will not have a problem with putting your configuration ROM at the top of the 16-MB space; in fact, we recommend this. To determine if the machine is a Rev A Macintosh II, you can use the tester application that is on the developer CD. Just run the application and if the machine is one of the old ones, a dialog box displays that tells the user to upgrade (for free) to a new motherboard.

Macintosh CAS2 instruction needs bus error trap handler
Date Written: 11/29/90
Last reviewed: 12/19/90

When we use the CAS2 instruction with the Apple's 8•24 GC card, we get a bus error. Why?

___

The reason you get a bus error has to do with the way the NuBus™ interface works on the Macintosh II. If an incoming NuBus cycle to the motherboard happens exactly at the same time a Read-Modify-Write (RMW) instruction such as CAS2 tries to go out over NuBus (something a card with bus master capability might do), the processor gets a bus error due to the bus conflict. What your software must do is set a bus error trap handler before the CAS2 instruction and eliminate the trap handler after the instruction. The trap handler needs only to do an RTE instruction to cause the CPU to retry the CAS2 instruction.

Don't access Macintosh RBV chip VIA emulation registers directly
Date Written: 11/12/90
Last reviewed: 12/19/90

Are the VIA emulation registers of the Macintosh RBV chip documented? If not, what are the offsets to the VIA emulation registers on the RBV chip? Is there more than one way to access the RBV registers?

___

The VIA emulation registers of the RBV chip are not documented because Apple does not want developers to use them. We have not supported the use of the VIA registers and we do not plan on supporting the registers of the various new ASICs that Apple has developed. If you decide to use these registers anyway, you are on your own. We don't know of another way to access the RBV registers.

Eliminating 1-bit per pixel mode
Date Written: 10/29/90
Last reviewed: 2/20/91

How do I declare 8- and 24-bit modes without having 1 bit-per-pixel (bbp) mode in my sResources?

___

The trick is that your first video mode (which, in your code, would be 8 bpp) must start with ID 128. The modes have to be sequential so the next mode (24 bpp in your case) would be ID 129.

Besides Designing Cards and Drivers for the Macintosh Family (2nd edition), there are some MacDraw drawings of configuration ROMs, one of which shows an old Apple video card (sometimes referred to as the “TFB” card). It is out of date, especially with regard to 32-Bit QuickDraw and multiple mode families, but it will give you an idea about how these things interact. A picture showing the exploded ROM and all these video sResource components may help when reading C&D. You can find these drawings on our developer CD and on AppleLink under the MacDTS folder of Developer Services (look for the hardware hints/slots folder).

The only other gotcha I can think of is whether or not you can be the startup device if you don’t have 1 bit per pixel. And I think this was due to older MacsBugs not being able to handle more than 1 bpp, which of course is fixed now.

Macintosh II video connector pinouts

Date Written: 11/17/89
Last reviewed: 12/17/90

What are the pinouts for the Macintosh II video card and IIci built-in video connector?

___

Here are the pinouts for the Apple 13" RGB and 12" Monochrome monitors for Macintosh II video cards and IIci built-in video:

Pin	Signal	Description	12" B/W 13" RBG	15" B & W
1	RED.GND	Red Video Ground	RED.GND	n/c
2	RED.VID	Red Video	RED.VID	n/c
3	CSYNC~	Composite Sync	CSYNC~	n/c
4	MON.ID1	Monitor ID, Bit 1	ID1.GND	n/c
5	GRN.VID	Green Video	GRN.VID	n/c
6	GRN.GND	Green Video Ground	GND.GND	n/c
7	MON.ID2	Monitor ID, Bit 2	n/c	ID2.GND
8	n/c		n/c	n/c
9	BLU.VID	Blue Video	BLU.VID	BLU.VID
10	MON.DI3	Monitor ID, Bit 3	n/c	ID3.GND
11	C&VSYNC.GND	CSYNC & VSYNC Ground	CSYNC.GND	VSYNC.GND
12	VSYNC~	Vertical Sync	n/c	VSYNC~
13	BLU.GND	Blue Video Ground	BLU.GND	BLU.GND
14	HSYNC.GND	HSYNC Ground	n/c	HSYNC.GND
15	HSYNC~	Horizontal Sync	n/c	HSYNC~
shell	CHASSIS.GND	Chassis Ground	CHASSIS.GND	CHASSIS.GND

Macintosh IIci video MON.ID information

‘0’ means that the line is grounded inside the monitor to provide information to the Macintosh IIci as to what kind of monitor is connected. If no monitor is connected to the IIci’s built-in video port, then the port is automatically disabled at startup.

ID3	ID2	ID1	Monitor Type
0	0	0	Unsupported
0	0	1	15" B&W Portrait Monitor (640x870)
0	1	0	RESERVED for use be Apple
0	1	1	Unsupported

```
 1    0    0    Unsupported
 1    0    1    RESERVED for use by Apple
 1    1    0    Mac II 12" B&W, 13" Hi-Res RGB (640x480)
 1    1    1    No external monitor connected (automatic)
 -------------------------------------------------------
```

Macintosh 8•24 GC card software

Date Written: 10/19/90
Last reviewed: 2/15/93

What's the current version of the 8•24 GC card software and which Macintosh models does it support?

___

The current 8•24 GC card software is "8•24 GC 7.0.1." It provides full 8•24 GC card support for all Macintosh models except the Macintosh Quadra. The 8•24 GC card works with the Macintosh Quadra as a video card only. You can turn your card in for a refund from your Apple dealer if you purchased an 8•24 GC card for a Macintosh Quadra and it's not useful to you as a video card.

By the way, Rev. B of the 8•24 GC card supports Apple's 16" monitor. If you own an Apple 16" monitor and a Rev. A 8•24 GC card, you can exchange your Rev. A card for a Rev. B card from your Apple dealer.