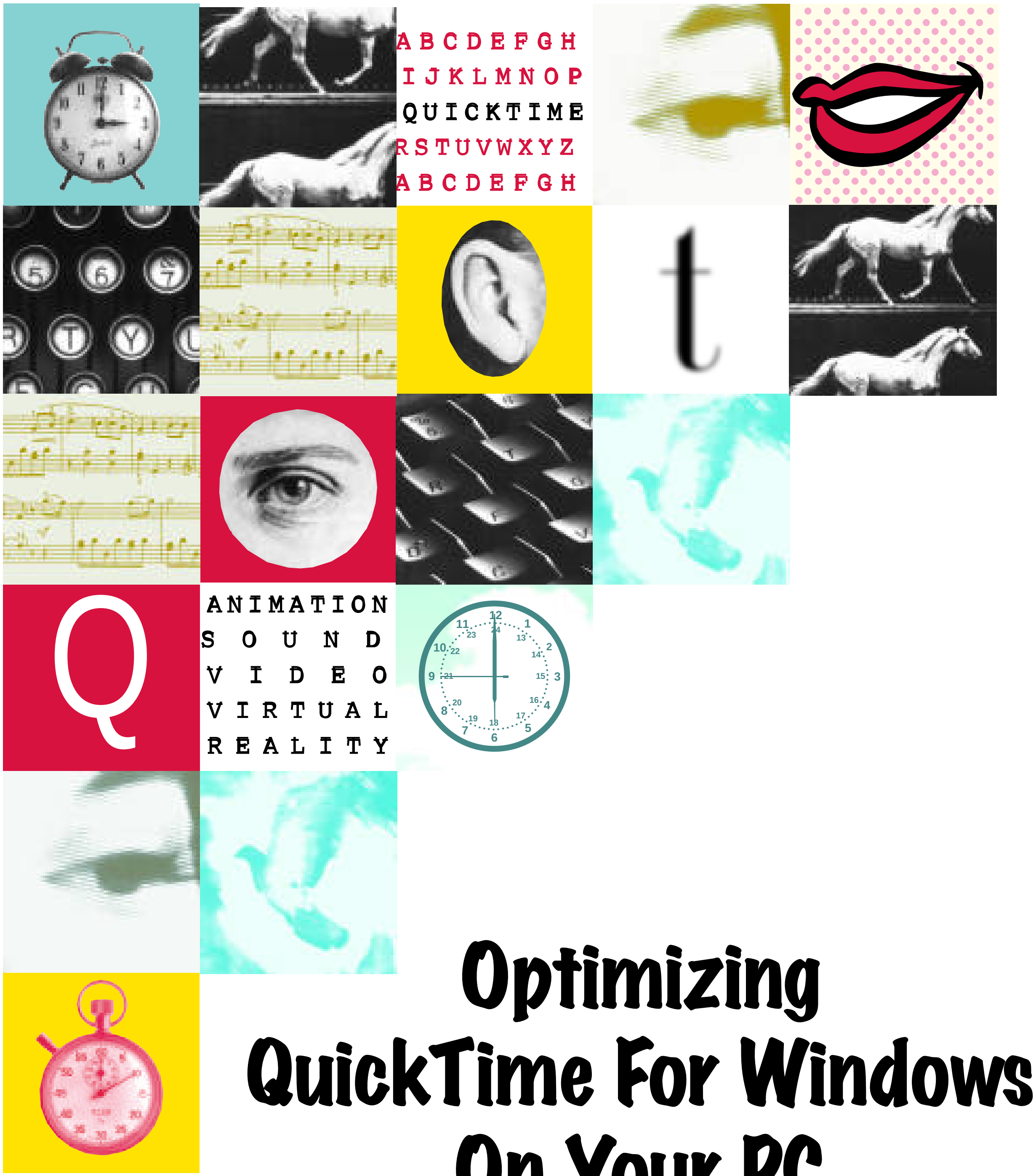


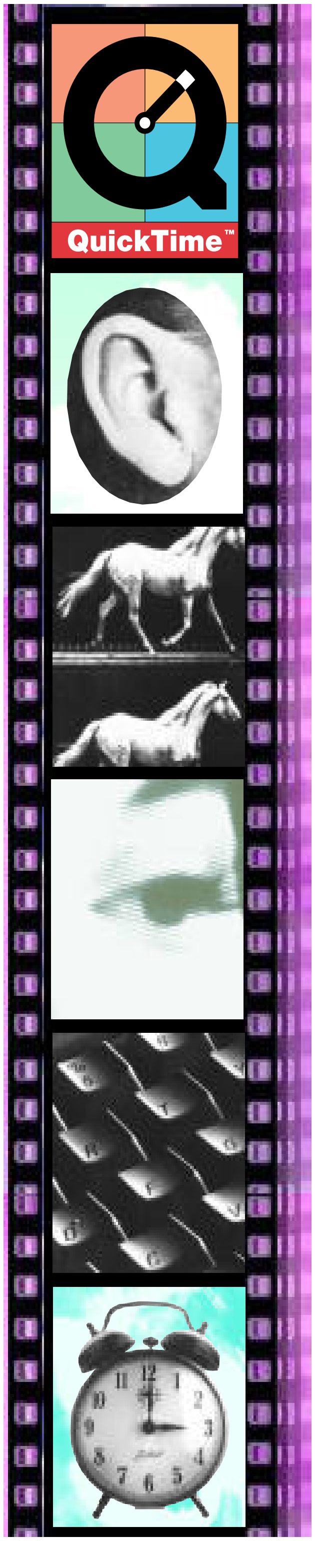
Optimizing QuickTime For Windows On Your PC

QuickTime™ Development Team
Apple Computer, Inc

QuickTime™ ***QuickFacts*** Number 1
December 1994

Table of Contents

What is QuickTime?



QuickTime is the container and the time-based delivery system for multimedia data types. QuickTime 2.0 has evolved into premier and most popular distribution mechanism for New Media. Unlike other multimedia delivery mechanisms that are limited to one or two data types, QuickTime lives up to the full potential of multimedia by allowing for the flexible handling of several different data types.

QuickTime 2.0, available for both Macintosh and Windows operating systems, supports two basic types of files: Movies and Pictures. Picture files contain still images. Typically these images are compressed. Movies contain time-based multimedia data. Although the most typical type of movie contains just Sound and Digital Video, more complex movies can also contain Animation, Music/ MIDI tracks, Text, and MPEG. A QuickTime movie can contain any or all of these data types. For example, a movie containing video and sound could be annotated with a text track that is synchronized along side with the other data. This movie might also contain a music track, which contains background music that can be turned on or off by the user. Other movies can contain multiple tracks of the same type of data. For example, a movie can contain multiple text track that are referenced upon user input. This powerful feature allows the user to interact with and control the information that is presented.

With its highly flexible architecture and tremendous ease of use, QuickTime provides you the power to deliver the full promise of multimedia to a generation of eager audiences.

Unlike most Macintoshes, PC’s underwent a considerable amount of evolution before it arrived upon the multimedia scene. In many ways, the Macintosh was born a multimedia machine. It had a built in sound card, optimized video, and support for plug and play CD-ROMS. PCs, on the other hand, grew into multimedia as a result of the development of individual components that had to get along with each other.

One major difference between a Mac and a PC is the way their components work together. A Mac is made of a proprietary architecture in which all components were developed to work with each other. A typical PC's vital organs can come from as many as 6 different vendors. Each of these components must be compatible with each other in order for the overall multimedia experience to be favorable. To throw more complexity to the mix, different components from different vendors may work completely different from each other. For example, a double speed CD-ROM from Brand X may have a data throughput of 270k per second, whereas a similarly advertised component from Brand Y may have a maximum throughput of only 220k per second. Both are double speed CD-ROMs, but they have different performance ratings.

As multimedia became popularized on the PC, standards were set up to establish a criteria for what was a multimedia machine, and what was not. MPC-1, the original multimedia standard, required at least a 386/33 CPU, an 8 bit sound card with speakers, and a single speed CD. This standard was changed in 1993 to the MPC-2 standard. This standard requires at least a 486/25 CPU, a double speed CD-ROM, local bus video, and a 16 bit sound card with speakers. Nearly all computers sold today fit the MPC-2 standard.

The first step in optimizing your PC to run QuickTime for Windows is to understand what hardware you have. Once you have determined what you have, you must understand the limits of that hardware and achieve performance that hits the ceiling of your hardware limits.

QuickTime offers some serious performance advantages over other multimedia playback options. These advantages allow multimedia developers better control over their content. QuickTime for Windows supports six different data types that may be used simultaneously. It also allows for faster, smoother performance of large sized movies. For example, a base level MPC-2 machine should be able to play a 320x240 pixel movie at 15 frames per second with minimal frame droppage. Other PC playback software solutions are limited to playing 240x180 pixel movies at between 10 to 12 frames per second. In order to bring you these performance advantages, QuickTime capitalizes on a unique architecture that departs from tradition operating system limitations.

DIRECT HARDWARE SUPPORT

One of the significant advantage QuickTime has over other playback standards is that it offers direct hardware support. Rather than draw video through the performance depriving GDI, QuickTime attempts to bypass GDI and write its video directly to the video card's frame buffers. This is especially effective because rather than having to write the video to the DRAM and then copy that information to the VRAM, the process is simplified to one step.

QuickTime is capable of bypassing GDI by being able to recognize the users video cards. Upon installation, QuickTime attempts to determine if your video hardware is recognized as one of the chipsets that may benefit from direct hardware support. A list of recognized graphic cards is located in Appendix A, at the back of this article. One of the keys in determining if your PC is performing optimally is to check the QTW.INI file, which is located in the Windows directory. To determine if your video card is one of the optimized chipsets, please read the section titled, "User Guide to the QTW.INI file."

TIMEBASED EXECUTION ENGINE

One of the principal advantages QuickTime for Windows has over competing products is that it is based on a timebased execution model. While other playback engines rely on synching sound and video on a frame by frame basis, QuickTime uses a timescale metaphor to synchronize the playback. The result is that QuickTime offers superior playback for movies of any duration, without the necessity of manual synchronization when adding data. Any data track can be synchronized automatically with other type. For example, if you have an existing movie, and you want to add a MIDI track to it, it is as simple as pasting it in, and reflattening the movie. The new data is automatically synchronized into the existing data.

Installing QuickTime for Windows is relatively quick and easy. However, in the past developers who have licensed QuickTime for Windows have implemented the installation in numerous different ways. QuickTime for Macintosh can be installed in only one manner; it had to be in the Extensions folder, or it did not load. Whenever you bought a new game, or other titles that used QuickTime for the Mac, the installer would check the Extensions folder and see if QuickTime was loaded. If it was, it would then check the version number to ensure that the latest version of QuickTime was going to be used. This model works well because the latest and greatest version of QuickTime was always used, and was always in the same central location. If you bought a CD-ROM a year ago, and used QuickTime 1.5, moving us to QuickTime 2.0 would yield better, smoother playback of movies.

QuickTime for Windows operates on a slightly different principal. QuickTime for Windows relies on a series of dynamically linked libraries (.dlls) and QuickTime components (.qtcs) in order to do its job. The way Windows works, these dlls can be almost anywhere on the user's hard disk and still work. They can even reside on the user's CD-ROM. In the past, title developers traditionally installed QuickTime for Windows one of two ways. One approach that was suggested by Apple in the past was to install all the .dll and .qtc files into a QTW directory, and then add a line to the Path statement of the user's Autoexec.bat statement so that when Windows needed those dlls, it would find them by virtue of the Path statement. The other approach that was used was to install the QuickTime files in the same directory as the executable file. This way, when QuickTime needed to be initialized, the proper dlls could be found since they resided in the same place as the executable. The problem with these approaches is that there could be numerous versions of QuickTime on the user's hard disk. Say for example, you purchased six titles that used QuickTime and they used a combination of the two approaches. The potential for having six copies of QuickTime on your hard disk was great. Furthermore, the title was locked into using the version of QuickTime it shipped with and was not able to take advantage of performance increases and bug fixes as time went along.

There is a solution to this problem. We said before that there were two approaches to installing QuickTime for Windows. The aim of both approaches was to ensure that when the QuickTime .dlls were called in order to be linked, they could be found. Windows uses the following hyarachcial search pattern to search for needed .dlls:

1. Current Directory/Working Directory
2. Windows Directory
3. Windows/System Directory
4. Module Path (where the executable resides)
5. Path Statement

As you can see, the first method for installing QuickTime for Windows was a fail-safe method for finding the needed .dlls because it was the last step in the chain. The problem with that method is that you must edit your autoexec.bat and the Path statement is limited to 128 characters. With the release of QuickTime for Windows 2.0, a simpler approach was taken. All the .dll and .qtc files were placed into the Windows/System directory. As you can see by the order chart, this method falls within the fail-safe bounds of the Windows search path without the need for modification of the Autoexec.bat file. Furthermore, all titles that used QuickTime were guaranteed to find the needed files there. There was no need for more than one copy of QuickTime on your machine. The QuickTime for Windows 2.0 installer searches your hard drive and attempts to remove all older versions of QuickTime. As was previously mentioned, QuickTime for Windows 2.0 is not only fully backwards compatible with all older version of QuickTime, it also provides a significant performance increase over previous versions. This is now the required method for installing QuickTime for Windows.

While this new approach to installing QuickTime for Windows greatly simplifies the installation process and ensures that the best version of QuickTime is always used, because of the method developers chose in previous versions of QuickTime for Windows, there is a scenario in which you can have some problems.

If you own a CD title that tries to run an older version of QuickTime directly off the CD-ROM, you may experience an error message called “call to undefined dynalink”, and program will fail to load.

The reason for this problem that the .dll files are being run straight from the CD. From a developer's point of view, this method of “install-less” execution of QuickTime for Windows is ill advised for three reasons. First, by having to load the .dlls from the CD, there is a high potential for having to do a “double-seek”. Instead of reading the movie datastream from the CD, the program is forced to stop reading data, and go back to the spot on the CD where the .dlls are located to get the execution code it needs on order to keep playback going. The second reason this meathod is discouraged is becuase there is no way to use a newer version of QuickTime. Since Windows found the correct .dlls on the CD, it won’t try to look on the hard disk. This locks the title with the version of QuickTime that it shipped with. Lastly, the reason for the “call to undefined dynalink” error is that although Windows does not continue looking for .dll files on the user’s hard disk, it does find other QuickTime components (.qtc files) These components are found by the QuickTime Component Manager and a linking of these files is attempted. The Component Manager that shipped with version 1.x did not allow for the incompatabilities of component files between version 1.x and newer QuickTime versions. The solution for this problem is to install QuickTime for Windows 2.01, which corrects this problem, by hiding new components from the old component manager.

A PC is comprised of several key elements that work together in providing the total multimedia experience. A bottleneck in one key playback area can undermine the entire performance of an otherwise fine playback machine. There are essentially three components that are critically important to playback performance. They are your video card, your data bus, and your CD-ROM drive. We will take a look at how these key components can limit your playback performance

VIDEO CARD AND DATA BUS

There are many different types of Video Cards available for consumer level PCs. Most currently available cards fall under the four different categories of PC data buses. They are ISA, EISA, VESA local bus, and PCI. PCI is a relatively new standard in PC video, but offers superior video data transfers. The two most popular types of cards are ISA and VESA Local Bus (VLB). Between the two, VLB is clearly superior in its ability to keep up with QuickTime's rapid transfer of data through the data bus. When ISA and EISA were a standard, they were not necessarily designed to handle the type of data throughput that multimedia application demanded. The Graphical User Interface's that became a standard on consumer level PC's in the early 90's put the ISA/EISA architecture to the test. As a general rule, playback rates are severely limited if you have an ISA or EISA video card. If you have VLB slots on your motherboard, upgrading to a VLB video card would remove a serious bottleneck in multimedia playback performance. Determining what type of card is in your system is relatively easy. If the major components, connectors, and chips on your card are on the left side of your card, and the connection to the bus is short, with a dense pinset, then the card is probably a PCI card. If the card has two closely positioned connectors, with the pinsets further apart, then the card is probably an ISA/EISA card. A VLB card is generally very long, with two connection points that are dense and on opposite ends of the card. Many of the current MPC-2 consumer level PCs come with on-board local bus video that is suitable for good QuickTime playback.

CD-ROMs

CD-ROMs are crucially important to multimedia playback. The first CD-ROMs that came to market were single speed drives that were capable of sustaining a maximum of 150k per second throughput, depending on the quality of the CD-ROM driver. The current standard for CD-ROMs is a double speed drive that is capable of sustaining a maximum of 300k per second. If you are playing a movie who's data rate exceeds the mechanical limits of your CD-ROM, then heavy frame dropping will occur. Before purchasing a CD-ROM title, always make sure that you check to see is it calls for a single or double speed CD-ROM.

INTRODUCTION

As was previously mentioned, QuickTime for Windows achieves its performance advantage by checking your hardware configuration, and if it recognizes the video card’s chipset as one that can be optimized, it attempts direct hardware access. As an end user, it is relatively easy to open your QTW.INI file and gets some important information from it. From the Program Manager, select “Run” from the File menu. In the command line, simply type, “QTW.INI”. Notepad should open up with your copy of the QTW.INI file. The top section contains most of the information we need. It may look something like this:

```
[QuickTime for Windows]
VHDW Component=Base Apple VHDW Component
Video Hardware=8-bit palettized pixels; S3 801 or 805 internal
hardware support
Implementation=Full

[Video]
PreferredComponent=C:\WINDOWS\SYSTEM\QTVHDW.QTC
```

Lets examine each line of the .INI file, and see what significance it has on the overall performance of QuickTime. The first line, `[QuickTime for Windows]`, is a section header whose information directly indicates QuickTime's hardware support options. The second line,`VHDW Component=Base Apple VHDW Component`, indicates that QuickTime is using a standard base level hardware support component in order to write directly to the hardware. The next line, `Video Hardware=8-bit palettized pixels; S3 801 or 805 internal hardware`, is very important in regards to performance. If this line has information in it, it means that QuickTime has recognized the video hardware. In this case, QuickTime has recognized this chipset as being either a S3 801 or 805 chipset. It will achieve the highest quality performance for the PC’s video bus. For example, if you have direct hardware support, but have an ISA card, you will get the maximum performance allowable for the data transfer bandwidth of an ISA card. If you have a supported VLB card, you should get the very best video playback performance. The next line of the file indicates that all QuickTime options are installed and available. This line should always say “Full”.

The next section is the Video section. The line directly under it will specifies the path to the component that QuickTime is using in order to write directly to the video frame buffers. Make sure never to delete this file, or QuickTime will have to resort to GDI level performance.

Now lets look at a file from a machine that does not offer direct hardware support.

```
[QuickTime for Windows]
Implementation=Full
VHDW Component=None
Video Hardware=Not directly supported
```

AS you can see, quite a bit of information is left out here. You will notice that no VHDW component is specified, and the Video Hardware section indicated that it is not directly supported. There is also no [Video] section in this document. Not having direct hardware support does not necessarily mean bad performance though. If you have a machine without direct hardware support, but you have a VLB card, you will not see optimal performance, but the high data transfer rate of the card should yield good performance. If however, you have an non-optimized chipset, and it is also and ISA card, you will notice some performance degradation due to the data bus bottleneck.

A section near the end of the file may also have some important significance. Let's take a look:

```
[Override]
oldVersion=38
```

This section lets you know that you have an older version of QuickTime on your drive. Generally, it's not good to have older versions of QuickTime around. The **qtold.dll** checks for older versions of QuickTime on your drive whenever QuickTime is initialized. If it finds one, it adds a digit to this counter. Whenever oldVersion gets to a multiple of ten, a warning message appears after QuickTime is initialized that advises the user of the older version. To determine if you have an older version floating around, set this number to one that ends in 9, and launch the Movie Player. If you got the warning dialogue, you have an older version on your drive. The best way to get rid of older versions of QuickTime is to run the latest QuickTime installer. It automatically searches for older version, and gives you the option of getting rid of them.

TROUBLESHOOTING

While the QTW.Ini file's main purpose is to record and track your machine's hardware specifications, there are some other uses for it that can help increase performance for machines experiencing playback problems.

As was previously mentioned, the Video section of the .ini file controls the preferred playback driver for the video data tracks. If you have supported hardware, the information in this section should have a path statement to the preferred QuickTime component that will be used to provide direct hardware access. If you don't have supported hardware, and you are experiencing playback problems with video, a simple edit in this section will help solve performance problems.

If you have a PCI video card, QuickTime 2.0 does not currently have supported drivers to optimize access to the hardware. If you have a PCI card, you may be experiencing some playback performance problems. Adding the following lines to the QTW.INI file may help alleviate those issues. Here are the lines to add.

You should not already have a [Video] section and should add these lines

```
[Video]
Optimize=Driver
```

That's it. Save and close the .ini file and the next time you launch QuickTime you should see some performance gains.

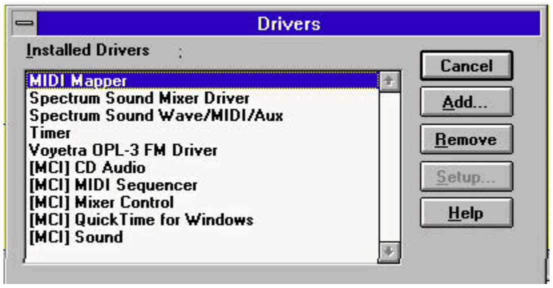
If you have some other video card and your performance is not what you would expect for the hardware, try the above solution. If that still does not do the job, try this modification to the file instead:

```
[Video]
Optimize=BMP
```

You should remember to save and close the file and relaunch QuickTime to examine the results.

Now that we know how QuickTime gains optimal performance through hardware support, the next step is to isolate any performance loss that may occur through the improper use of sound and video drivers. Drivers operate as a “middle man” between the hardware and the operating system. For example, a video card needs to be able to communicate with the operating system in order to ensure compatibility between the chipset on the card and the output that the operating system is expecting. This is achieved by way of the video drivers. Video Card manufacturers periodically update their drivers to take advantage their cards.

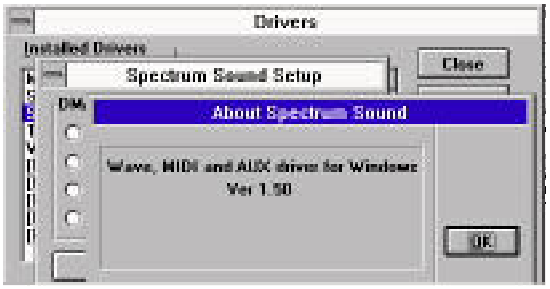
If you are experiencing performance problems with QuickTime, the first thing to check is that you have the latest version of your video and sound card’s drivers. In many cases, hardware manufacturers update their drivers to make them compatible with the latest multimedia technology. To check the version number of sound drivers, open the Control Panel. You should see an icon labeled drivers. Click on the drivers icon to open it. You should open a window that looks like this:


Drivers

Installed Drivers

MIDI Mapper
Spectrum Sound Mixer Driver
Spectrum Sound Wave/MIDI/Aux
Timer
Voyetra OPL-3 FM Driver
[MCI] CD Audio
[MCI] MIDI Sequencer
[MCI] Mixer Control
[MCI] QuickTime for Windows
[MCI] Sound

In this example, the sound drivers are Spectrum Sound drivers. If you double click onto each Spectrum Driver, you will enter a setup screen. Now click the about box. You may see something similar to this:


Drivers

Installed Drivers

Spectrum Sound Setup

About Spectrum Sound

Wave, MIDI and AUX driver for Windows
Ver 1.50

You should check the version number of the driver against the vendor's recommended latest version. If you need to update the driver, contact the vendor, or if you have an on-line service, check Compuserve for the latest driver software. In many cases, updating video and sound drivers may increase performance significantly

Here are some performance tips that will increase QuickTime for Windows playback performance.

CD-ROM CACHING

if you have software CD-ROM caching turned on, turning it off will greatly increase your performance. In QuickTime for Wiindows 2.0, we completely rewrote and optimized our internal data caching. External caching software, such as SmartDrive, which is build into DOS, will interfere with the playback of the datastream and result in less than optimal performance. You can turn off your CD cache in SmartDrive without having to turn off your disk cache. Here’s how you can check to see if you have SmartDrive working.

From the DOS prompt, type
Smartdrv

You should get sometinging like this:

```
Microsoft SMARTDrive Disk Cache version 5.0
Copyright 1991,1993 Microsoft Corp.

Cache Size:2,097,152 bytes
Cache Size while running Windows:     2,097,152 bytes

                  Disk Caching Status
Drive      read cache          write cache         buffering
—-------------------------------------------------------------
A:         yes                        no                          no
B:         yes                        no                          no
C:         yes                        no                          no
```

You have to check your Autoexec.bat file to see if SmartDrive has CD caching enabled or disabled. SmartDrive 5.0 has a very simple approach to turning of CD-ROM caching through a simple modification of the autoexec.bat file Here’s how:

1. Open you Autoexec.bat file and find the line that refers to SMARTDrive

in my case, my line says:
C:\DOS\SMARTDRV.EXE C+ 2048 512 /B:16
this means that SmartDrive is caching for my CD. Now I am going to turn it off.

2. Add the characters “/U” to the end of that line so that it reads

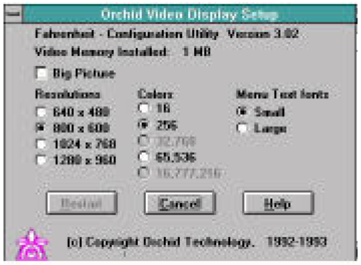
C:\DOS\SMARTDRV.EXE C+ 2048 512 /B:16 /U

3. Save the changes to your file and reboot. That should turn off caching for the CD-ROM drive.

VIDEO PERFORMANCE TIPS

If you have updated your video hardware drivers, and are still experiencing some playback problems, check the monitor depth that your card is set for. Typical screen resolutions are 640x480 pixels at 256 colors. This is called 8 bit mode. Video cards that have more memory (VRAM) allow you to select higher monitor depths and different screen resolutions. For example, if you have enough VRAM, you may be able to drive your monitor to at 32 thousand colors. This is called 16 bit mode. Finally, if memory permits, you may be able to drive your monitor so that is displays millions of colors. This is called 24 bit mode.

QuickTime plays movies best at the monitor depth that they were captured at. In most cases, this is either 8 bit or 24 bit mode. If you are seeing heavy frame droppage, and you have gone through all the other checks, a possible source of your playback problems could be due to your monitor depth being set at 16 bit mode. There are two approaches to changing your screen depth. Some card manufacturers, such as ATI and Orchid, provide a control panel file that changes the screen resolution without having to change your Windows setup. Here is an example of a control panel driven monitor depth changer:



Orchid Video Display Setup

Fahrenheit - Configuration Utility Version 3.02

Video Memory Installed: 1 MB

☐ Big Picture

Resolutions
○ 640 x 480
● 800 x 600
○ 1024 x 768
○ 1280 x 960

Colors
○ 16
● 256
○ 32,768
○ 65,536
○ 16,777,216

Menu Text fonts
● Small
○ Large

(c) Copyright Orchid Technology. 1992-1993

If you do not have such a control panel, then this change will have to be done with the Windows Setup program in the Main program group.

If you have tried all the fine tuning tips included in this document, and are still havingperformanceproblems, you can reach the QuickTime team on-line at:

quicktime@applelink.apple.com

Addendum 1

Direct Video Hardware Optimization Supported

Supported Video				
Adapter	**Driver**	**Depth**	**Resolution**	**Optimized?**
Standard VGA	Windows VGA	4	640x350	No
Standard EGA	Windows EGA	4	640x480	No
IBM 8514	Windows	8	1024x768	No
IBM XGA	07/27/92	4	640x480	No
IBM XGA	07/27/92	16	1024x768	No
ATI VGA XL	04/09/92	4	800x600	Yes
ATI VGA XL	04/10/92	8	1024x768	Yes
ATI VGA XL	04/20/92	16	800x600	Yes
ATI ULTRA	04/22/92	8	1024x768	No
ATI ULTRA	OEM	16	640x480	No
ATI Mach 32	01/05/93	8	640x480	Yes
ATI Mach 32	01/05/93	16	1024x768	Yes
Orchid IIs	03/01/92	8	800x600	Yes
Orchid IIs	03/01/92	16	800x600	Yes
Video 7 VRAM2 ERGO	03/10/92	8	800x600	Yes
Orchid Fahrenheit VA	02/19/93	8	640x480	Yes
Orchid Fahrenheit 1280	08/10/92	8	640x480	Yes
Orchid Fahrenheit 1280	08/10/92	16	800x600	Yes
Diamond Stealth	09/25/92	8	640x480	Yes
Diamond Stealth	09/25/92	16	640x480	Yes
Diamond SpeedStar 24x	04/28/92	8	1024x768	Yes
Diamond SpeedStar 24x	04/28/92	16	800x600	Yes
Diamond SpeedStar 24x	04/28/92	24	640x480	Yes

Addendum 2

Supported Audio Hardware

Adapter	Driver	Date
Creative Labs SoundBlaster Pro	SBPSND.DRV	05/15/92
Creative Labs SoundBlaster Pro	SBPSND.DRV	02/05/92
Creative Labs SoundBlaster Pro 16	SB16SND.DRV	04/14/93
Creative Labs ThunderBoard	SNDBLST2.DRV	03/10/92
Creative Labs ThunderBoard	SNDBLST2.DRV	05/13/92
Media Vision ProAudio Spectrum	MVPROAUD.DRV	02/03/93 1.3
Media Vision ProAudio Spectrum Plus	MVPROAUD.DRV	02/03/93 1.3
Media Vision ProAudio Spectrum 16	MVPROAUD.DRV	02/03/92
Media Vision Audio Port	MVAPORT.DRV	04/14/92
Microsoft Sound System	SNDSYS.DRV	09/21/92 1.0
Cardinal Technologies Sound Studio	TAPIGSS1.DRV	12/28/92
Orchid Sound Producer Pro.	PRODUCER.DRV	01/13/93
Orchid Sound Producer Pro.	PRODUCER.DRV	10/01/92
Turtle Beach MultiSound	MULTISND.DRV	08/27/92 1.1
ATI Stereo F/X	SFX.DRV	05/04/92

Addendum 3

Compatible Video Cards

Adapter	Driver	Depth	Resolution	Optimized?
Actix Graphics Engine Ultra Plus	03/25/93	16	1024x768	No
Actix Graphics Engine Ultra Plus	03/25/93	8	1024x768	No
Actix Graphics Engine Ultra Plus	03/25/93	16	800x600	No
Actix Graphics Engine Ultra Plus	03/25/93	8	800x600	No
Actix Graphics Engine Ultra Plus	03/26/93	24	640x480	No
Actix Graphics Engine Ultra Plus	03/25/93	16	640x480	No
Actix Graphics Engine Ultra Plus	03/25/93	8	640x480	No
ATI VGA XL	08/14/92 1.42	16	640x480	No
ATI VGA XL	04/20/92	8	800x600	Yes
ATI VGA XL	04/10/92	16	640x480	Yes
ATI VGA XL	04/20/92	8	640x480	Yes
ATI VGA XL	06/25/92	8	1024x768	Yes
ATI VGA XL	06/25/92	8	800x600	Yes
ATI VGA XL	07/28/92	16	640x480	Yes
ATI VGA XL	06/25/92	8	640x480	Yes
ATI VGAWONDER XL24	06/25/92	8	1024x768	Yes
ATI VGAWONDER XL24	07/28/92	16	800x600	Yes
ATI VGAWONDER XL24	06/25/92	8	800x600	Yes
ATI VGAWONDER XL24	07/28/92	16	640x480	Yes
ATI VGAWONDER XL24	06/25/92	8	640x480	Yes
ATI Graphics Ultra	08/14/92	4	800x600	No
ATI Graphics Ultra	08/14/92	4	640x480	No
ATI Graphics Ultra	06/25/92	8	640x480	No
ATI Graphics Ultra Pro	03/10/92	8	8514/a	No
ATI Graphics Ultra Pro	03/13/93	8	1280x1024	Yes
ATI Graphics Ultra Pro	03/13/93	16	1024x768	Yes
ATI Graphics Ultra Pro	03/13/93	8	1024x768	Yes
ATI Graphics Ultra Pro	03/13/93	24	800x600	No
ATI Graphics Ultra Pro	03/13/93	16	800x600	No
ATI Graphics Ultra Pro	03/13/93	8	800x600	Yes
ATI Graphics Ultra Pro	03/13/93	16	640x480	No
ATI Graphics Ultra Pro	03/19/93	24	640x480	No
ATI Graphics Ultra Pro	03/19/93	16	640x480	No
ATI Graphics Ultra Pro	03/19/93	8	640x480	Yes
ATI Graphics/Pro	11/27/92	8	1024x768	No
ATI Graphics/Pro	11/27/92	16	640x480	No
Dell 466/M S3	09/04/92(1.2)	4	1280x1024	Yes

Dell 466/M S3	09/04/92(1.2)	8	1024x768	Yes
Dell 466/M S3	09/04/92(1.2)	8	800x600	Yes
Dell 466/M S3	09/04/92(1.2)	4	800x600	Yes
Dell 466/M S3	09/04/92(1.2)	15	640x480	Yes
Dell 466/M S3	09/04/92(1.2)	8	640x480	Yes
Diamond Stealth Pro	12/07/92	8	1280x968	No
Diamond Stealth Pro	12/07/92	8	1280x1024	No
Diamond Stealth Pro	12/16/92	16	1024x768	No
Diamond Stealth Pro	01/06/93	8	1024x768	No
Diamond Stealth Pro	12/18/92	16	800x600	No
Diamond Stealth Pro	01/06/93	8	800x600	No
Diamond Stealth Pro	02/23/93	24	640x480	No
Diamond Stealth Pro	01/06/93	16	640x480	No
Diamond Stealth Pro	01/06/93	8	640x480	No
Diamond SpeedStar 24	04/14/92	8	1024x768	Yes
Diamond SpeedStar 24	04/14/92	15	800x600	No
Diamond SpeedStar 24	04/14/92	8	800x600	Yes
Diamond SpeedStar 24	04/14/92	15	640x480	No
Diamond SpeedStar 24	04/14/92	8	640x480	Yes
Diamond Viper VLB	04/27/93	8	1152x900	Yes
Diamond Viper VLB	04/27/93	16	1024x768	Yes
Diamond Viper VLB	04/27/93	8	1024x768	Yes
Diamond Viper VLB	04/27/93	24	800x600	No
Diamond Viper VLB	04/27/93	16	800x600	Yes
Diamond Viper VLB	04/27/93	8	800x600	Yes
Diamond Viper VLB	04/27/93	24	640x480	No
Diamond Viper VLB	04/27/93	16	640x480	Yes
Diamond Viper VLB	04/27/93	8	640x480	Yes
Genoa Windows VGA 8500	02/16/93	8	1024x768	Yes
Genoa Windows VGA 8500	10/23/92(turbo)	16	800x600	Yes
Genoa Windows VGA 8500	08/24/92	16	800x600	Yes
Genoa Windows VGA 8500	10/28/92	8	800x600	Yes
Genoa Windows VGA 8500	12/01/92(turbo)	24	640x480	No
Genoa Windows VGA 8500	10/14/92	24	640x480	No
Genoa Windows VGA 8500	11/11/92(turbo)	16	640x480	Yes
Genoa Windows VGA 8500	08/24/92	16	640x480	Yes
Genoa Windows VGA 8500	10/28/92	8	640x480	Yes
IBM XGA2	07/27/92	8	640x480	No
IBM XGA2	07/27/92	16	640x480	No
IBM XGA2	07/27/92	8	800x600	No
IBM XGA2	07/27/92	16	800x600	No
Orchid Fahrenheit VA	02/19/93 10:00	16	1024x768	Yes
Orchid Fahrenheit VA	02/19/93 10:00	24	640x480	Yes
Orchird VLB (Local Bus)	02/19/93	16	800x600	Yes
Orchird VLB (Local Bus)	02/19/93	15	800x600	Yes
Orchird VLB (Local Bus)	02/19/93	8	800x600	Yes
Orchird VLB (Local Bus)	02/19/93	24	640x480	No
Orchird VLB (Local Bus)	02/19/93	16	640x480	Yes

Orchird VLB (Local Bus)	02/19/93	15	640x480	Yes
Orchird VLB (Local Bus)	02/19/93	8	640x480	Yes
Orchid Fahrenheit 1280	08/10/92 10:00	15	800x600	No
Orchid Fahrenheit 1280	08/10/92 10:00	8	800x600	Yes
Orchid Fahrenheit 1280	08/10/92 10:00	16	640x480	No
Orchid Fahrenheit 1280	08/10/92 10:00	15	640x480	No
Orchid Pro designer IIs/D 1.1	04/7/92	15	800x600	No
Orchid Pro designer IIs/D 1.1	04/7/92	8	800x600	Yes
Orchid Pro designer IIs/D 1.1	04/7/92	8	640x480	Yes
Orchid Pro II	03/01/92	8	1024x768	Yes
Orchid Pro II	03/01/92	15	800x600	No
Orchid Pro II	03/01/92	8	800x600	Yes
Orchid Pro II	03/01/92	15	640x480	No
Orchid Pro II	03/01/92	8	1024x768	Yes
Sigma Legend GX	04/01/92(1.13)	8	1024/768	No
Sigma Legend GX	04/01/92(1.13)	8	800x600	No
Sigma Legend GX	03/31/92(1.13)	8	640x480	No
Sigma Legend GX	04/01/92(1.13)	8	640x480	No
Sigma Legend 24X	11/13/92	8	1024x768	No
Sigma Legend 24X	11/13/92	16	800x600	No
Sigma Legend 24X	11/13/92	8	800x600	No
Sigma Legend 24X	11/14/92	24	640x480	No
Sigma Legend 24X	11/13/92	16	640x480	No
Sigma Legend 24X	11/13/92	8	640x480	No
Matrox 1024	08/21/92 2.07	24	1024 x 768	No
MiroCrystal 8S	04/13/93	8	1024x768	No
MiroCrystal 8S	04/13/93	16	800x600	No
MiroCrystal 8S	04/13/93	8	800x600	Yes
MiroCrystal 8S	04/15/93	16	640x480	No
MiroCrystal 8S	04/13/93	8	640x480	Yes

Addendum 4

Compatible Sound Cards

Adapter	Driver	Date
Creative LabsThunderBoard	SNDBLST2.DRV	03/10/92
Creative LabsThunderBoard	SNDBLST2.DRV	05/13/92
Creative LabsSound Blaster 2.0		02/16/93
Digispeech PortAble Sound Plus	PRTSND.DRV	04/14/93
IBMM/Audio	ACPA.DRV	10/29/92 11:38
IBMM/Audio	ACPA.DRV	8/28/92
MediaVisionThunder and Lightning	TLWAVE.DRV	08/25/92