



# QuickTime<sup>TM</sup> Technology Brief

## QuickTime<sup>TM</sup> 3.0

April 5, 1997

© 1997, Apple Computer, Inc. All rights reserved.

## **Introduction**

QuickTime is Apple Computer's award-winning, industry-standard, software architecture that makes it possible to create, integrate, and publish all types of digital media. Using QuickTime, software applications can offer the ability to work with a wide variety of media file formats and media encodings in an easy, consistent way. QuickTime was designed from the ground up to simplify the task of working with and integrating the widest possible range of digital media types—not just sound and video. QuickTime today truly provides the most comprehensive, most flexible, and most integrated set of media services available to content creators and software developers regardless of whether you publish to video tape, CD-ROM, or the Internet.

QuickTime is composed of three distinct elements—the QuickTime Movie file format, the QuickTime Media Abstraction Layer, and a rich set of built-in QuickTime media services. These three elements enable users to realize the full benefits of the QuickTime software architecture.

The QuickTime Movie file format specifies a standard means of storing digital media compositions. Using this powerful container format, it is possible to store not only individual media assets, such as video frames or audio samples, but also to store a complete description of the overall media composition. This might include a description of the spatial and auditory relationships between multiple video and audio channels in a more complex composition.

QuickTime also specifies a comprehensive software architecture known as the QuickTime Media Abstraction Layer. This abstraction layer specifies how software tools and applications access the rich set of media support services built into QuickTime. It also specifies how hardware can accelerate performance critical portions of the QuickTime system. Finally, the QuickTime Media Abstraction Layer outlines the means by which component software developers can extend and enhance the media services accessible through QuickTime.

If Apple had stopped here, QuickTime would be the most powerful and flexible digital media architecture on the market today. However in order for software developers and customers to take full advantage of the architecture, it was recognized that high-quality software implementations of QuickTime's many media services would be essential. Therefore, QuickTime includes a comprehensive set of built-in capabilities that software developers can use as a foundation for their tools. In practical terms, this means the time and resources required to create QuickTime-based software applications are dramatically reduced when compared to traditional turnkey solutions. Furthermore, because QuickTime provides a wide variety of useful and ready-to-use media services in a totally integrated environment, QuickTime-based applications are able to provide a richer, more powerful range of features to customers than would otherwise be possible. These features provide software developers the sorts of advantages that can mean the difference between success or failure in the marketplace. And of course, customers have long enjoyed the benefits of rich interoperability across the wide range of QuickTime-based tools available in the market today.

## **QuickTime Movie File Format**

As previously mentioned, QuickTime specifies its own file format called a QuickTime Movie. This file format is an extremely flexible container format for virtually any type of digital media asset. While QuickTime in no way *requires* you to store your media in QuickTime Movie files, most customers find that the QuickTime Movie file format is among the most convenient and powerful formats for storing common digital media types such as audio and video. Furthermore, QuickTime Movie files are platform neutral, open, and extensible. These characteristics have convinced many major vendors to commit to the QuickTime Movie format. For example, Silicon Graphics has chosen the QuickTime Movie file as the standard media container format on their platforms. QuickTime Movies are well supported on the Windows platform as well, with QuickTime-enabled software available from Apple, Microsoft, Macromedia, Adobe, and numerous other developers. Because of this ability to work with QuickTime Movie files on any computing platform, the QuickTime Movie format continues to grow in popularity. Because a QuickTime Movie file can be used on any user's computer with most video editing software, it provides an unrivaled ability for users to share these files in collaborative work environments made up of a diverse collection of media editing and capture stations.

The QuickTime Movie format isn't limited to use in a single market segment. It is an efficient format for use in many different markets. For example, Scitex Digital uses QuickTime Movie files for storing digital video for their professional VideoSphere editing station. QuickTime Movie files have long been the most popular format for delivering video on personal computer CD-ROMs. More recently, the QuickTime Movie format has become the most widely adopted format for publishing digital video on the Internet.

## **Support for Other Media File Formats**

While the QuickTime Movie format has many advantages as a container for digital media, there are many other digital media file formats that are in common use. Rather than requiring video editing applications to add explicit support for each of these media filetypes, QuickTime provides transparent access to the most popular digital media formats. If an application uses the QuickTime Media Abstraction Layer to access and manipulate digital media, it will be able to work with a wide variety of digital media formats, including QuickTime Movies, with no additional work. By providing this powerful capability, QuickTime allows all video editing applications to support a broad range of media formats with relatively little effort.

Because digital media files are often very large, it isn't practical to convert these files into a QuickTime Movie file by duplicating the media from the original file into a QuickTime Movie. This process would not only be time consuming, but would require a significant increase in storage space. Because QuickTime is able to work with most media file formats directly in place, there is no need to waste time or storage space duplicating the media into a QuickTime movie file. Further, in any translation process there is the risk of losing information, nor is any information lost since the original file is always still available.

QuickTime contains support for over a dozen popular digital media file formats. More importantly, the QuickTime architecture defines how any developer can extend QuickTime so that it supports additional digital media file formats. This capability means that new file format support can be added to QuickTime at any time, and that it will work seamlessly with all existing QuickTime enabled applications. This extensibility helps to ensure that QuickTime applications are well positioned to take advantage of new trends in the fast changing world of digital media.

## **Digital Video File Formats**

On the Windows platform, Microsoft has long promoted the AVI (Audio/Video Interleaved) file format for the storage of digital media. As a result, many Windows users have digital video stored in AVI format. QuickTime 3.0 has the ability to work with these AVI files. This capability means that Windows users can step up to the power and flexibility of the QuickTime architecture without having to leave their existing media content behind.

Because the AVI file format lacked many of the features required for use in professional video editing environments, an industry group created the OpenDML file format standard which extends the AVI format to make it more useful for the professional market. QuickTime 3.0 is able to work with OpenDML files, providing support for the additional features defined in this format. Because of the ease of adding new file format support to the QuickTime architecture, QuickTime was among the first software technologies to publicly demonstrate support for the OpenDML standard.

To enable media interchange in the high end of the digital video market, Avid created the Open Media Framework (OMF) file format. The OMF format is used by many Avid products, as well as products from other developers working in that market. The OMF format is a relatively complex format, and is therefore not often supported by video editing applications in the desktop/PC markets. Apple has developed a software extension to QuickTime which enables direct access to OMF media files from QuickTime enabled applications. The availability of this QuickTime extension means that users of OMF media now have fast and easy access to the broad range of QuickTime enabled tools.

The MPEG video standard was developed to provide a format for delivering digital media to the consumer market. Because MPEG provides relatively high quality sound and video and relatively low data rates, it has been successful in some markets. QuickTime provides direct access to MPEG-1 audio and video.

Recently, the first generation of digital video cameras have appeared. Manufacturers such as Sony and Panasonic are shipping these high quality, low cost video cameras under the name DV-Cam or DVC-Pro. Because these cameras use digital storage and digital transfer technology, there is no generation quality loss when the media is edited or copied. And because of the high quality images and ease of use features, DVC is being rapidly accepted in the professional video market, and is expected to succeed in the consumer space in the future. All DVC cameras use the same format for storing digital sound and video. QuickTime 3.0 is able to directly work with the digital media streams from these cameras. This means that QuickTime applications are immediately enabled to work with DVC media, including the ability to losslessly edit together different DVC clips.

## **Audio File Formats**

One of the most common formats for storing digital audio, is the AIFF (Audio Interchange File Format) file created by Apple Computer. It provides the ability to store almost any format of digital audio, including multi-channel sound and very high resolution sound. Because of these abilities, it is a popular format for storing audio in the professional audio market, as well as the consumer and Internet markets. QuickTime provides direct access to digital audio stored in AIFF files.

Microsoft also provides a file format for storing digital audio, called a Wave file. As a result of Microsoft's support, this format is common on the Windows platform, primarily in the consumer market. It is also a popular format for audio on the Internet. QuickTime provides direct access to most digital audio stored in most Wave audio files.

In the high end digital audio market on the Macintosh, Sound Designer II files have become very common. While the Sound Designer II format isn't particularly flexible, its relative simplicity has made it a very widely supported standard. QuickTime 3.0 provides direct access to all Sound Designer II audio files.

One of the most common audio files on the Internet is a format developed by Sun called "AU". This format has similar capabilities as Microsoft's Wave format, and is most commonly used for relatively low quality audio. QuickTime directly supports AU audio files, ensuring that audio files from the Internet can be easily accessed by any application.

Another common audio file format on the Internet is MPEG layer 2 audio. MPEG audio provides significantly better audio quality in the same data size used by Wave and Au files. QuickTime provides direct access to MPEG layer 2 audio streams.

### **Animation Formats**

On the Macintosh platform, animations are most commonly stored directly in QuickTime Movie files. On the Windows platform, one of the most popular formats for storing animation is the Flic file format. Flic files are a compact and efficient way to store short animations. While originally created for use with the AutoDesk Animator application, because of the relative ease of supported Flic files, they have become a common format for distributing computer generated animations. QuickTime 3.0 provides direct access to Flic files, enabling applications to easily perform operations like editing and chroma keying that are otherwise difficult with Flic files.

### **Still Image Formats**

While QuickTime is best known as software for working with dynamic media types such as video and sound, QuickTime also provides extensive support for working with still images. This capability is extremely important to applications working with dynamic media, because it is very common to combine still images with video. QuickTime provides the same features in its still image support, as in its dynamic media support. That means that image files are accessed directly in place with no transcoding or data duplication, that support for a rich set of standard file formats is built in to QuickTime, and that developers can easily add support for new image file formats.

One of the most powerful tools for digital image manipulation is Adobe Photoshop. As a result, Adobe Photoshop files are one of the most common contains for still images. QuickTime supports direct access to Adobe Photoshop files, including display of the layers created by Photoshop version 3.0. QuickTime also provides access to the alpha channel stored in some Photoshop files, making it easy to composite Photoshop files on top of digital video.

On the Macintosh platform, QuickDraw pictures have always been a popular image format because of their great power and flexibility, and the built-in operating system support. QuickTime provides direct access to QuickDraw Picture files, ensuring consistent access to this popular format.

On the Windows platform, Bitmap (BMP) files have been popular because of their simplicity and built-in operating system support. QuickTime 3.0 enables applications directly access to BMP images.

On the Internet, GIF and JPEG (JFIF) image files are used on almost every web page. Because of the incredible pervasiveness of the Internet, there is a vast library of images stored in these formats. By providing direct access to both GIF and JPEG images, QuickTime makes it easy for any application to use these pervasive file formats.

The Silicon Graphics platform has long been popular as an environment for creating and working with extremely high resolution digital images. Many of the images created on the SGI platform are stored in the Silicon Graphics image format. QuickTime 3.0 provides direct access to these image files, including the alpha channel compositing information stored in some Silicon Graphics image files. This capability ensures that QuickTime users will be able to work with images created on the Silicon Graphics platform.

### **Media Formats Summary**

QuickTime's software architecture ensures that applications have direct access to a broad range of digital media formats. By providing the support for these media formats in QuickTime itself, rather than requiring each application vendor to provide this support, QuickTime ensures consistent access to all digital media. This means that users don't need to worry about the format that their media is stored in - QuickTime ensures that the media will be accessible. And because QuickTime is extensible to support new media types, compression formats, and digital media file formats, QuickTime will continue to be able to support new digital media formats as they emerge. QuickTime's broad support for most popular digital media formats makes it the best choice for building software applications today, and its well proven extensibility ensures that it will remain the best choice tomorrow.

Unlike many other media architectures, QuickTime has been designed to support continuity with other file formats and standards. QuickTime doesn't obsolete existing data formats, so you can continue to use your existing content as you switch over to or start using a QuickTime-based solution. QuickTime does not demand that legacy data be abandoned or that time-consuming and expensive conversions be performed before your non-QuickTime data can be used.

## QuickTime Media Abstraction Layer

The key to QuickTime's flexibility and performance is the QuickTime Media Abstraction Layer. This foundation technology is an advanced, component-based software architecture that provides software and hardware developers full access to the built-in QuickTime services and also specifies how to accelerate and extend QuickTime's capabilities through a powerful plug-in framework.

The QuickTime Media Abstraction Layer specifies a comprehensive set of services covering virtually all aspects of digital media creation, editing, and playback. An abbreviated listing of some of these services includes:

- Timing and synchronization
- Audio and image data compression and decompression
- Image blitting, format conversion, scaling, composition, and transcoding
- Audio mixing, sample rate conversion, and format conversion
- Audio and video effects and transitions
- Synchronized storage read and write
- Media capture
- Media import and export
- Standard user interface elements, such as movie controllers, media previewers, and media capture dialogs

One of the most compelling features of the QuickTime Media Abstraction Layer is the broad range of media types supported. QuickTime 3.0 includes built-in support for 10 different media types (video, audio, text, timecode, music/MIDI, sprite/animation, tween, MPEG, VR, 3D). For each of the built-in media types, QuickTime provides a rich set of media-specific services appropriate for managing each particular media type. QuickTime supports media types that are appropriate for all types of digital media publishing, whether the intended target is tape, CD-ROM, or the Internet.

The QuickTime Media Abstraction Layer provides developers a high degree of isolation from underlying media hardware services. The abstraction of low-level hardware services such as clocks and timers, storage devices, video displays and frame buffers, audio mixers, music synthesizers, media capture devices, and graphics chips means that QuickTime-based applications are easier to develop and maintain. In most cases, developers need not be concerned with the details of how QuickTime media interacts with the underlying hardware. When you consider the huge diversity of hardware configurations in the market today, it is reassuring to know that the QuickTime Media Abstraction Layer protects the developer from having to understand and manage the details of all this hardware complexity.

Additionally, the QuickTime Media Abstraction Layer ensures that QuickTime-based software applications will be able to take advantage of new hardware improvements without requiring software changes. For hardware developers, this abstraction layer ensures that hardware advancements can be introduced into the marketplace with the confidence that existing software applications will be compatible with the new hardware. For customers, these two advantages mean that QuickTime-based solutions will be more compatible and that they will more easily be able to take advantage of technology advancements as they become available. QuickTime-based solutions are better able to keep pace with rapid innovations in digital media technology thanks to the QuickTime Media Abstraction Layer.

An essential component of high-performance digital multimedia has always been hardware acceleration. While great advances in processor speeds have enabled many media processing tasks to be adequately handled entirely in software, evolving customer expectations and needs continue to drive demand for new acceleration features. The QuickTime Media Abstraction Layer provides a rich specification for how hardware can be used to accelerate performance in many areas of QuickTime. This specification shows how high-speed image codec chips, 2-D and 3-D graphics acceleration chips, audio processors, music synthesizers, video effects processors, and specialized storage devices can be integrated into QuickTime to achieve optimal performance. And of course, QuickTime-based applications are automatically able to take advantage of these accelerated features without requiring modifications to their software. Hardware developers who adopt QuickTime often find that they no longer need to invest as heavily in developer evangelism programs to ensure that their cards and chips are supported by a broad range of tools. By making your hardware QuickTime-compatible, you automatically are compatible with the industry's leading digital media applications, tools, and titles.

As shown above, the QuickTime Media Abstraction Layer provides a very rich and very broad set of multimedia services designed to satisfy the needs of most developers and customers. However, Apple recognizes that innovation is at the heart of the digital media industry. QuickTime's component-based architecture was designed from the beginning to be extensible. QuickTime 3.0 consists of over 175 different software components. These software components are divided into one of over 20 different categories, with the components in each category providing a specific service. This component-based architecture allows QuickTime to be continuously updated to support new technologies by adding new components or by enhancing existing ones. Developers, too, can add their own software components, allowing their technology to take advantage of all the benefits of QuickTime integration.



## **QuickTime Dynamic Media Types**

In the remainder of this overview of the QuickTime 3.0 architecture, several specific media types supported in QuickTime will be explored in greater detail. While not entirely comprehensive, each of the examples below serves to illustrate some of the scope and depth of QuickTime's media capabilities.

### **Video**

Working with digital video can be extremely confusing. There are dozens of different compression formats to choose between. There are both hardware-based and software-based playback engines. There are several different file formats used by different markets. And most tools for working with video only support one or two file formats, and a few different compression formats. Some tools may not be able to take advantage of hardware acceleration. Fortunately, QuickTime provides solutions to these problems. By building digital media tools on top of QuickTime, tool vendors can solve all these problems.

QuickTime's media abstraction layer provides the ability to work with any file format and any compression format. QuickTime provides support for working with popular digital video file formats including QuickTime Movies, OMF, AVI, and OpenDML. Tools created to work with QuickTime are immediately enabled to work with all these file formats.

Similarly, QuickTime provides the ability to work with nearly any video compression format through its Image Compression Manager. The Image Compression Manager provides a standard way for tools to manipulate any kind of compressed video, ensuring that all QuickTime based tools can support all compression formats in use today, as well as new formats that will emerge. The Image Compression Manager makes it just as easy to work with uncompressed video, ensuring that the highest possible quality video can be manipulated by QuickTime tools. The Image Compression Manager also provides a hardware abstraction layer which isolates tools from the underlying implementation of each compression format. This means that users can install new video hardware and existing QuickTime based tools will be able to immediately take advantage of the new hardware.

The QuickTime Movie file format doesn't impose any restrictions on the characteristics of video that may be stored in a QuickTime Movie file. For example, video can be stored at any frame rate, not just those frame rates which are common in the broadcast industry. While this ensures that QuickTime Movies can easily be used to store NTSC, PAL, and film frame rate video, it also means that QuickTime Movies can be used for MPEG, CD-ROM, and Internet video. Furthermore, because there is no upper limit on the frame rate of a QuickTime movie, it is an ideal format for high frame rate animations and it is ready to support future digital video standards which may enable even higher frame rates.

The QuickTime Movie file format is also able to store any number of video channels. This capability is essential for supporting real-time rendering of transitions and special effects. QuickTime 3.0's play back engine is able to play several different channels of video simultaneously, providing these streams of video to dedicated hardware which decompresses and combines these video channels in real-time using a variety of transitions.

In addition to providing support for playing back digital video, QuickTime also provides a set of tools for manipulating digital video. Because most professional video work is field based, QuickTime contains a set of services which make it easy for tools to independently access each field in a video stream, and easily reorder or duplicate fields to provide effects such as backwards playback or slow motion. QuickTime is also able to combine video images in a variety of ways including cross-fades and alpha channel based compositions.

Because there are so many different Motion JPEG formats in common use, Apple Computer worked with a group of digital video companies to standardize how Motion JPEG is stored. QuickTime contains support for these standard Motion JPEG formats as well as support for a number of other Motion JPEG formats, such as AVR (Avid Video Resolution). To make it easy to move between different kinds of Motion JPEG, QuickTime contains a number of transcoders which allow video to be converted from one Motion JPEG format to another without introducing any recompression artifacts.

## **MPEG**

MPEG format media has become an important data type in a variety of established and emerging digital media markets. MPEG-1 content is commonly employed in VideoCD and CD-ROM titles because of its relatively good image quality and is used widely on the Internet due to its comparatively low bit-rate requirements.

Responding to increased interest in this industry-standard media format, Apple added support for MPEG-1 to the QuickTime architecture in 1994. MPEG-1 playback has traditionally required special purpose hardware to provide high-quality playback performance. Indeed, the original QuickTime implementation relied upon dedicated hardware to decode MPEG-1 streams.

With the advances in processor technologies, however, it has now become increasingly feasible to decode these streams in software without the need for dedicated decoder hardware. And using the latest version of the QuickTime MPEG Extension, QuickTime customers are now able to take full advantage of MPEG-1 without any special purpose hardware.

The integration of MPEG-1 into QuickTime provides an excellent study in how QuickTime adds value to existing industry-standards.

For all the benefits of MPEG-1 as an audio and video format (VHS-quality images and high-quality audio at single-speed CD-ROM data rates), MPEG-1 has many characteristics that have made its use in PC-based interactive multimedia applications relatively limited. With QuickTime, many of these limitations can be overcome.

The first challenge in making MPEG content easily accessible is to cope with the wide range of encoding conventions utilized in different manufacturers encoders. The Internet contains a great deal of MPEG content that is only partially compliant with the MPEG-1 specification. Likewise, early MPEG encoders often generated MPEG streams that were playable only when using specific hardware decoders. QuickTime's MPEG support is very robust, enabling it to work with a very wide variety of MPEG-1 content.

Due to the complex compression scheme used for MPEG-1 media, basic media transport operations such as frame-accurate positioning, fast forward and reverse playback have usually not been supported in other MPEG-1 systems. When integrated into QuickTime, however, MPEG-1 inherits these standard capabilities associated with all QuickTime dynamic media.

Basic editing operations such as copy and paste on MPEG-1 streams have generally been relegated to high-end MPEG encoding workstations. Again, the complexities of the MPEG format have made this capability all but absent in PC-based decoding solutions. Under QuickTime, MPEG-1 content can be manipulated in largely the same ways that other forms of video and audio streams can be used. In fact, it is entirely straightforward to use

QuickTime's audio mixing and image compositing features to combine MPEG content with other QuickTime media types such as text titles, sprite animations, or music tracks.

For software developers, QuickTime makes adding support for MPEG streams to your applications an easy task. Under QuickTime, MPEG content is accessed and controlled from your application in the same way that other media types are supported. When this ease-of-integration is considered alongside QuickTime's high-quality, high-performance MPEG-1 software decode capabilities, MPEG media becomes a more compelling data type for many uses.

QuickTime 3.0 adds support for a media encoding architecture. Designed with the needs of MPEG encoders in mind, it is expected that QuickTime's support for MPEG media will continue to expand, while at the same time, the qualities that have made the integration of MPEG-1 and QuickTime a natural and powerful combination will be preserved.

As QuickTime's MPEG support is enhanced further to support more types of MPEG content such as MPEG Layer III audio, MPEG-2, and its variants used in DVD-Video, it is expected that QuickTime will be able to support similar ease-of-use and integration features that have been provided for MPEG-1.

## **DV**

In the last year, most major video camera manufacturers have introduced digital cameras based on the DV standard. Examples of these cameras include the Sony DSR-30 and DCR-VX 1000 and the Panasonic AJ-D750 and AJ-D700. These DV cameras use a new tape format which stores the video and sound in a purely digital format. The video and sound data can be transferred to another DV camera or deck using a new high speed connector called FireWire (also known as IEEE 1394). Because the video and sound are transferred through a digital connector there is no loss of image or sound quality - every copy is identical to the original.

Hardware manufacturers such as Apple, TrueVision, Miro, and Adaptec are creating products which provide FireWire connectors for personal computers. The availability of FireWire connectors on personal computers means that users will be able to digitally transfer the sound and video captured with their DV cameras into their computers. The video and sound quality of these DV clips are of very high quality, in part because of the many advantages of the digital tape format and digital data transfer.

Once DV clips are stored on the computer's hard disk, the issue that arises is how to work with the video. The manufacturers of the DV cameras created an entirely new compression technology to store the video, and a new file format to go with it. As a result, there are almost no dedicated tools available to work with DV media. QuickTime bridges the differences between the new DV format and existing sound and video tools.

The QuickTime software architecture has the ability to work with a wide range of digital media file formats and video compression formats. QuickTime 3.0 has built in support for the DV file format. This means that all currently QuickTime enabled applications are able to work with DV streams without any changes. DV data can be played back, edited, combined with other digital video standards like Motion JPEG, and even converted into other formats. QuickTime users don't have to learn a whole new set of tools to take advantage of DV

Because DV data can be used from any QuickTime enabled application, the DV sound and video can be easily converted to other formats for delivery. The final format might be a

QuickTime movie compressed with Cinepak or Indeo for CD-ROM delivery, a QuickTime movie compressed with ClearVideo for Internet delivery, or an MPEG bit-stream for delivery on a Video CD. In all these cases, the highest possible image quality is maintained because all data is digitally transferred, so no noise is ever introduced into the signal.

Another place that edited DV data may be delivered to is back to a DV tape. QuickTime contains a DV encoder which enables users to create DV streams directly. If users have been working with DV data, no recompression is necessary so the full image quality is maintained in the final DV stream. But users can also start with more traditional digital video formats like Motion JPEG. QuickTime can re-encode the Motion JPEG media into the DV format. Once QuickTime has encoded the digital media into a DV stream, that file can be transferred back to a DV camera or deck using FireWire.

All of QuickTime's support for DV is software based. This means that any computer running QuickTime can access DV data. On high end machines, QuickTime's software DV decoder will provide the ability to play back DV streams at quarter image size. And for off-line processing, QuickTime's software DV decoder can provide full frame, full quality images. On other machines, the performance will be lower. Many users require full screen, full motion play back in order to accurately edit their DV media. The QuickTime architecture is both scaleable and extensible, enabling third party developers to accelerate QuickTime with DV play back hardware. Many companies have announced support for QuickTime 3.0's DV capabilities including Promax, Octopos, DPS, Miro, Adaptec, and Radius. These DV accelerators will allow full screen, full motion playback and editing of DV media. And because of QuickTime's Media Abstraction Layer, the DV acceleration hardware will work with all QuickTime enabled applications, not just those written to work with a particular vendor's hardware.

The built-in capabilities of QuickTime 3.0 mean that users of DV cameras don't have to worry about the details of the digital media formats they are using. Users can choose from a wide range of QuickTime enabled tools for their DV editing, instead of having to wait for tools to emerge that support this new video format. Users can easily work with DV media in the same way they work with digital video today. This ease of use, high level of integration, and rapid technology adoption are all possible because the QuickTime architecture was designed to be easily extended to accommodate new technologies such as DV as they continue to emerge.

## **Visual Effects**

One of the most time consuming aspects of working with digital video on a computer workstation today is rendering of visual effects. Video editors have often have to limit their use of motion, transitions and filters because of the lengthy rendering times that these effects can require. Hardware vendors have created plug-in boards that can accelerate the rendering of many of these effects, often performing them in real-time. Unfortunately, these hardware solutions have only been available in closed proprietary systems. There have been no open standards for real time visual effects.

QuickTime 3.0 contains enhancements to the QuickTime software architecture which will enable QuickTime based applications to work with visual effects in a standard way, and take advantage of hardware acceleration when available. QuickTime's support for visual effects is comprised of three key elements.

- Standard effects description container. QuickTime 3.0 defines a standard way to describe visual effects. This effects description is built on a flexible and extensible data container format called QuickTime Atoms. The use of the QuickTime Atom container format ensures

that QuickTime will be able to describe any visual effect in use today, and the amazing new effects that will be created in the future. By providing a standard container for describing visual effects, QuickTime provides a common means for software applications to communicate with visual effects rendering engines. This enables the same description to be used with either a hardware or a software visual effects render, ensuring that the description of the effect is independent of the underlying technology used to render the effect.

- Built-in renderers for common effects. While QuickTime 3.0 has the ability to work with any kind of visual effect, there are a set of visual effects which are commonly used. To ensure that all users have access to these common effects, QuickTime 3.0 includes a rich set of built-in software based effects. These effects include cross-fade, chroma keying, SMPTE wipes, and color adjustments. By providing a complete set of standard visual effects, QuickTime 3.0 helps to establish common ground around which greater industry-wide standardization of effects descriptions can be achieved.
- Architecture supports both hardware and software visual effects renders. QuickTime 3.0 provides a software architecture that makes it possible for video editing applications to work with visual effects which are implemented both in hardware and software. QuickTime isolates the application developer from the details of how the effect is implemented through its media abstraction layer. This enables an application to transparently take advantage of visual effects hardware accelerators. It enables suppliers of visual effects to write to a single programming interface to supply their effects to any application on any platform. And as more and more visual effects are available for QuickTime, it will enable users to use a wider range of visual effects in a wider range of tools.

QuickTime 3.0 provides the foundation for more productive video editing. Users will be able to use hardware accelerated visual effects in a range of tools. QuickTime provides the common ground that enables software effects, hardware accelerated effects, and video editing tools to work transparently together. In many cases, users will be able to directly play back the results of their edits without any rendering delays. And as new effects algorithms become available, whether in hardware or software, QuickTime enabled applications will be able to take immediate advantage of them.

## **Audio**

Because audio plays such an important part in almost any digital media project, QuickTime provides an unparalleled level of support for working with all kinds of digital audio. QuickTime's support for audio goes well beyond the ability to simply synchronize one audio channel with video play back. A QuickTime movie file can contain an unlimited number of parallel sound tracks. Each sound track has an independent sample rate, sample size, and audio format. This flexibility is critical for digital media creators who are working with audio from a variety of different types of devices. For example, digital audio from an Audio CD has a sample rate of 44.1 kHz, from a DAT 48 kHz, and from a DVCam 32 kHz. The ability to store any combination of audio formats and sample rates in a single file is a powerful feature of the QuickTime movie format as it means that digital media creators can effortlessly combine audio from a variety of sources without having to be concerned about format differences.

Because the QuickTime Movie file format allows for complex digital audio compositions to be created, the QuickTime software architecture must be able to support the playback of these compositions. Built into QuickTime is the ability to mix together an arbitrary number of simultaneous audio channels. The only limitation on the number of simultaneous audio channels is the performance of the computer that QuickTime is running on. The ability to

work with an unlimited number of audio channels, each potentially of a different audio format is made possible because of the Sound Manager portion of the QuickTime software architecture.

The Sound Manager provides a rich set of services for playing, mixing, synchronizing, digitizing, decompressing, and compressing digital audio. The Sound Manager contains software services which allow a system with only one or two hardware audio channels to play back any number of digital audio channels. If the digital audio is compressed, the Sound Manager will decompress it in real-time as the audio is being played.

While the Sound Manager provides all of its services in software, any of its capabilities can be enhanced by the addition of hardware accelerators. For example, the audio mixing and sample rate conversion can be carried out by a hardware card which might use more sophisticated algorithms than the Sound Manager can provide in real-time on the system's main processor. Because of QuickTime's hardware abstraction layer, users can install such hardware and it becomes transparently available to all QuickTime-enabled applications.

The Sound Manager also provides powerful synchronization services. A sample clock is available for each audio channel which allows applications to know exactly which audio sample is currently being played. This sample clock also provides the basis for synchronizing other digital media with the audio channel, for example digital video or another channel of audio. Without the availability of a sample clock applications can never know exactly which audio sample is currently being played, which makes precise synchronization impossible. Yet many other systems for working with digital media fail to provide this essential capability.

While the services of the Sound Manager for mixing, resampling, and decompressing are most commonly used in real-time for playing back one or more channels of digital audio, these same services can be used by tools in an off-line mode. This capability is important because it means that tools can leverage the built-in capabilities of QuickTime to provide some, or all, of their audio processing capabilities. For users, this means that more tools are able to provide a broader range audio capabilities, because QuickTime makes providing fundamental audio capabilities easy.

Because there are so many different file formats for storing digital audio, QuickTime provides the ability to work most popular audio file formats. In addition to the QuickTime Movie format, which is an extremely flexible means for storing digital audio compositions, QuickTime also supports the following other audio formats: AIFC, AIFF, AU, AVI, Macintosh Sound Resources, OMF, Sound Designer II,  $\mu$ Law, and Wave. Because QuickTime provides access to so many standard audio file formats, tool developers are able to easily support these audio file formats as well. For users of these tools, that means no need to worry about what format digital audio is stored in because QuickTime makes it possible to work with most popular formats.

## **Music/MIDI**

While the audio services provided by QuickTime are extremely powerful and flexible, there are a number of applications where standard audio data is either impractical due to size or data rate constraints or where there exists a need for a more interactive form of audio experience. Especially in the context of the Internet, the challenges of delivering high-quality, interactive audio over the relatively narrow-band data channels most users can access are substantial.

MIDI provides very compact representations of musical performances. The MIDI data format describes a performance in terms of a very compact time-based sequence of discrete musical events such as keyboard events (key downs, key ups, pedal events, etc.) rather than as a series of digitized audio samples. Furthermore, since MIDI separates the description of the musical performance from the descriptions of the specific instruments needed to realize the complete composition, a great deal of interactive potential is embodied in a MIDI performance.

QuickTime provides rich support for MIDI within its interactive music architecture. In fact the QuickTime Music Architecture exists entirely apart from MIDI. MIDI is not required to take advantage of the QuickTime Music Architecture. However, as a practical matter most music applications work with MIDI data today. So for compatibility reasons, QuickTime supports a superset of the functionality implied by the MIDI format. The QuickTime Music Architecture is composed of four major elements:

- Music Synthesizer Architecture
- Instrument Library Component
- Note Sequencer
- MIDI Importer/Exporter

QuickTime provides a fully-specified architecture for implementing and accessing music synthesizers. These synthesizers can be hardware-based or software-based. There is a well-defined architecture to allow different synthesizers to plug-in to QuickTime. By default, QuickTime 3.0 includes a built-in software-based synthesizer that provides a rich set of capabilities that can be delivered across the full range of computers that support QuickTime. For composers and title developers there is a huge benefit to providing a ready-to-use synthesizer in the base QuickTime architecture. With standard MIDI data, there has been a long-standing problem of ensuring consistency of the MIDI performance across a wide variety of software and hardware platforms. While General MIDI addresses this problem somewhat, it has its own drawbacks. By delivering music content through QuickTime, the issue of consistency is largely solved since QuickTime is able to ensure a consistent platform for music playback across all the various software and hardware systems.

All music synthesizers need instruments to be able to make sound. To that end, QuickTime augments its built-in synthesizer with a General MIDI compatible instrument library. This library provides a standard set of instruments in a very compact form. Since the range of expressiveness of a synthesizer is largely defined by the sorts of instruments it can utilize, the QuickTime Music Architecture also includes support for custom instruments that can augment or replace the standard instruments that ship with QuickTime 3.0. Within this framework, there is virtually no limit to the range of instrumentation that can be created in QuickTime.

The third key element of the QuickTime Music Architecture is the note sequencer. The note sequencer component serves the purpose of communicating with QuickTime synthesizers to start and stop individual notes. The note sequencer manages an overall performance as described by a list of note events to be performed over time. By including this higher-level service, much of the difficulty of working with music data from within applications can be eliminated. It becomes a simple matter to create, edit, and control the playback of complex compositions.

The last component of the QuickTime Music Architecture is the MIDI file import/export service. By providing data interchange with MIDI files, it becomes a simple matter to work with a wide range of existing music composition and editing software tools. As with

support for other media file formats, the MIDI file importer significantly reduces the effort required to integrate standard music data formats with other QuickTime media.

## **Summary**

This overview of QuickTime 3.0 has provided a small sampling of the full range of capabilities available in QuickTime. Only a subset of the supported media types has been discussed in any detail. Within each of the areas that has been addressed, limitations of space dictate that only a small portion of the full QuickTime story can be told. However, it is hoped that from these descriptions, something of the power and flexibility of the QuickTime approach to digital media can be seen.

It has always been the goal of QuickTime to help simplify the use and integration of digital media of all kinds. As QuickTime has evolved since its introduction in 1991, an ever expanding set of built-in services has been added to the base architecture. Over the years, support for a much richer set of media types has been added. And at the same time these advancements have been added to QuickTime, the key principles and architectural concepts introduced in QuickTime 1.0 remain intact in version 3.0. This stability has enabled software and hardware vendors to enjoy excellent overall compatibility and technological continuity while being able to reap the substantial benefits of an evolving QuickTime.