find it

main

T E C H N O T E 1087

Maximizing Your Media:

A Brief Guide To The Latest and Greatest QuickTime Media Types

By Drew Colace
Apple Developer Technical Support (DTS)
drex@apple.com

CONTENTS

In 1991, creating a QuickTime movie player or editor was relatively straightforward. Back then, there were two types of media to work with -- video and sound -- and a canvas of APIs. At the time, QuickTime demos featured trees swaying in the wind, clouds rolling overhead, and the moon cascading its light down through an evening sky -- and the movies were postage stamp size! Not the most compelling examples of multimedia. Now, don't get me wrong. I love QuickTime, but I'm glad those days are over!

Today, QuickTime delivers the most dynamic media available -- professional quality video and sound, MPEG, 3D objects, sprites, text and MIDI, and its capabilities are continuing to grow. With QuickTime 2.5, it is easier now for movie tracks to act on one another, as is the case with our Tween track, which sends its information to another track to change it dynamically over time. The possibilities for truly exposive media content are there, yet so many QuickTime applications act as though we are still back in 1991 and can only deal with video and sound tracks. Well folks, it's many years later and it's time to get with the program.

This Technote explains how you can make sure your QuickTime application isn't blind to any media types available to it and some simple things you can do with that media to add value to your application.

Identifying the Media

Some applications, in an effort to make sure they can successfully edit or otherwise manipulate a particular movie's data, will take a look at the movie's media to see what types are available. In general, this was fine in the early days, since video and sound were the first two pioneering media types of the QuickTime frontier. But now, with the burgeoning of new media types, QuickTime applications are not necessarily taking full advantage of their capabilities by looking for particular media types.

Using `GetMovieIndTrackType`

Instead of looking for a specific media type, you should be looking for a media characteristic. In QuickTime 2.0, Apple introduced `GetMovieIndTrackType()` for this explicit purpose. You can use this call to conveniently find those tracks which draw and those which play sound using the two flags, `VisualMediaCharacteristic` and `AudioMediaCharacteristic`. For the standard media types that are included with QuickTime 2.5, `VisualMediaCharacteristic` will pick out the following types of tracks: Video, MPEG, Sprite, QuickDraw3D, Text, TimeCode. `AudioMediaCharacteristic` will pick out: Sound, Music, MPEG. (MPEG is somewhat of a special case, since it can contain both sound and video data in one stream and hence one track.)

Not only can you use `GetMovieIndTrackType` to find tracks that share a common media characteristic, but you can also use it to find tracks that share a particular media type when you really need that information. Which search method you employ is controlled by the flags that you pass into the last parameter of the call. A fully parameterized call to look for tracks with media that has a video characteristic would look like:

```
theTrack = GetMovieIndTrackType(sourceMovie, trackIndex, VisualMediaCharacteristic,
movieTrackCharacteristic);
```

whereas a fully parameterized call to look for tracks with media of a certain type would look like:

```
theTrack = GetMovieIndTrackType(sourceMovie, trackIndex, VideoMediaType,
movieTrackMediaType);
```

You might use one of these calls in a routine like this:

```
Track  theTrack;
long   trackIndex = 0;

while(theTrack = GetMovieIndTrackType(sourceMovie, ++trackIndex,
VisualMediaCharacteristic, movieTrackCharacteristic)) {

// theTrack meets your search request so do something with it

}
```

Note:
You can use either `movieTrackMediaType` or `movieTrackCharacteristic` but not both in a single call to `GetMovieIndTrackType()`; you can combine, however, `movieTrackEnabledOnly` with either call. Also, the trackIndex value is not equal to a track's number. Rather, it is an indexed value of the tracks found that meet your criterion.

Using Video Media

Video media is one of the first things you think about when working with QuickTime, since so many "movies" are pictures and sound. Video media can be found by calling `GetMovieIndTrackType()`, using either the `VideoMediaType` or `VideoMediaCharacteristic` flags. While basically all QuickTime applications deal with video media in one way or another, there are several cool things that you can do to take advantage of some new as well as some old QuickTime features.

First Cool Thing: Using `SetTrackClipRgn()`

You can use `SetTrackClipRgn()` to apply an interesting or aesthetic frame to a video track. Many developers have shied away from this feature because clipping typically slows down QuickTime movie playback and may affect the overall presentation. However, there are several QuickTime acceleration cards available today that have hardware assistance for clipping, scaling and color conversion, thus unburdening QuickTime from having to make these modifications itself. And, of course, today's PowerMacs are really starting to scream. Unless you're going for full-screen, full-motion video, many PowerMacs can handle the overhead of applying and using a track clip region. Which leads to my second point about scaling and translation.

Second Cool Thing: Using `SetTrackMatrix()`

Using `SetTrackMatrix()`, you can scale and distort (but not yet rotate) your video track in interesting ways. Again, with today's video cards, many can provide hardware assistance in this operation, giving you creative flexibility like you haven't had before.

Note:
While both the track's clip region and matrix can be set statically using either of the aforementioned calls, both can be changed dynamically by using a Tween track.

More recently introduced are the concepts of Track References and Modifier Tracks. One way to provide more value to video tracks is to add a text track which has descriptions of the various parts of the video track. This idea has been promoted since the Text track's introduction in QuickTime 1.5. Initially, the thought was to add a text track to act as subtitling for movie dialog or as descriptive text to identify a scene or sound clip. But now text tracks have a new capability when you are using the Movie Controller interface. Text track data can now be used as chapter or scene descriptors with its info being displayed in a pop-up menu in the Movie Controller play bar. Typically, the text track is hidden or disabled, although the track can be displayed as well (its utility and aesthetic quality would seem to detract from the controller's function). In order to get the movie controller to display the chapter name in the play bar, you must create a reference between the text track containing the scene names and one or more of the other tracks in the movie. This is done using `AddTrackReference`. In the example below, the reference is added from a video track to the text track. It could also be added from a sound track, an MPEG track - just about any type.

```
AddTrackReference(videoTrack, textTrack, 'chap', nil);
```

Sound Media

QuickTime sound tracks have a media type of `SoundMediaType` and have an `AudioMediaCharacteristic`. The use of QuickTime sounds has been enhanced over the years with the introduction of new sound compression methods. This has enabled the use of much higher quality and quantity of audio in applications and multimedia titles. It is also just starting to enable some new possibilities in using this media type over the Internet.

Moving Sound Around

Yes, just playing sounds hardly stands out today. And since you have a few new controls at your fingertips that can make your sound more dynamic, it just might be time to start exercising that option. Using `MediaSetSoundBalance()`, you can adjust the left-to-right balance of any track that has an `AudioMediaCharacteristic`. Setting the balance is hardly a new concept, but how it gets used in an application is something to discuss. For regular audio playback, it's reasonable to set the sound balance once and leave it for the duration of the program. However, in the case where you need dynamically changing sounds, you can still use QuickTime and the convenient QuickTime movie file container by utilizing the `MediaSetSoundBalance()` call.

Text Media

QuickTime Text tracks have a media type of `TextMediaType` and have a `VisualMediaCharacteristic`. Much of the power of the Text track is in how it adds value to other QuickTime media. In reading the other sections of this Note, you'll get an idea of some of the power of Text media.

MPEG Media

MPEG media in QuickTime has had a dual life for some time. On the one hand, there has been support for MPEG media within QuickTime since version 2.0. The other hand is the 'but' as in "...but QuickTime 2.0 requires a special hardware decoder board to see or hear that MPEG media in QuickTime."

Today, there is good news for QuickTime, MPEG and those developers who want to take advantage of both because of the introduction of the QuickTime MPEG Extension for Power Macintosh. This extension lets all Power Macs play and use MPEG media, type `MPEGMediaType`, as a full fledged citizen in the QuickTime world. But MPEG is unique among the media types in that it can contain either video or sound media or both. This is why it is so important to use the MediaCharacteristic flags to find appropriate media to deal with, since you wouldn't want to miss out on using the versatile MPEG media type. To be certain that you'll find MPEG media when it's available use either the `AudioMediaCharacteristic` flag, `VisualMediaCharacteristic` flag or both flags when checking the tracks' media characteristics.

When you do get down to work with MPEG media that has video information, you might want to take advantage of a nice capability of the QuickTime MPEG Extension. It has the ability to set a rectangular clip region on visual MPEG media without any performance penalty. This can be especially useful for poorly captured and encoded MPEG media that show visual garbage along the edges of the video frame - unfortunately, an all too common occurence. Using scaling and clipping during runtime can handily hide the glitch.

QuickDraw3D Media

QuickDraw3D media, recently introduced with QuickTime 2.5, has a media type of `ThreeDeeMediaType` and has a `VideoMediaCharacteristic`. Besides being unique because its sample data is made up of 3D objects, QuickDraw 3D media is also very interesting because of its relationship to the various other media types. Besides being unique because its sample data is made up of 3D objects, QuickDraw 3D media is also very interesting because of its relationship to the various other media types. With most other media, new sample data is needed over time to see or hear anything as a movie plays. Although you can approach using QuickDraw 3D data the same way, it is much more powerful to use the new Tween track (in QuickTime 2.5) and its sample data to override QuickDraw 3D objects, lights and cameras, therefore animating a 3D model using relatively little data bandwidth. Also, it is possible to map the data from video tracks onto the surface of a QuickDraw 3D object in a QuickTime movie - and the video track can play and change as you might expect, all while being blasted onto the face of a 3D object. How's that for interesting?!

Sprite Media

QuickTime Sprites may be the most misunderstood media type that QuickTime supports, probably because its elements do not necessarily have to be confined to a timeline or run from beginning to end. But, to get started, you can think of using QuickTime Sprites in the traditional way -- a linear track moving from start to finish -- then grow from that point.

QuickTime Music (MIDI) Media

QuickTime Music, often thought of as MIDI, has been available in QuickTime since version 2.0 but has recently been significantly enhanced in QuickTime 2.5.

Summary

QuickTime, being more versatile and powerful today, calls out to your application to take full advantage of its capabilities Some people may think that flipbook animations, such as those provided by Java and animated GIFs, are the great. Well, I thought they were great, too, in 1987 when VideoWorks first came out, but with all the VERY COOL features that QuickTime 2.5 offers today, it is almost a crime against humanity not to do something with them. Therefore, you ought to use some of the methods discussed in this Note to find and manipulate the new media types. That way your app can be the best it can be!

Further References

- *Inside Macintosh: QuickTime*
- *QuickTime 2.0 SDK Documentation*
- *QuickTime 2.1 Developer Note*

Acknowledgments

Thanks to the QuickTime engineering team for information, inspiration and perspiration!

Send feedback to drex@apple.com
Updated: 20-Dec-96