

New Technical Notes

Macintosh



®

Developer Support

Movie Toolbox Q&As

QuickTime

M.QT.MovieTB.Q&As

Revised by: Developer Support Center

October 1992

Written by: Developer Support Center

October 1990

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

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

|New Q&As and Q&As revised this month are marked with a bar in the side margin.

QuickTime Movie Toolbox “globals” are stored in system heap

Written:

Last reviewed:

According to the QuickTime Movie Toolbox documentation, “The Movie Toolbox maintains a set of global variables for every application using it.” How much global memory is required? Our application is shy on global data space.

—

The information maintained is not kept with the application’s global variables. The handle created by the EnterMovies call is stored in the system heap, not in the application heap. You don’t have to worry about how much space to allocate in your application. This initialization does not affect your A5 world either.

EnterMovies initializes everything, including setting up the necessary data space and creating a handle to it. When you’re done, be sure to make a call to ExitMovies to clean up the QuickTime data.

If an application makes multiple calls to EnterMovies, a different set of “globals,” or data area, is set up for each call. A call to ExitMovies should be made before exiting the area that made the call to EnterMovies. For example, an application that uses QuickTime will call EnterMovies and set up the QuickTime world. Then an external may be called upon that wants to use QuickTime. This external would have to make a call to EnterMovies to set up another QuickTime world for its use. Before leaving, the external should call ExitMovies to clean up

after itself. The application can then continue to use QuickTime with the original world it set up.

QuickTime file audio retrieval

Written:

Last reviewed:

How can I retrieve audio from QuickTime files in 1-second chunks? I need a sound equivalent of GetMoviePict.

—

You will have to write your own audio extraction routine. The Movie Toolbox does not provide a simple call to, say, GetMovieSound(movie, time, duration). To get around this limitation, use GetMediaSample and collect the sound data by hand. The format for the final result is up to you; in the case of a 'snd ' resource you'd create the handle fill in the other fields and put the sample data in it.

You can use the QuickTime SoundDescription data structure (1) to find frequency, format, and other details, (2) to help you figure out how many samples you need for a second, and (3) to fill in the resource data fields.

Of course, this scheme gets more complicated in the case when there are more than a single sound track. The best source of information on sound resources and such is the Sound Manager chapter in Inside Macintosh Volume VI.

Current QuickTime supports only video and sound media

Written:

Last reviewed:

Can I add a media to a QuickTime movie that is not video or audio? If so is there anything special I need to do to add text notes that can potentially accompany each frame in my "movie," which can follow the video frames if a user edits the movie in any way.

—

The current release of QuickTime only allows for video and sound media. There is no way to install your own type. This is a high QuickTime priority and is likely to make it in a future release, but at the present there is no mechanism to do it.

Status of rotating matrix support

Written:

Last reviewed:

What is the status of RotateMatrix and its use with SetMovieMatrix and SetTrackMatrix?

RotateMatrix works fine. But rotating matrixes are not supported for movies or images. So, although RotateMatrix will give you the correct mathematical result, unless you are using the matrix to transform something else (as with TransformFixedPoints) it has little use.

Rotation is a very important future direction that will get more attention in the future.

How to get the first video frame

Written:

Last reviewed:

Stepping through QuickTime movie video frames in the order they appear in the movie is simple using `GetMovieNextInterestingTime`, except for getting the first frame. If I set the time to zero and rate to 1, then I get the second frame, not the first. In addition, the video may start later than at 0. How do you suggest finding this first frame of video?

—

To get the first frame in the conditions you describe you have to pass the flag `nextTimeEdgeOK = $2000` to `GetMovieInterestingTime`. What this flag does is to make the call return the current interesting time instead of the next, if the current time is an interesting time. You need to do this since there is no way to go negative and then ask for the next interesting time.

QuickTime interfaces for Think Pascal

Written:

Last reviewed:

Has Apple created QuickTime headers/interfaces for Think Pascal? The MPW Pascal headers don't seem to be compatible.

—

There are no Think Pascal interfaces, but your Pascal Package comes with a program called Pascal Source Converter, which converts MPW Pascal sources to Think. This should be all you need to be able to use the provided interfaces with your favorite development package.

Standard controller with MCCut or MCClear

Written:

Last reviewed:

When I select all frames, then do an `MCCut` or `MCClear`, the standard controller gets larger and redraws itself at the top of the movie. Is this a situation I should be prepared to handle or a bug? Does the controller behave strangely when the `selectionTime` of a movie is -1 or when the duration of the movie is 0?

—

The behavior you are observing is to be expected if the controller is attached to the movie. In this case, the controller goes to wherever the bottom-left corner of the movie box takes it. If the movie loses all of its "visible" parts, the movie controller will jump to the top of the

window.

The only way to get around this is to detach the controller when the movie box is empty; this is also something to keep in mind for the cases when the movie contains only sound, since pure sound movies have no dimensions.

QuickTime and sound channel deallocation

Written:

Last reviewed:

Our QuickTime application gets a Sound Manager error -201 after playing movies in succession, apparently because sound channels used in the previous movies have not been reclaimed. How does QuickTime decide to deallocate sound channels? It doesn't seem to happen in my "while (!IsMovieDone(theMovie) && !Button())_" play loop.

Sound channels are released by active movies when they notice that some other movie needs them. This is currently only done at MoviesTask time. Before entering your loop to play a single movie, you can do one or both of the following:

- Preroll the movie you are about to play and check the error. If preroll returns -201 then do a MoviesTask(0,0) to give the other active movies a chance to give up their sound channels. A subsequent preroll of theMovie should return noErr.
- SetMovieActive(othermovies, false). Deactivate the movies that you aren't playing to force them to give up their resources.

Clipping QuickTime movie posters

Written:

Last reviewed:

Our application uses the movie poster as a still frame in a cell, similar to using a PICT. If a user sizes the cell width so it's narrower than the poster, even though we clip the drawing to the cell size, QuickTime posters draw their full width, writing over whatever is in the way. Pictures clip through DrawPicture; why doesn't ShowMoviePoster stay within clip region?

ShowMoviePoster, as well as the movie and preview showing calls, uses the movie clipping characteristics rather than the destination port cliprgn. You must set the movie's clip to obtain the results you want. An easier way to do this is to get the picture for the poster, by calling GetMoviePosterPict, and then simply use DrawPicture to display the poster. Because this is just a picture, the clip region of the port is honored. This way you don't need different code for movies and pictures.

PutMovieIntoHandle and data forks

Written:

Last reviewed:

I save PICTs to my document's data fork by writing the contents of the PicHandle. To save movies, do I convert the movie to a handle, and then save that like I would with PICTs? I just want the file references, not the data itself.

To save movies that are suitable for storage in a file, use `PutMovieIntoHandle`. The result of this call can then be saved in the data fork of your files, and then you can call `NewMovieFromHandle` to reconstruct the movie for playback or editing.

You should also read the documentation regarding Movie Toolbox `FlattenMovie`, which creates a file that contains the 'moov' resource and the data all in the data fork. The advantage here is that the movie file you create using `FlattenMovie` can be read by any other QuickTime-capable application.

SetMovieRate and controlling movie playback rate

Written:

Last reviewed:

QuickTime is a joy! But I've run aground with SetMovieRate. I am trying to change the rate at which a movie plays back, but if I call SetMovieRate the movie starts playing immediately, the controller goes wild and the next time I hit the play button it ignores the previous rate.

—

SetMovieRate takes effect immediately; that's why the movie starts playing as soon as you make the call, with a rate other than zero. Also, calling SetMovieRate behind the controller's back can only cause confusion because you are changing the state of the movie without letting the movie controller know about the change. Note that in normal operation the movie controller plays back movies at the standard speed, rate = 1; this is the current behavior. It is possible that in a future release the movie controller will use the rate the movie was saved with or the one set with SetPreferredMovieRate.

A little known fact is that the standard controller does contain a primitive mechanism for controlling the rate of playback. If you hold the control key down and then click the mouse over the stepping buttons, you can, for example, play the movie backwards. Furthermore, if you hold the mouse down you'll get a slider control that does let you play the movie at different rates backward or forward.

The slider provided by the standard controller is not intended to set the rate, so if you play once at low speed the rate does not stick and, as you have found, the next time you click on the play button you go back to the normal speed. If you need the selected rate to remain for the session, you'll have to provide your own method of selection.

Once you know your desired speed, you'll need to provide your own filter procedure and install it calling MCSetActionFilter. Upon receiving any mcActionPlay actions for rate changes, you'll need to call SetMovieRate to set the movie in motion at the desired rate (and return true). Using a filter proc is the legal way of doing this because the controller can keep in sync with the actions regardless of the fact that it's your code that actually affects the action.

Note that you will have to do some extra work to mimic the normal behavior of the standard controller. For example, when you're at the end of the movie and the user hits the play button, the controller goes back to the beginning and plays the movie. Your

QuickTime track and movie sound volume

Written:

Last reviewed:

What do the values of a movie's or track's volume represent? Is there no way to make a track louder?

—

Here's the scoop:

The volume is described as a small fract 8:8 and its values go from -1 to 1 with negative values as placeholders. The maximum volume you can get is 0x0100 with the minimum being 0 (or any negative value). The advantage of using negative volumes is that you can turn off sound

while maintaining the level of volume. For example, -1 and 0 both equate to no volume, but the -1 implies that 1 should be the volume when sound is turned back on, whereas the 0 does not.

The volume for a track is scaled to the movie's volume, and the movie's volume is scaled to the value the user specifies for the speaker volume using the Sound control panel. This means that the movie volume represents the maximum loudness of any track in the movie.

QuickTime 1.0 FS6Patch elucidation

Written:

Last reviewed:

The FS6Patch description in the QuickTime 1.0 CD Read Me file says, "There is a known bug with HFS on System 6." What specifically is this bug? According to the Read Me file, "The exact situation under which the problem occurs is somewhat rare." What is the exact situation? "If the patch is not necessary, it will not install itself." What criteria is used to determine necessity?

—

When closing or flushing a file that has multiple open paths to a given fork, it is possible that the blocks marked as available for the other paths can also be allocated to the other fork. If this happens before the other paths are closed, blocks will be mapped to more than one fork, potentially trashing at least one of them.

The condition that can trigger this condition is multiple writeable paths to the same fork. Once this is satisfied you would need to make some changes to the fork, close the path, use any of the other paths to modify the fork, access the other fork and modify data. If all these things happen you could see the damage to one of the forks (the second-to-last being accessed).

The code that loads the patch checks the system first; the range where the patch is used is $6.0.4 < \text{system} < 7.0$. The version of the ROM is also used to decide when to install. If the ROM indicates the machine is later than the Macintosh IIfx, then a fix in ROM is assumed. The last check is to see if the code being patched is already in RAM (the assumption is that being in RAM means fixed). If so, the patch doesn't get used.

QuickTime alias and FSSpec system services under System 6

Written:

Last reviewed:

I recall reading that QuickTime includes an implementation of the Alias Manager for System 6, but I haven't found any precise description of just what is included. Is it a bare minimum to support QuickTime? Or is the full Alias Manager there? Also, is there any way I can use

the FSSpec interface to the File Manager, or must I revert to the System 6 interface?

Here is some information on the alias and FSSpec system services that QuickTime supplies under System 6:

1. QuickTime support for aliases on System 6:

Most of the Alias Manager is available, with these few exceptions:

- NewAlias will accept a “fromFile” parameter, but it never creates a relative alias.

- NewAliasMinimalFromFullPath is not available.
- ResolveAlias will accept a “fromFile” parameter, but it ignores it.
- ResolveAliasFile is not present.
- MatchAlias may be called, but the kARMSearchMore, kARMSearchRelFirst, and kARMMultVols flags are not available. If you pass them in, they will be ignored. Furthermore, if you pass in a matchProc, it will never be called.
- UpdateAlias will accept a “fromFile” parameter, but ignores it.
- The System 6 Alias Manager will not mount network volumes.

So to summarize, on System 6 the Alias Manager doesn’t handle relative aliases, multiple volume searches, “searchMore” searches, and network volume mounting.

On the good side, nearly all calls are present. Aliases created on System 6 are compatible with System 7 aliases. And Aliases made on System 7 will work on System 6.

Unfortunately, QuickTime does not currently install an Alias Manager Gestalt selector as it is only a partial implementation. You can check for the Alias Manager using Gestalt and if it is not present, look for QuickTime (using Gestalt) and if QuickTime is present, assume you have an Alias Manager, subject to the limitations above.

2. FSp file system calls and System 6

QuickTime also makes extensive use of the FSSpec data structure introduced in the System 7 File Manager. Nearly all the FSSpec calls are available on System 6 when QuickTime is installed. The following calls are available on System 6, and should behave as documented for System 7:

- FSMakeFSSpec
- FSpOpenDF
- FSpOpenRF
- FSpCreate
- FSpDirCreate
- FSpDelete
- FSpGetFInfo
- FSpSetFInfo
- FSpSetFLock
- FSpRstFLock
- FSpRename
- FSpCatMove
- FSpOpenResFile
- FSpCreateResFile
- FSpGetCatInfo

The following call is not available when using the QuickTime System 6 version of the FSp calls:

FSpExchangeFiles

Again, the Gestalt selector for the FSp calls is not installed when QuickTime is there. The means that the gestaltFSAttr Gestalt selector may not be present, and gestaltHasFSSpecCalls may not be set, even if gestaltFSAttr is present.

'pnot' resource format for QuickTime-like preview in dialog box

Written:

Last reviewed:

I would like to implement the preview/thumbnail feature in the Standard File dialog, just like the extension included with QuickTime. Is that code available separate from QuickTime? If not, could I at least get information on how the preview is created so that I use the same technique?

—

To implement your own preview/thumbnail feature, simply duplicate the Standard File dialog, add the necessary 'DITL' resources, and install a custom filter procedure for handling preview commands. On the System 7 CD is an example, StdFileSample, which shows exactly how to create a custom file dialog. The Macintosh Technical Note “Customizing Standard File” describes how to do this as well. For generating and displaying the preview, you can use the following PreviewResourceRecord, found at the end of

MCSetClip and clipping with the movie controller

Written:

Last reviewed:

I use SetMovieDisplayClipRgn to set my movie clip, but the movie doesn't obey my clipping. Does the movie controller component ignore this clipping?

—

You probably are directly modifying a movie that is attached to a controller without notifying the controller of the changes. The controller uses the display clip for its own purposes, such as for badges.

If you want to do clipping with the movie controller you must use MCSetClip. MCSetClip takes two regions. The first clips both the movie and the controller. The second clips just the movie, and is equivalent to the movie display clip. If both clips are set, the controller does the right thing and merges them as appropriate. If you don't want one or the other of the clips, set them to zero.

In general, if you are going to do something to a movie that is attached to a controller you must either do it through the controller, using the action calls, or you must call MCMovieChanged. Otherwise, the controller would need to constantly poll the movie to see if its state changed. Clearly this would be Evil and Slow.

Problem with disabling a movie video track

Written: 5/29/92

Last reviewed: 9/15/92

When I disable a track in a movie, another random track becomes disabled as well. Is this a QuickTime bug?

Yes, this problem will be fixed in the next QuickTime release. But, there's no apparent workaround in the meantime. You'll need to remove the track since disable doesn't work.

Determining whether a movie is set to loop or not

Written: 8/11/92

Last reviewed: 9/15/92

How does Simple Player determine whether a movie is set to loop or not? Movie files that are set to loop seem to have a string of 'LOOP' at the end of the 'moov' resource. Does Simple Player check 'LOOP'?

Simple Player identifies whether movies are set to loop by looking within the user data atoms for the 'LOOP' atom, as you've noticed. It's a 4-byte Boolean in which a value of 1 means standard looping and a value of 0 means palindrome looping. Your applications should add the user data 'LOOP' atom to the end of the movie when a user chooses to loop. We recommend this method as a standard mechanism for determining the looping status of a movie. If the 'LOOP' atom doesn't exist, there's no looping. The calls you need to access this information are GetMovieUserData, GetUserData, AddUserData, and RemoveUserData, as defined in the Movie Toolbox chapter of the QuickTime documentation.