

## **QuickTime Answers to Random Questions**

by: Dick Trismen  
Object Media  
AppleLink: TRISMEN.D  
Compuserve: 71402,1110  
(908) 494-8821

### **How do I get a movie from a file?**

```
Movie GetMovieFromFile(void)
{
    OSErr err;
    SFReply reply;
    Point where = {100,100};
    SFTYPEList types;
    FSSpec movieFSSpec;
    short movieResRefNum;
    short actualResId;
    Movie theMovie;

    // Have the user choose a movie file using SFGetFile.
    // The type of a movie is 'MooV'.

    types[0] = 'MooV';
    SFGetFile( where, "\p", nil, 1, types, nil, &reply);
    if (!reply.good) return ((Movie)0);

    // Make an FSSpec record of the file.  Many Movie
    // Toolbox calls refer to files using an FSSpec.

    err =
        FSMakeFSSpec(reply.vRefNum, 0, reply.fName, &movieFSSpec);

    // Open up the movie file's resource fork using movieFSSpec.
    // The actual Resource RefNum is returned in movieResRefNum.
    // You need this number to close the file later.  fsRdPerm
    // indicates that you will only be reading from this file,
    // not writing.  The (MoviesDataRef *)0 indicates that you
    // don't want a DataReference for this file.

    err = OpenMovieFile(&movieFSSpec, *movieResRefNum,
                        fsRdPerm, (MoviesDataRef *)0 );
    if(err) return ((Movie)0 );

    // Get the movie from the file.  The movie will be returned
```

```

// in theMovie. The file to get the movie from is specified
// by movieResRefNum which is the file just opened above.
// Because movies are stored as resources, there can be more
// than one Public Movie in a movie file. You can have the
// Movie Toolbox get the first Public Movie. This is done by
// setting actualResID to DoTheRightThing. On return,
// actualResID will be set to the actual resource ID that was
// loaded. The Public Movie's resource name is also
// returned, but since you don't care here you pass in
// (char *)0. When a new movie is loaded, by default it is
// inactive. You can override this by passing in
// "newMovieActive". The final parameter indicates whether
// any media data references were updated when the file was
// loaded. For this example, you don't care, so you pass in
// "(Boolean *)0" .

```

```

actualResId = DoTheRightThing;
err = NewMovieFromFile(&theMovie, movieResRefNum,
                      &actualResId, (char *)0,
                      newMovieActive, (Boolean *)0 );
if(err) return ( (Movie)0 );

// Since the movie has now been brought into memory you can
// now close the movie file:

err = CloseMovieFile(movieResRefNum);
if(err) ; // do something

// At this point you have a movie in memory in theMovie.

return (theMovie);

```

## **How do I play a Movie with the Movie Toolbox?**

```

OSErr PlayMovie(Movie theMovie)
{
CWindowPtr window;
Rect movieBounds;

// First get the dimensions of the movie

GetMovieBox(theMovie, &movieBounds);

// The movie bounds returned may not be zero based, so use
// OffsetRect to make it zero based.

OffsetRect(&movieBounds, -movieBounds.left,

```

```

                                -movieBounds.right);
SetMovieBox(theMovie, &movieBounds);

// Make a window to play the movie in

OffsetRect(&movieBounds, 100, 100);
window = NewCWindow(0L,&movieBounds,"pMovie!",1,0,-1L,0,0L);

// Set the GWorld of the movie to the window so that it will
// draw into the window

SetMovieGWorld(theMovie, (CGrafPtr>window, 0);
if (GetMoviesError()) return;

// When the movie is loaded into memory, you don't know what
// the current time is. Reset to the start of the movie.

GotoBeginningOfMovie(theMovie);

// Make sure that the movie is active.

SetMovieActive(theMovie, TRUE);

// Start the movie playing

StartMovie(&theMovie):

while ( !IsMovieDone(theMovie) )
    MoviesTask(theMovie,0);

DisposeMovie(theMovie); // if you're done with it

```

## **How do I play a Movie with the Movie Controller?**

```

void main(void)
{
Component aMCThing;
ComponentDescription foo;
MovieController aMC;
WindowPtr theWindow;
Movie theMovie;
Boolean done;
OSErr err;

// (Initialize the managers and the MovieToolbox here.
// I don't feel like typing it in this example.)

```

```

// Use the Component Manager to find a Movie Controller
// and open it for use.

foo.componentType = 'play';
foo.componentSubType = 0;
foo.componentManufacturer = 0;
foo.componentFlags = 0;
foo.componentFlagsMask = 0;
aMCThing = FindNextComponent( (Component)0, &foo);
if (aMCThing == 0) return;
aMC = OpenComponent(aMCThing);
if(aMC == 0) return;

// get a movie and a window from somewhere (see previous example)
theWindow = some window;
theMovie = some movie;

err = MCNewAttachedController(aMC, theMovie,
                             topLeft(theWindow->portRect), theWindow);
if(err) return;

done = false;
while(!done) {
    EventRecord e;
    GetNextEvent(everyEvent, &e);

    // After getting each event, give the Movie Controller a shot
    // at handling it.  If the Movie Controller handles the event,
    // it returns TRUE, so you can skip the rest of the event loop.
    // Otherwise do your normal event processing.

    if (MCIsPlayerEvent(aMC, &e))
        // it took care of it, so continue
        continue;
    switch(e.what) {
        // do other event handling
    }
}

// When you are done with the Movie Controller, you should
// close it to release any memory it may have been using.

CloseComponent(aMC);

DisposeMovie(theMovie);
// should also dispose the window you allocated here.

ExitMovies();

```

## **How do I Write a Movie to the Scrap?**

```
OSErr err;
Handle publicMovie;

publicMovie = NewHandle(0);
err = MakePublicMovie(aMovie, publicMovie);
if (err == noErr) {
    ZeroScrap();
    HLock(publicMovie);
    err = PutScrap(GetHandleSize(publicMovie),
                  'moov', *publicMovie);
    DisposHandle(publicMovie);
    DisposeMovie(aMovie);
    /* Should put a PICT on the scrap here, too */
}
```

## **How do I Set and Show the Poster?**

```
// Note: Poster will not show in your movie window until the
//       first call to MoviesTask is made.
```

```
TimeValue    aTime;
TimeRecord    currentTime;
Movie         aMovie;

aTime = GetMovieTime( aMovie, &currentTime );
SetMoviePosterTime( aMovie, aTime );
SetMovieGWorld( aMovie, nil, nil );
MoviesTask( aMovie, 0L );
```

## **How do I use Callbacks with a Slaved TimeBase?**

```
// Useful for playing a non-linear collection of scenes
// from a movie. The new TimeBase ticks off time up to
// the total duration of all scenes to be played. The
// original TimeBase is driven by this new TimeBase as
// it jumps around to the beginning time of each scene via
// a callback.
```

```
// Does this really still work?
```

```
QTCallBack    CB;
TimeRecord    TR;
TimeBase      newTB;
```

```

pascal void      MovieCB();

newTB = NewTimeBase();
TR.value[0] = 0; TR.value[1] = durEdits;
TR.scale = GetMovieTimeScale(aMovie);
TR.base = GetMovieTimeBase(aMovie);
SetTimeBaseMasterTimeBase(TR.base, newTB, 0L);
SetTimeBaseStopTime(newTB, &TR);
CB = NewCallBack(newTB, callBackAtTime);
anyTime = 0L; /* or whatever time you want */
err = CallMeWhen(CB,MovieCB,anyLong,triggerTimeFwd,
                 anyTime,TR.scale);

StartMovie(aMovie);
SetTimeBaseRate(newTB, 0x10000); /* master TB */
while ( GetTimeBaseStatus(newTB, &aTimeRecord)
        != timeBaseAfterStopTime )
MoviesTask(aMovie,0);

DisposeCallBack(CB);
DisposeTimeBase(newTB);

/*-----*/
pascal void MovieCB(QTCallBack qt, long refCon)
{
// do something at callback time here, like jumping
// the movie to the next scene to be played.
}

```

## **What are the boilerplate calls that every Movie-playing app has to make?**

Call `EnterMovies` before you make any calls to the Movie Toolbox. Call `MoviesTask` often enough at idle time to service all of the currently running movies. Call `ExitMovies` when your app quits.

## **How do I create a movie?**

Use the **Sequence Grab Component** -- it's the easiest way to write a Movie Creation App. It does this for you:

- Previews and digitizes video and sound
- Live or single frame capture
- To disk or RAM
- Compresses video
- Creates Movie file
- Extensible

## **How do I edit a movie?**

Use Movie Toolbox calls:

- SetMovieSelection
- CopyMovieSelection
- ClearMovieSelection
- AddMovieSelection
- PasteMovieSelection

## **How do I undo edits to a movie?**

- HasMovieChanged
- NewMovieEditState
- UseMovieEditState
- DisposeMovieEditState
- Same routines for Tracks
- Only works if "Movie" structure is being changed

## **How do I save a modified movie?**

- Call FlattenMovie if you want a standalone movie
- Use SetMovieProgressProc to show your cool progress dialog
- MakePublicMovie makes the Public Movie form which can't be modified or played, but is a hell of a lot smaller.
- Use UserData section of Public Movie for your private data

## **How do I manage multiple movies at once?**

- Maintain a list of Movies yourself.
- Call SetMovieGWorld on each to set window it will play in.
- Manage a separate idle task for each movie, with a call to MoviesTask servicing just that movie. Adjust frequency of service if movie is not frontmost.
- Call SetMovieActive when you want a movie turned on.

## **How do I handle updates to a movie window?**

If you use the Movie Controller Component, it does all the work for you,

making sure your movie updates and redraws when it's uncovered, etc. Otherwise, call `UpdateMovie` yourself between `BeginUpdate` and `EndUpdate` in your window update routine.

### **I turned off a part of the movie, but a little of it still plays! How do I stop that?**

Use `SetMovieActiveSegment` to assure that samples you want to ignore never get played.

### **How do I clip movies in odd-shaped windows?**

When using the Movie Controller, call `MCSetClip`. The `MovieController` needs to use the Movie display clip first before it passes it on to `SetMovieDisplayClipRgn`.

If you are using the Movie Controller with `MCIsPlayerEvent`, it will henceforth take care of setting the clip and doing the updating for you. Just call:

```
BeginUpdate(movieWindow);  
EraseRect(&movieWindow->portRect);  
EndUpdate(movieWindow);
```

If you are using the Movie Controller with `MCDraw` so you can handle the update events yourself, clip to `MCGetWindowRgn()`.

If you are using the Movie Toolbox directly for everything, then call `Set/GetMovieDisplayBoundsRgn()` to set and get the clip.

### **How do I change the movie screen background?**

This is an issue when you paste together two movies of different screen dimensions, so when the smaller part plays there is dead space in the background. By default, QuickTime fills the dead space with white. Call `SetMovieCoverProc` to specify a routine that you wrote to draw whatever you like in the dead space.

### **How do I find the next frame/edit point/key frame?**

Use `GetMovieNextInterestingTime` or `GetTrackNextInterestingTime`. It can find any of the above for you.