Movie Controller Reference

QuickTime > Movie Basics

2006-05-23

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

Movie Controller Reference

Framework:	Frameworks/QuickTime.framework
Declared in	HIMovieView.h Movies.h OSTypes.h

Overview

Movie controllers provide a user interface for playing and editing movies, eliminating much of the complexity of working with movies. Movie controllers are implemented in QuickTime as components. This allows customized controllers to be plugged in to QuickTime for use by your application.

Functions by Task

Associating Movies With Controllers

`MCGetIndMovie` (page 30)

Lets your application to retrieve the movie reference for a movie that is associated with a movie controller.

`MCNewAttachedController` (page 39)

Associates a specified movie with a movie controller.

`MCSetMovie` (page 50)

Associates a movie with a specified movie controller.

Customizing Event Processing

`MCActivate` (page 16)

Lets a controller respond to activate, deactivate, suspend, and resume events.

`MCClick` (page 19)

Lets a controller respond when the user clicks in a movie controller window.

`MCDraw` (page 22)

Responds to an update event.

`MCIdle` (page 34)

Performs idle processing for a movie controller.

MCKey (page 38)
Handles keyboard events for a movie controller.

Editing Movies With a Controller

MCClear (page 19)
Removes the current movie selection from the movie associated with a specified controller.

MCCopy (page 20)
Returns a copy of the current movie selection from the movie associated with a specified controller.

MCCut (page 21)
Returns a copy of the current movie selection from the movie associated with a specified controller and then removes the current movie selection from the source movie.

MCEnableEditing (page 24)
Enables and disables editing for a movie controller.

MCGetMenuString (page 31)
Retrieves the text string for a movie controller menu command.

MCIsEditingEnabled (page 36)
Determines whether editing is currently enabled for a movie controller.

MCPaste (page 40)
Inserts a specified movie at the current movie time in the movie associated with a specified controller.

MCSetUpEditMenu (page 51)
Correctly highlights and names the items in your application's Edit menu.

MCUndo (page 53)
Lets your application discard the effects of the most recent edit operation.

Getting and Setting Movie Controller Time

MCGetCurrentTime (page 28)
Obtains the time value represented by the indicator on the movie controller's slider.

MCSetDuration (page 49)
Lets your application set a controller's duration in the case where a controller does not have a movie associated with it.

Handling Movie Events

MCGetControllerInfo (page 27)
Determines the current status of a movie controller and its associated movie, for menu highlighting.

MCIsPlayerEvent (page 36)
Handles all events for a movie controller.

MCPtInController (page 41)
Reports whether a point is in the control area of a movie.

Managing Controller Attributes

`MCDrawBadge` (page 23)
Displays a controller's badge.

`MCGetClip` (page 25)
Obtains information describing a movie controller's clipping regions.

`MCGetControllerBoundsRect` (page 25)
Returns a movie controller's boundary rectangle.

`MCGetControllerBoundsRgn` (page 26)
Returns the actual region occupied by the controller and its movie.

`MCGetControllerPort` (page 28)
Returns a movie controller's color graphics port.

`MCGetVisible` (page 32)
Returns a value that indicates whether or not a movie controller is visible.

`MCGetWindowRgn` (page 33)
Determines the window region that is actually in use by a controller and its movie.

`MCIsControllerAttached` (page 35)
Returns a value that indicates whether a movie controller is attached to its movie.

`MCPositionController` (page 40)
Controls the position of a movie and its controller on the computer display.

`MCSetClip` (page 45)
Lets you set a movie controller's clipping region.

`MCSetControllerAttached` (page 46)
Lets your application control whether a movie controller is attached to its movie or detached from it.

`MCSetControllerBoundsRect` (page 47)
Lets you change the position and size of a movie controller.

`MCSetControllerPort` (page 48)
Lets your application set the graphics port for a movie controller.

`MCSetVisible` (page 52)
Lets your application control the visibility of a movie controller.

Movie Controller Action Functions

`MCDoAction` (page 22)
Invokes a movie controller component and makes it perform a specified action.

`MCMovieChanged` (page 38)
Informs a movie controller component that your application has used the Movie Toolbox to change the characteristics of its associated movie.

`MCSetActionFilterWithRefCon` (page 44)
Establishes an action filter function for a movie controller.

Working With The Idle Manager

MCSetIdleManager (page 50)
Lets a movie controller component report its idling needs.

Supporting Functions

DisposeMCActionFilterUPP (page 11)
Disposes of a MCActionFilterUPP pointer.

DisposeMCActionFilterWithRefConUPP (page 11)
Disposes of a MCActionFilterWithRefConUPP pointer.

HIMovieViewChangeAttributes (page 12)
Changes the views attributes.

HIMovieViewCreate (page 13)
Creates an HIMovieView object.

HIMovieViewGetAttributes (page 13)
Returns the view's current attributes.

HIMovieViewGetControllerBarSize (page 14)
Returns the size of the visible movie controller bar.

HIMovieViewGetMovie (page 14)
Returns the view's current movie.

HIMovieViewGetMovieController (page 14)
Returns the view's current movie controller.

HIMovieViewPause (page 15)
Pauses the view's current movie.

HIMovieViewPlay (page 15)
Plays the view's current movie.

HIMovieViewSetMovie (page 16)
Sets the view's current movie.

MCAddMovieSegment (page 17)
Undocumented

MCAdjustCursor (page 18)
Undocumented

MCGetDoActionsProc (page 29)
Retrieves the DoMCActionProc callback attached to a movie controller.

MCGetInterfaceElement (page 30)
Gets the interface element of a specified type for a movie controller.

MCInvalidate (page 34)
Invalidates a region of a movie controller's display.

MCRemoveAllMovies (page 42)
Removes all movies associated with a controller.

MCRemoveAMovie (page 43)
Removes one movie from a multi-movie controller.

`MCRemoveMovie` (page 43)
: Removes a movie from a movie controller.

`MCSetActionFilter` (page 44)
: Sets the MCActionFilterProc callback for a movie controller.

`MCSetControllerCapabilities` (page 47)
: Undocumented

`MCTrimMovieSegment` (page 52)
: Undocumented

`NewMCActionFilterUPP` (page 54)
: Allocates a Universal Procedure Pointer for the MCActionFilterProc callback.

`NewMCActionFilterWithRefConUPP` (page 54)
: Allocates a Universal Procedure Pointer for the MCActionFilterWithRefConProc callback.

Functions

DisposeMCActionFilterUPP

Disposes of a MCActionFilterUPP pointer.

```
void DisposeMCActionFilterUPP (
   MCActionFilterUPP userUPP
);
```

Parameters

userUPP
: A `MCActionFilterUPP` pointer. See `Universal Procedure Pointers`.

Return Value

You can access this function's error returns through `GetMoviesError` and `GetMoviesStickyError`.

Version Notes

Introduced in QuickTime 4.1.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`Movies.h`

DisposeMCActionFilterWithRefConUPP

Disposes of a MCActionFilterWithRefConUPP pointer.

```
void DisposeMCActionFilterWithRefConUPP (
   MCActionFilterWithRefConUPP userUPP
);
```

Parameters

userUPP

A `MCActionFilterWithRefConUPP` pointer. See `Universal Procedure Pointers`.

Return Value

You can access this function's error returns through `GetMoviesError` and `GetMoviesStickyError`.

Version Notes

Introduced in QuickTime 4.1.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

QTKitMovieShuffler

Declared In

`Movies.h`

HIMovieViewChangeAttributes

Changes the views attributes.

```
OSStatus HIMovieViewChangeAttributes (
   HIViewRef inView,
   OptionBits inAttributesToSet,
   OptionBits inAttributesToClear
);
```

Parameters

inView

The `HIMovieView`.

inAttributesToSet

Attributes to set.

inAttributesToClear

Attributes to clear.

Return Value

An error code. Returns `noErr` if there is no error.

Discussion

Setting an attribute takes precedence over clearing the attribute.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

SimpleHIMovieViewPlayer

Declared In

`HIMovieView.h`

HIMovieViewCreate

Creates an HIMovieView object.

```
OSStatus HIMovieViewCreate (
   Movie inMovie,
   OptionBits inAttributes,
   HIViewRef *outMovieView
);
```

Parameters

inMovie

Initial movie to view; may be NULL.

inAttributes

Initial `HIMovieView` attributes.

outMovieView

Points to variable to receive new `HIMovieView`.

Return Value

Undocumented.

Discussion

If successful, the created view will have a single retain count.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`HIMovieView.h`

HIMovieViewGetAttributes

Returns the view's current attributes.

```
OptionBits HIMovieViewGetAttributes (
   HIViewRef inView
);
```

Parameters

inView

The `HIMovieView`.

Return Value

Undocumented.

Discussion

The view's current attributes are returned.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`HIMovieView.h`

HIMovieViewGetControllerBarSize

Returns the size of the visible movie controller bar.

```
HISize HIMovieViewGetControllerBarSize (
   HIViewRef inView
);
```

Parameters

inView

The HIMovieView.

Return Value
Undocumented.

Discussion
The size of the visible movie controller bar is returned.

Availability
Available in Mac OS X v10.3 and later.

Related Sample Code
SimpleHIMovieViewPlayer

Declared In
HIMovieView.h

HIMovieViewGetMovie

Returns the view's current movie.

```
Movie HIMovieViewGetMovie (
   HIViewRef inView
);
```

Parameters

inView

The HIMovieView.

Return Value
Undocumented.

Discussion
The view's current movie is returned.

Availability
Available in Mac OS X v10.3 and later.

Declared In
HIMovieView.h

HIMovieViewGetMovieController

Returns the view's current movie controller.

```
MovieController HIMovieViewGetMovieController (
   HIViewRef inView
);
```

Parameters

inView

The HIMovieView.

Return Value

Undocumented.

Discussion

The view's current movie controller is returned.

Availability

Available in Mac OS X v10.3 and later.

Declared In

HIMovieView.h

HIMovieViewPause

Pauses the view's current movie.

```
OSStatus HIMovieViewPause (
   HIViewRef movieView
);
```

Parameters

movieView

The movie view.

Return Value

An error code. Returns noErr if there is no error.

Discussion

This is a convenience routine to pause the view's current movie. If the movie is already paused, this function does nothing.

Availability

Available in Mac OS X v10.3 and later.

Declared In

HIMovieView.h

HIMovieViewPlay

Plays the view's current movie.

```
OSStatus HIMovieViewPlay (
   HIViewRef movieView
);
```

Parameters

movieView

The movie view.

Return Value

An error code. Returns `noErr` if there is no error.

Discussion

This is a convenience routine to play the view's current movie. If the movie is already playing, this function does nothing.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`HIMovieView.h`

HIMovieViewSetMovie

Sets the view's current movie.

```
OSStatus HIMovieViewSetMovie (
   HIViewRef inView,
   Movie inMovie
);
```

Parameters

inView

The `HIMovieView`.

inMovie

The new movie to display.

Return Value

An error code. Returns `noErr` if there is no error.

Discussion

This routine sets the view's current movie.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

SimpleHIMovieViewPlayer

Declared In

`HIMovieView.h`

MCActivate

Lets a controller respond to activate, deactivate, suspend, and resume events.

```
ComponentResult MCActivate (
   MovieController mc,
   WindowRef w,
   Boolean activate
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

w

A pointer to the window in which the event has occurred.

activate

The nature of the `event`. Set this parameter to TRUE for activate and resume events. Set it to FALSE for deactivate and suspend events.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

CarbonQTGraphicImport

MakeEffectMovie

qtbigscreen

qtbigscreen.win

QTCarbonShell

Declared In

`Movies.h`

MCAddMovieSegment

Undocumented

```
ComponentResult MCAddMovieSegment (
   MovieController mc,
   Movie srcMovie,
   Boolean scaled
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

srcMovie

The source movie. Your application obtains this movie identifier from such functions as `NewMovie`, `NewMovieFromFile`, or `NewMovieFromHandle`.

scaled

Undocumented

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 5.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`Movies.h`

MCAdjustCursor

Undocumented

```
ComponentResult MCAdjustCursor (
   MovieController mc,
   WindowRef w,
   Point where,
   long modifiers
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

w

A pointer to the window in which the cursor is located.

where

The location of the cursor. This value is expressed in the local coordinates of the window specified by the `w` parameter.

modifiers

The cursor form (see below). See these constants:

`kQTCursorOpenHand`

`kQTCursorClosedHand`

`kQTCursorPointingHand`

`kQTCursorRightArrow`

`kQTCursorLeftArrow`

`kQTCursorDownArrow`

`kQTCursorUpArrow`

`kQTCursorIBeam`

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`Movies.h`

MCClear

Removes the current movie selection from the movie associated with a specified controller.

```
ComponentResult MCClear (
   MovieController mc
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

MakeEffectMovie

mfc.win

qteffects.win

simpleplayersdi.win

vrmovies.win

Declared In

`Movies.h`

MCClick

Lets a controller respond when the user clicks in a movie controller window.

```
ComponentResult MCClick (
   MovieController mc,
   WindowRef w,
   Point where,
   long when,
   long modifiers
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

w

A pointer to the window in which the event has occurred.

where

The location of the click. This value is expressed in the local coordinates of the window specified by the `w` parameter. Your application must convert this value from the global coordinates returned in the `EventRecord` structure.

when

Indicates when the user pressed the mouse button. You obtain this value from the `EventRecord` structure.

modifiers

Specifies modifier flags for the event. You obtain this value from the `EventRecord` structure.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

QTCarbonShell

Declared In

`Movies.h`

MCCopy

Returns a copy of the current movie selection from the movie associated with a specified controller.

```
Movie MCCopy (
   MovieController mc
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

Return Value
A copy of the movie.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
MakeEffectMovie

mfc.win

qteffects.win

simpleplayersdi.win

vrmovies.win

Declared In
`Movies.h`

MCCut

Returns a copy of the current movie selection from the movie associated with a specified controller and then removes the current movie selection from the source movie.

```
Movie MCCut (
   MovieController mc
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

Return Value
A copy of the current movie selection.

Discussion
Your application is responsible for the returned movie. `MCCut` returns a movie containing the current selection from the movie associated with the specified controller. If the user has not made a selection, the returned movie reference is set to `NIL`.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
MakeEffectMovie

mfc.win

qteffects.win

simpleplayersdi.win

vrmovies.win

Declared In
`Movies.h`

MCDoAction

Invokes a movie controller component and makes it perform a specified action.

```
ComponentResult MCDoAction (
   MovieController mc,
   short action,
   void *params
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

action

The action to be taken. See `Movie Controller Actions`.

params

A pointer to the parameter data appropriate to the action. See `Movie Controller Actions` for information about the parameters required for each supported action.

Return Value
You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
MakeEffectMovie

qtcontroller

qtcontroller.win

qtinfo

qtstreamsplicer.win

Declared In
`Movies.h`

MCDraw

Responds to an update event.

```
ComponentResult MCDraw (
   MovieController mc,
   WindowRef w
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

w

A pointer to the window in which the update event has occurred.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

CarbonQTGraphicImport

QTCarbonShell

SimpleVideoOut

Declared In

`Movies.h`

MCDrawBadge

Displays a controller's badge.

```
ComponentResult MCDrawBadge (
   MovieController mc,
   RgnHandle movieRgn,
   RgnHandle *badgeRgn
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

movieRgn

The boundary region of the controller's movie.

badgeRgn

A pointer to a region that is to receive information about the location of the badge. The movie controller returns the region where the badge is displayed. If you are not interested in this information, you may set this parameter to `NIL`. Your application must dispose of this handle.

Return Value
You can access Movie Toolbox error returns through GetMoviesError and GetMoviesStickyError, as well as in the function result. See Error Codes.

Discussion
This function places the badge in an appropriate location based on the location of the controller's movie. MCDrawBadge can be useful in circumstances where you are using a movie controller component but do not want to incur the overhead of having the QuickTime movie in memory all the time. This function allows you to display the badge without having to display the movie. In addition, you can use the badge region to perform mouse-down event testing.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
Movies.h

MCEnableEditing

Enables and disables editing for a movie controller.

```
ComponentResult MCEnableEditing (
   MovieController mc,
   Boolean enabled
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from OpenComponent or OpenDefaultComponent, or from NewMovieController.

enabled

Specifies whether to enable or disable editing for the controller. Set this parameter to TRUE to enable editing; set it to FALSE to disable editing.

Return Value
You can access Movie Toolbox error returns through GetMoviesError and GetMoviesStickyError, as well as in the function result. See Error Codes.

Discussion
Once editing is enabled for a controller, the user may edit the movie associated with the controller.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
MakeEffectMovie

mfc.win

qtstreamsplicer

qtstreamsplicer.win

vrmovies.win

Declared In

`Movies.h`

MCGetClip

Obtains information describing a movie controller's clipping regions.

```
ComponentResult MCGetClip (
   MovieController mc,
   RgnHandle *theClip,
   RgnHandle *movieClip
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

theClip

A pointer to a field that is to receive a handle to the clipping region of the entire movie controller. You must dispose of this region when you are done with it. If you are not interested in this information, set this parameter to `NIL`.

movieClip

A pointer to a field that is to receive a handle to the clipping region of the controller's movie. You must dispose of this region when you are done with it. If you are not interested in this information, set this parameter to `NIL`.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`Movies.h`

MCGetControllerBoundsRect

Returns a movie controller's boundary rectangle.

```
ComponentResult MCGetControllerBoundsRect (
   MovieController mc,
   Rect *bounds
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

bounds

A pointer to a `Rect` structure that is to receive the coordinates of the movie controller's boundary rectangle. If there is insufficient screen space to display the controller, the function may return an empty structure.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

MakeEffectMovie

qtbigscreen

qtbigscreen.win

vrscript

vrscript.win

Declared In

`Movies.h`

MCGetControllerBoundsRgn

Returns the actual region occupied by the controller and its movie.

```
RgnHandle MCGetControllerBoundsRgn (
   MovieController mc
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

Return Value

A handle to a `MacRegion` structure that reflects the size, shape, and location of the controller. Your application must dispose of this structure.

Discussion

As with `MCGetControllerBoundsRect` (page 25), this function returns a region even if the controller is hidden. Some movie controllers may not be rectangular in shape. If the movie is not attached to its controller, the boundary region encloses only the control portion of the controller.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

QTCarbonShell

Declared In

`Movies.h`

MCGetControllerInfo

Determines the current status of a movie controller and its associated movie, for menu highlighting.

```
ComponentResult MCGetControllerInfo (
   MovieController mc,
   long *someFlags
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

someFlags

A pointer to flags (see below) that specify the current status and capabilities of the controller. More than one flag may be set to 1. See these constants:

`mcInfoUndoAvailable`

`mcInfoCutAvailable`

`mcInfoCopyAvailable`

`mcInfoPasteAvailable`

`mcInfoClearAvailable`

`mcInfoHasSound`

`mcInfoIsPlaying`

`mcInfoIsLooping`

`mcInfoIsInPalindrome`

`mcInfoEditingEnabled`

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Discussion

You can use the information returned by this function to control your application's menu highlighting.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
MakeEffectMovie

qtbigscreen

qtbigscreen.win

qteffects.win

vrmovies.win

Declared In
`Movies.h`

MCGetControllerPort

Returns a movie controller's color graphics port.

```
CGrafPtr MCGetControllerPort (
   MovieController mc
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

Return Value
A pointer to the movie controller's `CGrafPort` structure.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
CompressMovies

DigitizerShell

DragAndDrop Shell

mdiplayer.win

MovieGWorlds

Declared In
`Movies.h`

MCGetCurrentTime

Obtains the time value represented by the indicator on the movie controller's slider.

```
TimeValue MCGetCurrentTime (
   MovieController mc,
   TimeScale *scale
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

scale

A pointer to a field that is to receive the time scale for the controller.

Return Value

A time value containing the time shown by the indicator on the movie controller's slider.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`Movies.h`

MCGetDoActionsProc

Retrieves the DoMCActionProc callback attached to a movie controller.

```
ComponentResult MCGetDoActionsProc (
   MovieController mc,
   DoMCActionUPP *doMCActionProc,
   long *doMCActionRefCon
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

doMCActionProc

A pointer to a `DoMCActionProc` callback.

doMCActionRefCon

A reference constant that is passed to your callback. This parameter may point to a data structure containing information your function needs.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In
`Movies.h`

MCGetIndMovie

Lets your application to retrieve the movie reference for a movie that is associated with a movie controller.

```
Movie MCGetIndMovie (
   MovieController mc,
   short index
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

index

Index for the movie. When set to 0, this call duplicates the action of the previous call to this function.

Return Value
The movie identifier for the movie that is assigned to the specified controller, or `NIL` if there is no movie assigned to the controller.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`Movies.h`

MCGetInterfaceElement

Gets the interface element of a specified type for a movie controller.

```
ComponentResult MCGetInterfaceElement (
   MovieController mc,
   MCInterfaceElement whichElement,
   void *element
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

whichElement

A constant (see below) that identifies the interface element type. See these constants:

- kMCIEEnabledButtonPicture
- kMCIEDisabledButtonPicture
- kMCIEDepressedButtonPicture
- kMCIEEnabledSizeBoxPicture
- kMCIEDisabledSizeBoxPicture
- kMCIEEnabledUnavailableButtonPicture
- kMCIEDisabledUnavailableButtonPicture
- kMCIESoundSlider
- kMCIESoundThumb

element

A pointer to the element type.

Return Value

You can access Movie Toolbox error returns through GetMoviesError and GetMoviesStickyError, as well as in the function result. See Error Codes.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

Movies.h

MCGetMenuString

Retrieves the text string for a movie controller menu command.

```
ComponentResult MCGetMenuString (
   MovieController mc,
   long modifiers,
   short item,
   Str255 aString
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from OpenComponent or OpenDefaultComponent, or from NewMovieController.

modifiers

The current modifiers from the mouse-down or key-down event to which you are responding.

item

One of the movie controller Edit menu constants (see below). See these constants:

`mcMenuUndo`

`mcMenuCut`

`mcMenuCopy`

`mcMenuPaste`

`mcMenuClear`

aString

On entry, pass a string of type Str255; on exit, this string is set to the text of the menu item specified by the `item` parameter.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Discussion

`MCGetMenuString` is used by `MCSetUpEditMenu` (page 51).

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

Movie From DataRef

qteffects.win

qtgraphics.win

vrbackbuffer

vrmovies.win

Declared In

`Movies.h`

MCGetVisible

Returns a value that indicates whether or not a movie controller is visible.

```
ComponentResult MCGetVisible (
   MovieController mc
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

Return Value

If the controller is visible, the function result is set to 1. If the controller is not showing, the function result is set to 0. You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`. See `Error Codes`.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
MakeEffectMovie

qtbigscreen

qtbigscreen.win

vrscript

vrscript.win

Declared In
`Movies.h`

MCGetWindowRgn

Determines the window region that is actually in use by a controller and its movie.

```
RgnHandle MCGetWindowRgn (
   MovieController mc,
   WindowRef w
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

w

A pointer to the window in which the movie controller and its movie are displayed, if the control portion of the controller is attached to the movie. If the controller is detached and in a separate window from the movie, specify one of the windows.

Return Value
A handle to the `MacRegion` structure for the window that is actually in use. Your application must dispose of this structure.

Discussion
The region returned by this function contains only the visible portions of the controller and its movie.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
DigitizerShell

MCComponent

MovieBrowser

MovieGWorlds

vrscript.win

Declared In
`Movies.h`

MCIdle

Performs idle processing for a movie controller.

```
ComponentResult MCIdle (
   MovieController mc
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

Return Value
You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
CarbonQTGraphicImport

MakeEffectMovie

MovieBrowser

qtshellCEvents.win

vrscript.win

Declared In
`Movies.h`

MCInvalidate

Invalidates a region of a movie controller's display.

```
ComponentResult MCInvalidate (
   MovieController mc,
   WindowRef w,
   RgnHandle invalidRgn
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

w

A pointer to the window in which the movie controller and its movie are displayed, if the control portion of the controller is attached to the movie. If the controller is detached and in a separate window from the movie, specify one of the windows.

invalidRgn

A handle to a `MacRegion` structure that defines a region to invalidate.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

QTCarbonShell

vrscript

vrscript.win

Declared In

`Movies.h`

MCIsControllerAttached

Returns a value that indicates whether a movie controller is attached to its movie.

```
ComponentResult MCIsControllerAttached (
   MovieController mc
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

Return Value

If the controller is attached, the returned value is set to 1. If the controller is not attached, the returned value is set to 0. You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

MakeEffectMovie

qtbigscreen

qtbigscreen.win

qtcontroller

qtcontroller.win

Declared In
`Movies.h`

MCIsEditingEnabled

Determines whether editing is currently enabled for a movie controller.

```
long MCIsEditingEnabled (
   MovieController mc
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

Return Value
Returns 1 if editing is enabled; set to 0 if editing is disabled or if the controller component does not support editing.

Discussion
Once editing is enabled for a controller, the user may edit the movie associated with the controller.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`Movies.h`

MCIsPlayerEvent

Handles all events for a movie controller.

```
ComponentResult ADD_MEDIA_BASENAME() MCIsPlayerEvent
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

e

A pointer to the current `EventRecord` structure.

Return Value
A long integer indicating whether the movie controller component handled the event. The component sets this long integer to 1 if it handled the event. Your application should then skip the rest of its event loop and wait for the next event. The return value is 0 otherwise. Your application must then handle the event as part of its normal event processing.

Discussion

The movie controller component does everything necessary to support the movie controller and its associated movie. For example, the component calls `MoviesTask` for each movie. The movie controller component also handles suspend and resume events. It treats suspend events as deactivate requests and resume events as activate requests.

The following sample code shows how to convert Windows messages to Macintosh events and then pass those events to the QuickTime movie controller, using this function:

```
// MCIsPlayerEvent coding example
// See "Discovering QuickTime," page 240
MovieController   mc;                   // Movie controller for movie
LRESULT CALLBACK WndProc
        (HWND       hwnd,               // Handle to window
         UINT       iMsg,               // Message type
         WPARAM     wParam,             // Message-dependent parameter
         LPARAM     lParam)             // Message-dependent parameter
{
    MSG             msg;                // Windows message structure
    EventRecord     er;                 // Macintosh event record
    DWORD           dwPos;              // Mouse coordinates of message
    msg.hwnd        =hwnd;             // Window handle
    msg.message     =iMsg;             // Message type
    msg.wParam      =wParam;           // Word-length parameter
    msg.lParam      =lParam;           // Long-word parameter

    msg.time =GetMessageTime();        // Get time of message
    dwPos  =GetMessagePos();           // Get mouse position
    msg.pt.x =LOWORD(dwPos);           // Extract x coordinate
    msg.pt.y =HIWORD(dwPos);           // Extract y coordinate

    WinEventToMacEvent(&msg, &er);      // Convert to event
    MCIsPlayerEvent(mc, &er);           // Pass event to QuickTime

    switch (iMsg) {                     // Dispatch on message type

        . . .       // Handle message according to type

    }  // end switch (iMsg)

}  // end WndProc
```

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

MakeEffectMovie

Movie From DataRef

qtbigscreen

qtwiredactions

vrbackbuffer

Declared In
`Movies.h`

MCKey

Handles keyboard events for a movie controller.

```
ComponentResult MCKey (
   MovieController mc,
   SInt8 key,
   long modifiers
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

key

The keystroke. You obtain this value from the event structure.

modifiers

Modifier flags for the event. You obtain this value from the `EventRecord` structure.

Return Value
You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`Movies.h`

MCMovieChanged

Informs a movie controller component that your application has used the Movie Toolbox to change the characteristics of its associated movie.

```
ComponentResult MCMovieChanged (
   MovieController mc,
   Movie m
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

m

The movie that has been changed.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

qtinfo

qtinfo.win

qttext

qttext.win

SimpleVideoOut

Declared In

`Movies.h`

MCNewAttachedController

Associates a specified movie with a movie controller.

```
ComponentResult MCNewAttachedController (
   MovieController mc,
   Movie theMovie,
   WindowRef w,
   Point where
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

theMovie

The movie to be associated with the movie controller.

w

A pointer to the window in which the movie is to be displayed. The movie controller component sets the movie's graphics world to match this window. If you set the `w` parameter to `NIL`, the component uses the current window.

where

The upper-left corner of the movie within the window specified by the `w` parameter. The movie controller component uses the movie's boundary `Rect` structure to determine the size of the movie. `GetMovieBox` returns this structure.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`Movies.h`

MCPaste

Inserts a specified movie at the current movie time in the movie associated with a specified controller.

```
ComponentResult MCPaste (
   MovieController mc,
   Movie srcMovie
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

srcMovie

The movie to be inserted into the current selection in the movie associated with the movie controller specified by the `mc` parameter. If you set this parameter to `NIL`, the movie controller component retrieves the source movie from the scrap.

Return Value
You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
MakeEffectMovie

mfc.win

qteffects.win

simpleplayersdi.win

vrmovies.win

Declared In
`Movies.h`

MCPositionController

Controls the position of a movie and its controller on the computer display.

```
ComponentResult MCPositionController (
   MovieController mc,
   const Rect *movieRect,
   const Rect *controllerRect,
   long someFlags
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

movieRect

A pointer to a `Rect` structure that specifies the coordinates of the movie's boundary `Rect` structure.

controllerRect

A pointer to a `Rect` structure that specifies the coordinates of the controller's boundary `Rect` structure. The movie controller component always centers the control portion of the controller inside this rectangle. The movie controller component only uses this parameter when the control portion of the controller is detached from the movie. If you are working with an attached controller, you can set this parameter to `NIL`.

someFlags

Flags (see below) that control how the movie is drawn. If you set these flags to 0, the movie controller component centers the movie in the rectangle specified by `movieRect` and scales the movie to fit in that rectangle. See these constants:

`mcTopLeftMovie`

`mcScaleMovieToFit`

`mcPositionDontInvalidate`

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

MakeEffectMovie

Movie From DataRef

MovieBrowser

qteffects.win

vrmovies.win

Declared In

`Movies.h`

MCPtInController

Reports whether a point is in the control area of a movie.

```
ComponentResult MCPtInController (
   MovieController mc,
   Point thePt,
   Boolean *inController
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

thePt

The point to be checked. This point must be passed in local coordinates to the controller's window. This point is checked only against the movie controller's controls, not the movie itself.

inController

Returns true if the point is in the controller; false if it is not.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Discussion

While you could always determine if a point is contained in a movie, using `PtInMovie`, the `MCPtInController` function allows you to determine if a point is in the control area of a movie.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`Movies.h`

MCRemoveAllMovies

Removes all movies associated with a controller.

```
ComponentResult MCRemoveAllMovies (
   MovieController mc
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In
`Movies.h`

MCRemoveAMovie

Removes one movie from a multi-movie controller.

```
ComponentResult MCRemoveAMovie (
   MovieController mc,
   Movie m
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

m

The movie for this operation. Your application obtains this movie identifier from such functions as `NewMovie`, `NewMovieFromFile`, or `NewMovieFromHandle`.

Return Value
You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`Movies.h`

MCRemoveMovie

Removes a movie from a movie controller.

```
ComponentResult MCRemoveMovie (
   MovieController mc
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

Return Value
You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`Movies.h`

MCSetActionFilter

Sets the MCActionFilterProc callback for a movie controller.

```
ComponentResult MCSetActionFilter (
   MovieController mc,
   MCActionFilterUPP blob
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

blob

A Universal Procedure Pointer to an `MCActionFilterProc` callback.

Return Value
You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`Movies.h`

MCSetActionFilterWithRefCon

Establishes an action filter function for a movie controller.

```
ComponentResult MCSetActionFilterWithRefCon (
   MovieController mc,
   MCActionFilterWithRefConUPP blob,
   long refCon
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

blob

A pointer to your `MCActionFilterWithRefConProc` callback. Set this parameter to `NIL` to remove an existing callback.

refCon

A reference constant value. The movie controller component passes this reference constant to your action filter callback each time it calls it. Use this parameter to point to a data structure containing any information your callback needs.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Discussion

The movie controller component calls your action filter function each time the component receives an action for its movie controller. Your filter function is then free to handle the action or to refer it back to the movie controller component. If you refer it back to the movie controller component, the component handles the action.

If your filter function handles an action, you can handle the action in any way you desire. For example, your filter function could change the operation of movie controller buttons. More commonly, applications use the action filter function to monitor actions of the controller. For instance, your filter function might enable you to find out when the user clicks the play button, so that your application can enable appropriate menu selections. Alternatively, you can use the filter function to detect when the user resizes the movie.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

MakeEffectMovie

Movie From DataRef

qteffects.win

QTKitMovieShuffler

vrmovies.win

Declared In

`Movies.h`

MCSetClip

Lets you set a movie controller's clipping region.

```
ComponentResult MCSetClip (
   MovieController mc,
   RgnHandle theClip,
   RgnHandle movieClip
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

theClip

A handle to a region that defines the controller's clipping region. This clipping region affects the entire movie controller and its movie, including the controller's badge and associated controls. Set this parameter to `NIL` to clear the controller's clipping region.

movieClip

A handle to a region that defines the clipping region of the controller's movie. This clipping region affects only the movie and the badge, not the movie controller. Set this parameter to `NIL` to clear the movie clipping region.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`Movies.h`

MCSetControllerAttached

Lets your application control whether a movie controller is attached to its movie or detached from it.

```
ComponentResult MCSetControllerAttached (
   MovieController mc,
   Boolean attach
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

attach

The action for this function. Set the `attach` parameter to TRUE to cause the controller to be attached to its movie. Set this parameter to FALSE to detach the controller from its movie.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

MakeEffectMovie

qtbigscreen

qtbigscreen.win

qtskins.win

SimpleVideoOut

Declared In
`Movies.h`

MCSetControllerBoundsRect

Lets you change the position and size of a movie controller.

```
ComponentResult MCSetControllerBoundsRect (
   MovieController mc,
   const Rect *bounds
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

bounds

A pointer to a `Rect` structure that contains the new boundary `Rect` structure for the movie controller.

Return Value
See `Error Codes`. Returns a value of `controllerBoundsNotExact` if the boundary rectangle has been changed but does not correspond to the rectangle you specified. In this case, the new boundary rectangle is always smaller than the requested rectangle. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
CompressMovies

DigitizerShell

DragAndDrop Shell

MovieGWorlds

qtbigscreen

Declared In
`Movies.h`

MCSetControllerCapabilities

Undocumented

```
ComponentResult MCSetControllerCapabilities (
   MovieController mc,
   long flags,
   long flagsMask
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

flags

Undocumented

flagsMask

Undocumented

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 6.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`Movies.h`

MCSetControllerPort

Lets your application set the graphics port for a movie controller.

```
ComponentResult MCSetControllerPort (
   MovieController mc,
   CGrafPtr gp
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

gp

A pointer to the new graphics port for the movie controller. Set this parameter to `NIL` to use the current graphics port.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Discussion

Movie controller components use `MCSetControllerPort` each time you create a new movie controller. Hence, your component must be set to a valid port before creating a new movie controller. You can use this function to place a movie and its associated movie controller in different graphics ports. If you are using an attached controller, both the controller and the movie's graphics ports are changed. If you are using a detached controller, this function changes only the graphics port of the control portion of the controller. You must use `SetMovieGWorld` followed by `MCMovieChanged` (page 38) to change other portions.

```
pascal ComponentResult MCSetControllerPort (MovieController mc,
                                             CGrafPtr gp);
```

Special Considerations

The movie controller component may use the foreground and background colors from the graphics port at the time this function is called to colorize the movie controller.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

DragAndDrop Shell

qtbigscreen

qtbigscreen.win

QTCarbonShell

vrscript

Declared In

`Movies.h`

MCSetDuration

Lets your application set a controller's duration in the case where a controller does not have a movie associated with it.

```
ComponentResult MCSetDuration (
   MovieController mc,
   TimeValue duration
);
```

Parameters

`mc`

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

`duration`

The new duration for the movie. This duration value must be in the controller's time scale.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In
`Movies.h`

MCSetIdleManager

Lets a movie controller component report its idling needs.

```
ComponentResult MCSetIdleManager (
   MovieController mc,
   IdleManager im
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

im

A pointer to an opaque data structure that belongs to the Mac OS Idle Manager. You get this pointer by calling `QTIdleManagerOpen`.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 6.

Availability
Available in Mac OS X v10.2 and later.

Declared In
`Movies.h`

MCSetMovie

Associates a movie with a specified movie controller.

```
ComponentResult MCSetMovie (
   MovieController mc,
   Movie theMovie,
   WindowRef movieWindow,
   Point where
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

theMovie

The movie to be associated with the movie controller. Set this value to `NIL` to remove the movie from the controller.

movieWindow

The window in which the movie is to be displayed. The movie controller component sets the movie's graphics world to match this window. If you set the `w` parameter to `NIL`, the component uses the current window.

where

The upper-left corner of the movie within the window specified by the `movieWindow` parameter. The movie controller component uses the movie's boundary `Rect` structure to determine the size of the movie. `GetMovieBox` returns this structure.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

MovieBrowser

Declared In

`Movies.h`

MCSetUpEditMenu

Correctly highlights and names the items in your application's Edit menu.

```
ComponentResult MCSetUpEditMenu (
   MovieController mc,
   long modifiers,
   MenuRef mh
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

modifiers

The current modifiers from the mouse-down or key-down event to which you are responding.

mh

A menu handler for your current Edit menu. The first six items in your Edit menu should be the standard editing commands: Undo, a blank line, Cut, Copy, Paste, and Clear.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

Movie From DataRef

qteffects.win

qtgraphics.win

vrbackbuffer

vrmovies.win

Declared In

`Movies.h`

MCSetVisible

Lets your application control the visibility of a movie controller.

```
ComponentResult MCSetVisible (
   MovieController mc,
   Boolean visible
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

visible

Set to TRUE to cause the controller to be visible, or FALSE to make the controller invisible.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

MakeEffectMovie

MovieBrowser

qtbigscreen

qtbigscreen.win

QTCarbonShell

Declared In

`Movies.h`

MCTrimMovieSegment

Undocumented

```
ComponentResult MCTrimMovieSegment (
   MovieController mc
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 5.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`Movies.h`

MCUndo

Lets your application discard the effects of the most recent edit operation.

```
ComponentResult MCUndo (
   MovieController mc
);
```

Parameters

mc

The movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

Return Value

You can access Movie Toolbox error returns through `GetMoviesError` and `GetMoviesStickyError`, as well as in the function result. See `Error Codes`.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

MakeEffectMovie

mfc.win

qteffects.win

simpleplayersdi.win

vrmovies.win

Declared In

`Movies.h`

NewMCActionFilterUPP

Allocates a Universal Procedure Pointer for the MCActionFilterProc callback.

```
MCActionFilterUPP NewMCActionFilterUPP (
   MCActionFilterProcPtr userRoutine
);
```

Parameters

userRoutine

A pointer to your application-defined function.

Return Value

A new UPP; see `Universal Procedure Pointers`.

Discussion

This function is used with Macintosh PowerPC systems. See *Inside Macintosh: PowerPC System Software*.

Version Notes

Introduced in QuickTime 4.1. Replaces `NewMCActionFilterProc`.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`Movies.h`

NewMCActionFilterWithRefConUPP

Allocates a Universal Procedure Pointer for the MCActionFilterWithRefConProc callback.

```
MCActionFilterWithRefConUPP NewMCActionFilterWithRefConUPP (
   MCActionFilterWithRefConProcPtr userRoutine
);
```

Parameters

userRoutine

A pointer to your application-defined function.

Return Value

A new UPP; see `Universal Procedure Pointers`.

Discussion

This function is used with Macintosh PowerPC systems. See *Inside Macintosh: PowerPC System Software*.

Version Notes

Introduced in QuickTime 4.1. Replaces `NewMCActionFilterWithRefConProc`.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

Movie From DataRef

qteffects.win

QTKitMovieShuffler

vrbackbuffer

vrmovies.win

Declared In
`Movies.h`

Callbacks

MCActionFilterProc

Responds to movie controller actions.

```
typedef Boolean (*MCActionFilterProcPtr) (MovieController mc, short *action, void
 *params);
```

If you name your function `MyMCActionFilterProc`, you would declare it this way:

```
Boolean MyMCActionFilterProc (
    MovieController    mc,
    short              *action,
    void               *params );
```

Parameters

mc

Specifies the movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

action

A movie controller action. For a list of actions, see Chapter 2 of *Inside Macintosh: QuickTime* Components.

params

A pointer to a structure, such as `QTStatusStringRecord` or `ResolvedQTEventSpec`, that passes information to the callback. See `Movies.h`.

Return Value
Undocumented

Discussion
Movie controller components allow your application to field movie controller actions. You define an `MCActionFilterProc` in your application and assign it to a controller by calling the `MCSetActionFilterWithRefCon` function.

Declared In
`Movies.h, HIMovieView.h`

MCActionFilterWithRefConProc

Responds to movie controller actions with a reference constant.

```
typedef Boolean (*MCActionFilterWithRefConProcPtr) (MovieController mc, short
action, void *params, long refCon);
```

If you name your function `MyMCActionFilterWithRefConProc`, you would declare it this way:

```
Boolean MyMCActionFilterWithRefConProc (
    MovieController    mc,
    short              action,
    void               *params,
    long               refCon );
```

Parameters

mc

Specifies the movie controller for the operation. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

action

A movie controller action. For a list of actions, see Chapter 2 of *Inside Macintosh: QuickTime* Components.

params

A pointer to a structure, such as `QTStatusStringRecord`, that passes information to the callback. See `Movies.h`.

refCon

A reference constant that the client code supplies to your callback. You can use this reference to point to a data structure containing any information your callback needs.

Return Value

Undocumented

Declared In

`Movies.h, HIMovieView.h`

Data Types

MCActionFilterUPP

Represents a type used by the Movie Controller API.

```
typedef STACK_UPP_TYPE(MCActionFilterProcPtr) MCActionFilterUPP;
```

Availability

Available in Mac OS X v10.0 and later.

Declared In

`Movies.h`

MCActionFilterWithRefConUPP

Represents a type used by the Movie Controller API.

```
typedef STACK_UPP_TYPE(MCActionFilterWithRefConProcPtr) MCActionFilterWithRefConUPP;
```

Availability

Available in Mac OS X v10.0 and later.

Declared In
`Movies.h`

MCInterfaceElement

Represents a type used by the Movie Controller API.

```
typedef unsigned long MCInterfaceElement;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`Movies.h`

OptionBits

Represents a type used by the Movie Controller API.

```
typedef UInt32 OptionBits;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`OSTypes.h`

Constants

Movie Controller Options

Constants that represent options for movie controllers.

```
enum {
  kMCIEEnabledButtonPicture     = 1,
  kMCIEDisabledButtonPicture    = 2,
  kMCIEDepressedButtonPicture   = 3,
  kMCIEEnabledSizeBoxPicture    = 4,
  kMCIEDisabledSizeBoxPicture   = 5,
  kMCIEEnabledUnavailableButtonPicture = 6,
  kMCIEDisabledUnavailableButtonPicture = 7,
  kMCIESoundSlider              = 128,
  kMCIESoundThumb               = 129,
  kMCIEColorTable               = 256,
  kMCIEIsFlatAppearance         = 257,
  kMCIEDoButtonIconsDropOnDepress = 258
};
enum {
  mcFlagSuppressMovieFrame      = 1 << 0,
  mcFlagSuppressStepButtons     = 1 << 1,
  mcFlagSuppressSpeakerButton   = 1 << 2,
  mcFlagsUseWindowPalette       = 1 << 3,
  mcFlagsDontInvalidate         = 1 << 4,
  mcFlagsUseCustomButton        = 1 << 5
};
enum {
  mcInfoUndoAvailable           = 1 << 0,
  mcInfoCutAvailable            = 1 << 1,
  mcInfoCopyAvailable           = 1 << 2,
  mcInfoPasteAvailable          = 1 << 3,
  mcInfoClearAvailable          = 1 << 4,
  mcInfoHasSound                = 1 << 5,
  mcInfoIsPlaying               = 1 << 6,
  mcInfoIsLooping               = 1 << 7,
  mcInfoIsInPalindrome          = 1 << 8,
  mcInfoEditingEnabled          = 1 << 9,
  mcInfoMovieIsInteractive      = 1 << 10
};
```

Constants

`kMCIESoundThumb`

The indicator on the sound slider.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`mcFlagSuppressMovieFrame`

If this flag is set to 1, the controller does not display a frame around the movie. By default, this flag is set to 0.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`mcFlagSuppressStepButtons`

If this flag is set to 1, the controller does not display the step buttons. By default, this flag is set to 0.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`mcFlagSuppressSpeakerButton`

If this flag is set to 1, the controller does not display the speaker button. By default, this flag is set to 0.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`mcFlagsUseWindowPalette`

If this flag is set to 1, the movie controller does not manage the window palette. This ensures that a movie's colors are reproduced as accurately as possible. This flag is particularly useful for movies with custom color tables. By default, this flag is set to 0.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

Declared In

`Movies.h, HIMovieView.h`

MCAdjustCursor Values

Constants passed to MCAdjustCursor.

```
enum {
  kQTCursorOpenHand             = -19183,
  kQTCursorClosedHand           = -19182,
  kQTCursorPointingHand         = -19181,
  kQTCursorRightArrow           = -19180,
  kQTCursorLeftArrow            = -19179,
  kQTCursorDownArrow            = -19178,
  kQTCursorUpArrow              = -19177,
  kQTCursorIBeam                = -19176
};
```

Declared In

`Movies.h, HIMovieView.h`

MCGetMenuString Values

Constants passed to MCGetMenuString.

```
enum {
  mcMenuUndo                    = 1,
  mcMenuCut                     = 3,
  mcMenuCopy                    = 4,
  mcMenuPaste                   = 5,
  mcMenuClear                   = 6
};
```

Declared In

`Movies.h, HIMovieView.h`

MCPositionController Values

Constants passed to MCPositionController.

```
enum {
  mcPositionDontInvalidate      = 1 << 5
};
```

Declared In

Movies.h, HIMovieView.h

Document Revision History

This table describes the changes to *Movie Controller Reference*.

Date	Notes
2006-05-23	New document, based on previously published material, that describes the API for QuickTime movie controllers.

Index