

---

# QTKit Reference Update

[QuickTime](#) > [Cocoa](#)



2007-07-18



Apple Inc.  
© 2007 Apple Inc.  
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

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

Apple, the Apple logo, Cocoa, eMac, Mac, Mac OS, Objective-C, Pixlet, Quartz, QuickDraw, and QuickTime are trademarks of Apple Inc., registered in the United States and other countries.

Aperture and Numbers are trademarks of Apple Inc.

OpenGL is a registered trademark of Silicon Graphics, Inc.

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

---

## Introduction to QTKit Reference Update 5

---

Organization of This Document 5

See Also 5

---

## 10.5 Symbol Changes 7

---

### Classes 7

NSCoder 7

NSObject 7

NSNumber 7

QTCaptureAudioPreviewOutput (New) 8

QTCaptureConnection (New) 8

QTCaptureDecompressedVideoOutput (New) 9

QTCaptureDevice (New) 9

QTCaptureDeviceInput (New) 10

QTCaptureFileOutput (New) 11

QTCaptureInput (New) 12

QTCaptureLayer (New) 12

QTCaptureMovieFileOutput (New) 12

QTCaptureOutput (New) 12

QTCaptureSession (New) 13

QTCaptureVideoPreviewOutput (New) 13

QTCaptureView (New) 14

QTCaptureOptions (New) 15

QTFormatDescription (New) 15

QTMovie 16

QTMovieLayer (New) 18

QTMovieView 18

QTSampleBuffer (New) 19

### C Symbols 20

QTCaptureConnection.h 20

QTCaptureDevice.h 21

QTCaptureFileOutput.h 24

QTCaptureSession.h 24

QTErrors.h 25

QTFormatDescription.h 26

QTKitDefines.h 27

QTMedia.h 33

QTMovie.h 33

QTSampleBuffer.h 35

QTTime.h 35

QTTrack.h 36

## 10.4 Symbol Changes 37

---

C Symbols 37  
QTDataReference.h 37  
QTMedia.h 37  
QTMovie.h 39  
QTMovieView.h 42  
QTTime.h 43  
QTTimeRange.h 43  
QTTrack.h 44  
QTUtilities.h 45

## 10.3 Symbol Changes 47

---

Classes 47  
NSCoder 47  
NSObject 47  
NSValue 48  
QTDataReference (New) 48  
QTMedia (New) 50  
QTMovie (New) 50  
QTMovieView (New) 55  
QTTrack (New) 56  
C Symbols 58  
QTKit.h 58  
QTKitDefines.h 58  
QTMovie.h 59  
QTTime.h 59  
QTTimeRange.h 59

## Document Revision History 61

---

# Introduction to QTKit Reference Update

---

This document summarizes the symbols that have been added to the QTKit framework. The full reference documentation notes in what version a symbol was introduced, but sometimes it's useful to see only the new symbols for a given release.

If you are not familiar with this framework you should refer to the complete framework reference documentation.

## Organization of This Document

Symbols are grouped by class or protocol for Objective-C and by header file for C. For each symbol there is a link to complete documentation, if available, and a brief description, if available.

## See Also

For reference documentation on this framework, see *QTKit Framework Reference*.



# 10.5 Symbol Changes

---

This article lists the symbols added to `QTKit.framework` in Mac OS X v10.5.

## Classes

All of the classes with new symbols are listed alphabetically, with their new class, instance, and delegate methods described.

### NSCoder

---

Complete reference information is available in the `NSCoder` reference.

#### Instance Methods

---

<code>decodeSMPTETimeForKey:</code>	Decodes an <code>SMPTETime</code> structure encoded by the receiver for the given key.
<code>encodeSMPTETime:forKey:</code>	Encodes an <code>SMPTETime</code> for the given key.

### NSObject

---

Complete reference information is available in the `NSObject` reference.

#### Instance Methods

---

<code>captureOutput:didOutputVideoFrame:withSampleBuffer:fromConnection:</code>	Called whenever the video preview output outputs a new video frame.
---	---

### NSValue

---

Complete reference information is available in the `NSValue` reference.

#### Class Methods

---

<code>valueWithSMPTETime:</code>	Returns a new <code>NSValue</code> object containing an <code>SMPTETime</code> .
----------------------------------	--

## Instance Methods

---

<code>SMPTETimeValue</code>	Returns a <code>SMPTETime</code> structure contained in an <code>NSValue</code> .
-----------------------------	---

## QTCaptureAudioPreviewOutput (New)

---

Complete reference information is available in the `QTCaptureAudioPreviewOutput` reference.

### Instance Methods

---

<code>outputDeviceUniqueID</code>	Returns the unique ID of the Core Audio output device being used to play preview audio.
<code>setOutputDeviceUniqueID:</code>	Sets the unique ID of the Core Audio output device being used to play preview audio.
<code>setVolume:</code>	Sets the preview volume of the output.
<code>volume</code>	Returns the preview volume of the output.

## QTCaptureConnection (New)

---

Complete reference information is available in the `QTCaptureConnection` reference.

### Instance Methods

---

<code>attributeForKey:</code>	Returns the current value of the connection attribute for key.
<code>attributeIsReadOnly:</code>	Returns a Boolean value indicating whether the given attribute for the connection cannot be modified.
<code>connectionAttributes</code>	Returns a dictionary of all attributes set for the receiver.
<code>formatDescription</code>	Returns the format description of the receiver.
<code>isEnabled</code>	Returns a Boolean value indicating whether the receiver is enabled.
<code>mediaType</code>	Returns the QuickTime media type of the receiver.
<code>owner</code>	Returns the <code>QTCaptureInput</code> or <code>QTCaptureOutput</code> object that owns the receiver.
<code>setAttribute:forKey:</code>	Sets a connection attribute for the given key.
<code>setConnectionAttributes:</code>	Sets the connection's attributes from the key-value pairs specified in the given dictionary.
<code>setEnabled:</code>	Sets whether the receiver is enabled.



## QTCaptureDecompressedVideoOutput (New)

---

Complete reference information is available in the [QTCaptureDecompressedVideoOutput](#) reference.

### Instance Methods

---

<code>delegate</code>	Returns the receiver's delegate.
<code>outputVideoFrame:withSampleBuffer:fromConnection:</code>	Called whenever the receiver outputs a new video frame.
<code>pixelBufferAttributes</code>	Returns the Core Video pixel buffer attributes previously set by <code>setPixelBufferAttributes:</code> that determine what kind of pixel buffers are output by the receiver.
<code>setDelegate:</code>	Sets the receiver's delegate.
<code>setPixelBufferAttributes:</code>	Sets the Core Video pixel buffer attributes that determine what kind of pixel buffers are output by the receiver.

## QTCaptureDevice (New)

---

Complete reference information is available in the [QTCaptureDevice](#) reference.

### Class Methods

---

<code>defaultInputDeviceWithMediaType:</code>	Returns a <code>QTCaptureDevice</code> instance for the default device connected to the user's system of the given media type.
<code>deviceWithUniqueID:</code>	Returns a <code>QTCaptureDevice</code> instance with the identifier device UID.
<code>inputDevices</code>	Returns an array of devices currently connected to the computer that can be used as input sources.
<code>inputDevicesWithMediaType:</code>	Returns an array of input devices currently connected to the computer that send a stream with the given media type.

### Instance Methods

---

<code>attributeForKey:</code>	Returns a device attribute for the given key.
<code>attributeIsReadOnly:</code>	Returns whether the given attribute for the device cannot be modified.

<code>close</code>	Releases application control over the device acquired in the <code>open:</code> method.
<code>deviceAttributes</code>	Returns a dictionary of the device's current attributes.
<code>formatDescriptions</code>	Returns an array of stream formats currently in use by the device.
<code>hasMediaType:</code>	Returns whether the receiver sends a stream with the given media type.
<code>isConnected</code>	Returns YES if the device is connected to the computer.
<code>isInUseByAnotherApplication</code>	Returns YES if the device is connected, but being exclusively used by another application.
<code>isOpen</code>	Returns YES if the device is open in the current application.
<code>localizedDisplayName</code>	Returns a localized human-readable name for the receiver's device.
<code>modelUniqueID</code>	Returns the unique ID of the model of the receiver's device.
<code>open:</code>	Attempts to give the application control over the device so that it can be used for capture.
<code>setAttribute:forKey:</code>	Sets a device attribute for the given key.
<code>setDeviceAttributes:</code>	Sets attributes on the device from the key-value pairs in the given dictionary.
<code>uniqueID</code>	Returns the unique ID of the receiver's device.

## QTCaptureDeviceInput (New)

---

Complete reference information is available in the `QTCaptureDeviceInput` reference.

### Class Methods

---

<code>deviceInputWithDevice:</code>	Returns an autoreleased instance of <code>QTCaptureDeviceInput</code> associated with the given device.
-------------------------------------	---

### Instance Methods

---

<code>device</code>	Returns the device associated with the receiver.
<code>initWithDevice:</code>	Returns an instance of <code>QTCaptureDeviceInput</code> associated with the given device.

## QTCaptureFileOutput (New)

Complete reference information is available in the [QTCaptureFileOutput reference](#).

### Instance Methods

<code>compressionOptionsForConnection:</code>	Returns the options the receiver uses to compress media on the given connection as it is being captured.
<code>delegate</code>	Returns the receiver's delegate.
<code>maximumRecordedDuration</code>	Returns the maximum duration of the media that should be recorded by the receiver.
<code>maximumRecordedFileSize</code>	Returns the maximum file size, in bytes, of the file that should be recorded by the receiver.
<code>outputFileURL</code>	Returns the file written to by the receiver.
<code>recordedDuration</code>	Returns the duration of the media recorded by the receiver.
<code>recordedFileSize</code>	Returns the size, in bytes, of the data recorded by the receiver to output files.
<code>recordToOutputFileURL:</code>	Sets the file written to by the receiver.
<code>recordToOutputFileURL:bufferDestination:</code>	Sets the file written to by the receiver, specifying where the sample buffer currently in flight should be recorded.
<code>setCompressionOptions:forConnection:</code>	Sets the options the receiver uses to compress media on the given connection as it is being captured.
<code>setDelegate:</code>	Sets the receiver's delegate.
<code>setMaximumRecordedDuration:</code>	Sets the maximum duration of the media that should be recorded by the receiver.
<code>setMaximumRecordedFileSize:</code>	Sets the maximum file size, in bytes, of the file that should be recorded by the receiver.

### Delegate Methods

<code>captureOutput:didFinishRecordingToOutputFileAtURL: forConnections:dueToError:</code>	
<code>captureOutput:didOutputSampleBuffer: fromConnection:</code>	
<code>captureOutput:didStartRecordingToOutputFileAtURL: forConnections:</code>	

<code>captureOutput:mustChangeOutputFileAtURL: forConnections:dueToError:</code>	
<code>captureOutput:shouldChangeOutputFileAtURL: forConnections:dueToError:</code>	
<code>captureOutput: willFinishRecordingToOutputFileAtURL: forConnections:dueToError:</code>	
<code>captureOutput:willStartRecordingToOutputFileAtURL: forConnections:</code>	

## QTCaptureInput (New)

---

Complete reference information is available in the [QTCaptureInput reference](#).

### Instance Methods

---

<code>connections</code>	Returns an array of connections owned by the receiver.
--------------------------	--

## QTCaptureLayer (New)

---

Complete reference information is available in the [QTCaptureLayer reference](#).

### Class Methods

---

<code>layerWithSession:</code>	Creates an autoreleased <code>QTCaptureLayer</code> associated with the specified <code>QTCaptureSession</code> object.
--------------------------------	---

### Instance Methods

---

<code>initWithSession:</code>	Creates a <code>QTCaptureLayer</code> associated with the specified <code>QTCaptureSession</code> object.
<code>session</code>	Returns the capture session associated with a <code>QTCaptureLayer</code> object.
<code>setSession:</code>	Sets or resets the capture session associated with a <code>QTCaptureLayer</code> object.

## QTCaptureMovieFileOutput (New)

---

Complete reference information is available in the [QTCaptureMovieFileOutput reference](#).

## QTCaptureOutput (New)

---

Complete reference information is available in the [QTCaptureOutput reference](#).

## Instance Methods

---

<code>connections</code>	Returns an array of connections owned by the receiver that are currently connected to a capture session.
--------------------------	--

## QTCaptureSession (New)

---

Complete reference information is available in the [QTCaptureSession](#) reference.

## Instance Methods

---

<code>addInput:error:</code>	Adds an input to the receiver.
<code>addOutput:error:</code>	Adds an output to the receiver.
<code>inputs</code>	Returns an array of inputs connected to the receiver.
<code>isRunning</code>	Returns whether the receiver is running.
<code>outputs</code>	Returns an array of outputs connected to the receiver.
<code>removeInput:</code>	Removes an input from the receiver.
<code>removeOutput:</code>	Removes an output from the receiver.
<code>startRunning</code>	Tells the receiver to start capturing data from its inputs and sending data to its outputs.
<code>stopRunning</code>	Tells the receiver to stop capturing data from its inputs and sending data to its outputs.

## QTCaptureVideoPreviewOutput (New)

---

Complete reference information is available in the [QTCaptureVideoPreviewOutput](#) reference.

## Instance Methods

---

<code>delegate</code>	Returns the receiver's delegate.
<code>outputVideoFrame:withSampleBuffer:fromConnection:</code>	Called whenever the receiver outputs a new video frame.
<code>pixelBufferAttributes</code>	Returns the Core Video pixel buffer attributes previously set by <code>setPixelBufferAttributes:</code> that determine what kind of pixel buffers are output by the receiver.

<code>setDelegate:</code>	Sets the receiver's delegate.
<code>setPixelBufferAttributes:</code>	Sets the CoreVideo pixel buffer attributes that determine what kind of pixel buffers are output by the receiver.
<code>setVisualContext:forConnection:</code>	Sets the QuickTime visual context used to preview the video for the described connection.
<code>visualContextForConnection:</code>	Returns the QuickTime visual context used to preview the video for the given connection.

## QTCaptureView (New)

---

Complete reference information is available in the [QTCaptureView](#) reference.

### Instance Methods

---

<code>availableVideoPreviewConnections</code>	Returns an array of output video connections that can be previewed.
<code>captureSession</code>	Returns the capture session being previewed by the receiver.
<code>delegate</code>	Returns the receiver's delegate.
<code>fillColor</code>	Returns the fill color drawn in the area of the view not covered by the video preview.
<code>preservesAspectRatio</code>	Returns whether the receiver preserves the aspect ratio of the video preview when drawing it.
<code>previewBounds</code>	Returns the rectangle occupied by the video preview in the view.
<code>setCaptureSession:</code>	Sets the capture session to be previewed by the receiver.
<code>setDelegate:</code>	Sets the receiver's delegate.
<code>setFillColor:</code>	Sets the fill color drawn in the area of the view not covered by the video preview.
<code>setPreservesAspectRatio:</code>	Sets whether the receiver preserves the aspect ratio of the video preview when drawing it.
<code>setVideoPreviewConnection:</code>	Sets the output connection to be previewed by the receiver.
<code>videoPreviewConnection</code>	Returns the output connection being previewed by the receiver.

## Delegate Methods

---

<code>view:willDisplayImage:</code>	Delegates of <code>QTCaptureView</code> can implement this method to modify the image that is to be drawn into a <code>QTCaptureView</code> .
-------------------------------------	---

## QTCompressionOptions (New)

---

Complete reference information is available in the `QTCompressionOptions` reference.

### Class Methods

---

<code>compressionOptionsIdentifiersForMediaType:</code>	Returns all of the possible identifiers for the given media type that can be used with <code>compressionOptionsWithIdentifier:</code> on the user's system.
<code>compressionOptionsWithIdentifier:</code>	Returns a compression options object configured for the given identifier.

### Instance Methods

---

<code>isEqualToCompressionOptions:</code>	Returns whether the receiver contains options identical to those in the given compression options object.
<code>localizedCompressionOptionsSummary</code>	A localized summary of the receiver's compression options.
<code>localizedDisplayName</code>	A short localized name describing the receiver's compression options.
<code>mediaType</code>	The media type on which the receiver's compression options should be used.

## QTFormatDescription (New)

---

Complete reference information is available in the `QTFormatDescription` reference.

### Instance Methods

---

<code>attributeForKey:</code>	Returns the current value of the format description attribute for the given key.
<code>formatDescriptionAttributes</code>	Returns a dictionary of all attributes set for the receiver.
<code>formatType</code>	Returns the format type of the described media, a four character code representing the format or codec type.

<code>isEqualToFormatDescription:</code>	Returns whether the receiver describes the same format as the given format description.
<code>localizedFormatSummary</code>	Returns a localized summary of the media format.
<code>mediaType</code>	Returns the media type of the described media.
<code>quickTimeSampleDescription</code>	Returns the media's QuickTime SampleDescription.

## QTMovie

---

Complete reference information is available in the `QTMovie` reference.

### Class Methods

---

<code>enterQTKitOnThread</code>	Performs any QuickTime-specific initialization for the current (non-main) thread; must be paired with a call to <code>exitQTKitOnThread</code> .
<code>enterQTKitOnThreadDisablingThreadSafetyProtection</code>	Performs any QuickTime-specific initialization for the current (non-main) thread, allowing non-threadsafe components; must be paired with a call to <code>exitQTKitOnThread</code> .
<code>exitQTKitOnThread</code>	Performs any QuickTime-specific shut-down for the current (non-main) thread; must be paired with a call to <code>enterQTKitOnThread</code> or <code>enterQTKitOnThreadDisablingThreadSafetyProtection</code> .
<code>movieTypesWithOptions:</code>	Returns an array of UTIs that QuickTime can open.

### Instance Methods

---

<code>addChapters:withAttributes:error:</code>	
<code>attachToCurrentThread</code>	Attaches the receiver to the current thread; returns YES if successful, NO otherwise.
<code>autoplay</code>	Sets a movie to start playing when a sufficient amount of media data is available.
<code>chapterCount</code>	Returns the number of chapters in the receiver, or 0 if there are no chapters.



<code>chapterIndexForTime:</code>	Returns the 0-based index of the chapter that contains the specified movie time.
<code>chapters</code>	Returns an NSArray containing information about the chapters in the receiver.
<code>detachFromCurrentThread</code>	Detaches the receiver from the current thread; returns YES if successful, NO otherwise.
<code>frameImageAtTime:withAttributes:error:</code>	Returns an NSImage*, CIImage*, CGImageRef, CVPixelBufferRef, or CVOpenGLTextureRef for the movie image at the specified time
<code>hasChapters</code>	Returns YES if the receiver has chapters, NO otherwise.
<code>initWithWritableData:error:</code>	Useful for directly passing filenames and data objects. The QTMovie returned by this method is editable.
<code>initWithWritableDataReference:error:</code>	Creates a new storage container at the location specified by dataReference and returns a QTMovie object that has that container as its default data reference.
<code>initWithWritableFile:error:</code>	Useful for directly passing filenames and data objects. The QTMovie returned by this method is editable.
<code>insertSegmentOfTrack:fromRange:scaledToRange:</code>	
<code>insertSegmentOfTrack:timeRange:atTime:</code>	
<code>invalidate</code>	
<code>isIdling</code>	Returns the current idling state of a QTMovie object.
<code>removeChapters</code>	Removes any existing chapters from the receiver.
<code>removeTrack:</code>	
<code>setIdling:</code>	Sets the movie to idle YES or not to idle NO.
<code>setVisualContext:</code>	
<code>startTimeOfChapter:</code>	Returns a QTTime structure that is the start time of the chapter having the specified 0-based index in the list of chapters.
<code>visualContext</code>	

<code>writeToFile:withAttributes:error:</code>	Returns an <code>NSError</code> object if an error occurs and if <code>errorPtr</code> is non-NULL.
--	---

## QTMovieLayer (New)

---

Complete reference information is available in the `QTMovieLayer` reference.

### Class Methods

---

<code>layerWithMovie:</code>	Creates an autoreleased <code>QTMovieLayer</code> associated with the specified <code>QTMovie</code> object.
------------------------------	--

### Instance Methods

---

<code>initWithMovie:</code>	Creates a <code>QTMovieLayer</code> associated with the specified <code>QTMovie</code> object.
<code>movie</code>	Returns the movie associated with a <code>QTMovieLayer</code> object.
<code>setMovie:</code>	Sets the <code>QTMovie</code> object in a <code>QTMovieLayer</code> to <code>movie</code> .

## QTMovieView

---

Complete reference information is available in the `QTMovieView` reference.

### Instance Methods

---

<code>areStepButtonsVisible</code>	Returns the current visibility state of the specified controller bar button.
<code>areZoomButtonsVisible</code>	Returns the current visibility state of the specified controller bar button.
<code>delegate</code>	
<code>isBackButtonVisible</code>	Returns the current visibility state of the specified controller bar button.
<code>isCustomButtonVisible</code>	Returns the current visibility state of the specified controller bar button.
<code>isHotSpotButtonVisible</code>	Returns the current visibility state of the specified controller bar button.
<code>isTranslateButtonVisible</code>	Returns the current visibility state of the specified controller bar button.

<code>isVolumeButtonVisible</code>	Returns the current visibility state of the specified controller bar button.
<code>setBackButtonVisible:</code>	Sets the specified controller bar button to be visible or invisible, according to the state parameter.
<code>setCustomButtonVisible:</code>	Sets the specified controller bar button to be visible or invisible, according to the state parameter.
<code>setDelegate:</code>	
<code>setHotSpotButtonVisible:</code>	Sets the specified controller bar button to be visible or invisible, according to the state parameter.
<code>setStepButtonsVisible:</code>	Sets the specified controller bar button to be visible or invisible, according to the state parameter.
<code>setTranslateButtonVisible:</code>	Sets the specified controller bar button to be visible or invisible, according to the state parameter.
<code>setVolumeButtonVisible:</code>	Sets the specified controller bar button to be visible or invisible, according to the state parameter.
<code>setZoomButtonsVisible:</code>	Sets the specified controller bar button to be visible or invisible, according to the state parameter.

## QTSampleBuffer (New)

---

Complete reference information is available in the `QTSampleBuffer` reference.

### Instance Methods

---

<code>attributeForKey:</code>	Returns a sample buffer attribute for the given key.
<code>audioBufferListWithOptions:</code>	Returns a pointer to a Core Audio <code>AudioBufferList</code> containing audio data owned by the receiver.
<code>bytesForAllSamples</code>	Returns a pointer to the bytes of media data contained in the sample buffer.
<code>decodeTime</code>	Returns the decode time of the buffer.
<code>decrementSampleUseCount</code>	Decrements the use count of the sample data owned by the receiver, allowing the sample data to be invalidated after a matching call to <code>incrementSampleUseCount</code> .
<code>duration</code>	Returns the duration of the buffer.
<code>formatDescription</code>	Returns the format description of the buffer.

<code>getAudioStreamPacketDescriptions:inRange:</code>	Gets an array of Core Audio <code>AudioStreamPacketDescriptions</code> describing the lengths of samples in variable bit- rate audio buffers.
<code>incrementSampleUseCount</code>	Increments the use count of the sample data owned by the receiver, preventing the sample data from being invalidated until a matching call to <code>decrementSampleUseCount</code> .
<code>lengthForAllSamples</code>	Returns the length of the buffer returned by <code>bytesForAllSamples</code> .
<code>numberOfSamples</code>	Returns the number of media samples contained in the buffer.
<code>presentationTime</code>	Returns the presentation time of the buffer.
<code>sampleBufferAttributes</code>	Returns a dictionary of the sample buffer's current attributes.
<code>sampleUseCount</code>	Returns the use count of the sample data owned by the receiver.

## C Symbols

All of the header files with new symbols are listed alphabetically, with their new symbols described.

### QTCaptureConnection.h

---

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>QTCaptureConnectionAttributeDidChangeNotification</code>	Posted when one of the connection's attributes has changed.
<code>QTCaptureConnectionAttributeWillChangeNotification</code>	Posted when one of the connection's attributes is about to change.
<code>QTCaptureConnectionAudioAveragePowerLevelsAttribute</code>	An <code>NSArray</code> of <code>NSNumber</code> s that correspond to the average power, in decibels, of each audio stream sent through the connection.

<code>QTCaptureConnectionAudioMasterVolumeAttribute</code>	An <code>NSNumber</code> that specifies the master volume of all audio channels sent through the connection.
<code>QTCaptureConnectionAudioPeakHoldLevelsAttribute</code>	An <code>NSArray</code> of <code>NSNumber</code> s that correspond to the peak hold level, in decibels, of each audio channel sent through the connection.
<code>QTCaptureConnectionAudioVolumesAttribute</code>	An <code>NSArray</code> of <code>NSNumber</code> s that specify the volumes of audio channels sent through the connection.
<code>QTCaptureConnectionChangedAttributeKey</code>	Used as a key in the user info dictionary passed to <code>QTCaptureConnectionAttributeWillChangeNotification</code> , and <code>QTCaptureConnectionAttributeDidChangeNotification</code> to indicate the key of that attribute that changed.
<code>QTCaptureConnectionEnabledAudioChannelsAttribute</code>	An <code>NSIndexSet</code> that specifies which audio channels should be sent through the connection. The indices in the set should be between 0 and the number of volumes in <code>QTCaptureConnectionAudioVolumesAttribute</code> . This attribute allows applications to selectively disable certain audio channels from being sent through the connection. The value of this attribute should be an <code>NSIndexSet</code> that contains only the channels that should be used. By default, all audio channels are sent though a connection. This string value can be used in key paths for key-value coding, key-value observing, and bindings.
<code>QTCaptureConnectionFormatDescriptionDidChangeNotification</code>	Posted when the format description of a connection has changed.
<code>QTCaptureConnectionFormatDescriptionWillChangeNotification</code>	Posted when the format description of a connection is about to change.

## QTCaptureDevice.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>QTCaptureDeviceAttributeDidChangeNotification</code>	Posted when the one of device's attributes has changed.
<code>QTCaptureDeviceAttributeWillChangeNotification</code>	Posted when one of the device's attributes is about to change.
<code>QTCaptureDeviceAvailableInputSourcesAttribute</code>	For devices with multiple possible input sources, returns an array of dictionaries describing each available input source.
<code>QTCaptureDeviceAVCTransportControlsAttribute</code>	For AVC devices that read data from linear media, such as tapes, specifies the mode and speed at which that media is playing.
<code>QTCaptureDeviceAVCTransportControlsFastestForwardSpeed</code>	Media runs forward at greater than normal speed.
<code>QTCaptureDeviceAVCTransportControlsFastestReverseSpeed</code>	Media runs in reverse at greater than normal speed.
<code>QTCaptureDeviceAVCTransportControlsFastForwardSpeed</code>	Media runs forward at greater than normal speed.
<code>QTCaptureDeviceAVCTransportControlsFastReverseSpeed</code>	Media runs in reverse at greater than normal speed.
<code>QTCaptureDeviceAVCTransportControlsNormalForwardSpeed</code>	Media runs forward at normal speed.
<code>QTCaptureDeviceAVCTransportControlsNormalReverseSpeed</code>	Media runs in reverse at normal speed.
<code>QTCaptureDeviceAVCTransportControlsNotPlayingMode</code>	
<code>QTCaptureDeviceAVCTransportControlsPlaybackMode</code>	
<code>QTCaptureDeviceAVCTransportControlsPlaybackModeKey</code>	A value provided with the <code>QTCaptureDeviceAVCTransportControlsPlaybackModeKey</code> key that specifies whether the device previews audio and displays video while it is running through linear media.
<code>QTCaptureDeviceAVCTransportControlsPlayingMode</code>	
<code>QTCaptureDeviceAVCTransportControlsSlowestForwardSpeed</code>	Media runs forward at less than normal speed.
<code>QTCaptureDeviceAVCTransportControlsSlowestReverseSpeed</code>	Media runs in reverse at less than normal speed.
<code>QTCaptureDeviceAVCTransportControlsSlowForwardSpeed</code>	Media runs forward at less than normal speed.

<code>QTCaptureDeviceAVCTransportControlsSlowReverseSpeed</code>	Media runs in reverse at less than normal speed.
<code>QTCaptureDeviceAVCTransportControlsSpeed</code>	
<code>QTCaptureDeviceAVCTransportControlsSpeedKey</code>	Specifies the approximate rate at which the device runs through linear media.
<code>QTCaptureDeviceAVCTransportControlsStoppedSpeed</code>	Media is paused.
<code>QTCaptureDeviceAVCTransportControls-VeryFastForwardSpeed</code>	Media runs forward at greater than normal speed.
<code>QTCaptureDeviceAVCTransportControls-VeryFastReverseSpeed</code>	Media runs in reverse at greater than normal speed.
<code>QTCaptureDeviceAVCTransportControls-VerySlowForwardSpeed</code>	Media runs forward at less than normal speed.
<code>QTCaptureDeviceAVCTransportControls-VerySlowReverseSpeed</code>	Media runs in reverse at less than normal speed.
<code>QTCaptureDeviceChangedAttributeKey</code>	Indicates the key of the attribute that changed.
<code>QTCaptureDeviceFormatDescriptions-DidChangeNotification</code>	Posted when the device's formats that are returned by the <code>formatDescriptions</code> method have just changed.
<code>QTCaptureDeviceFormatDescriptions-WillChangeNotification</code>	Posted when the device's formats that are returned by the <code>formatDescriptions</code> method are about to change.
<code>QTCaptureDeviceInputSourceIdentifierAttribute</code>	Used to get and set the currently used input source for the device.
<code>QTCaptureDeviceInputSourceIdentifierKey</code>	An object representing a unique ID for the input source.
<code>QTCaptureDeviceInputSourceLocalizedDisplayNameKey</code>	The localized display name of an input source, suitable for display in a user interface.
<code>QTCaptureDeviceLegacySequenceGrabberAttribute</code>	An <code>NSValue</code> interpreted as a <code>ComponentInstance</code> for the legacy sequence grabber component used by the device.
<code>QTCaptureDeviceLinkedDevicesAttribute</code>	Returns an array of <code>QTCaptureDevice</code> objects that, although they are separate devices on the system, are a part of the same physical device as the receiver.

<code>QTCaptureDeviceSuspendedAttribute</code>	Returns whether or not data capture on the device is suspended due to a feature on the device.
<code>QTCaptureDeviceWasConnectedNotification</code>	Posted when a device is connected or turned on.
<code>QTCaptureDeviceWasDisconnectedNotification</code>	Posted when a device is disconnected or turned off.

## QTCaptureFileOutput.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>QTCaptureFileOutputBufferDestination</code>	
<code>QTCaptureFileOutputBufferDestinationNewFile</code>	
<code>QTCaptureFileOutputBufferDestinationOldFile</code>	

## QTCaptureSession.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>QTCaptureSessionErrorKey</code>	Used as a notification key in the user info dictionary passed to <code>QTCaptureSessionRuntimeErrorNotification</code> to indicate the error responsible for the notification. The value is an <code>NSError</code> .
<code>QTCaptureSessionRuntimeErrorNotification</code>	Posted when an error occurs that while a capture session is running prevents input media from being previewed or captured. The notification user info dictionary <code>QTCaptureSessionErrorKey</code> entry contains an <code>NSError</code> object that describes the error that prevented the session from running properly. Normally, such errors are caused by an invalid configuration of inputs and outputs.



## QLError.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

QLErrorCaptureInputKey	Use this key to retrieve the QTCaptureInput object for which the error occurred.
QLErrorCaptureOutputKey	Use this key to retrieve the QTCaptureOutput object for which the error occurred.
QLErrorDeviceAlreadyUsedbyAnotherSession	The device could not be added to the session because it experiences a runtime error trying to use a device already being used by another session.
QLErrorDeviceExcludedByAnotherDevice	The device is excluded by another device.
QLErrorDeviceInUseByAnotherApplication	The device is in use by another application.
QLErrorDeviceKey	Use this key to retrieve the QTCaptureDevice object for which the error occurred.
QLErrorDeviceNotConnected	The device is not connected to the computer.
QLErrorDeviceWasDisconnected	The recording has been automatically stopped because an input device was disconnected.
QLErrorDiskFull	The recording has been automatically stopped because the disk being used for recorded products is full.
QLErrorExcludingDeviceKey	Use this key to retrieve the QTCaptureDevice object for the device whose presence is excluding the device for which the error occurred.
QLErrorIncompatibleInput	The input could not be added to the specified session because it is incompatible with existing inputs and outputs in the session.
QLErrorIncompatibleOutput	The output could not be added to the specified session because it is incompatible with existing inputs and outputs in the session.
QLErrorInvalidInputsOrOutputs	
QLErrorMaximumDurationReached	Returned when recording has reached the maximum duration specified by the application.
QLErrorMaximumFileSizeReached	Returned when recording has reached the maximum file size specified by the application.

<code>QLErrorMediaChanged</code>	The recording has been automatically stopped because the format of the input media changed or the media samples were invalid.
<code>QLErrorMediaDiscontinuity</code>	Returned when there is a discontinuity in captured media, usually because of performance problems on the user's system or because of a change in a device's state. This error generally indicates that media samples have been dropped in order to maintain real time capture.
<code>QLErrorNoDataCaptured</code>	Returned when no data was successfully captured during a recording or other capture operation.
<code>QLErrorRecordingSuccessfullyFinishedKey</code>	Use this key to determine whether the products of a recording were successfully finished after recording stopped due to an error. The value is an <code>NSNumber</code> interpreted as a <code>BOOL</code> .
<code>QLErrorSessionConfigurationChanged</code>	The recording has been automatically stopped because an input or output has been added or removed, or the channels of an input or output have changed.
<code>QLErrorUnknown</code>	Indicates an unexpected or unknown error.
<code>QTKitErrorDomain</code>	The QTKit error domain identifier.

## QTFormatDescription.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>QTFormatDescriptionAudioChannelLayoutAttribute</code>	Returns an <code>NSData</code> interpreted as a Core Audio <code>AudioChannelLayout</code> for audio media.
<code>QTFormatDescriptionAudioMagicCookieAttribute</code>	Returns an <code>NSData</code> interpreted as a Core Audio magic cookie for audio media.
<code>QTFormatDescriptionAudioStreamBasicDescriptionAttribute</code>	Returns an <code>NSValue</code> interpreted as a Core Audio <code>AudioStreamBasicDescription</code> for audio media.

QTFormatDescriptionVideoCleanApertureDisplaySizeAttribute	Returns an NSValue interpreted as an NSSize that indicates the size of video media displayed through its clean aperture and scaled by its pixel aspect ratio.
QTFormatDescriptionVideoEncodedPixelSizeAttribute	Returns an NSValue interpreted as an NSSize that indicates the encoded size of video media.
QTFormatDescriptionVideoProductionApertureDisplaySizeAttribute	Returns an NSValue interpreted as an NSSize that indicates the size of video media scaled by its pixel aspect ratio but not displayed through its clean aperture.

## QTKitDefines.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

AliasDataHandlerSubType	
AVAILABLE_MAC_OS_X_VERSION_10_0_AND_LATER_BUT_DEPRECATED_IN_MAC_OS_X_VERSION_10_5	
AVAILABLE_MAC_OS_X_VERSION_10_1_AND_LATER_BUT_DEPRECATED_IN_MAC_OS_X_VERSION_10_5	
AVAILABLE_MAC_OS_X_VERSION_10_2_AND_LATER_BUT_DEPRECATED_IN_MAC_OS_X_VERSION_10_5	
AVAILABLE_MAC_OS_X_VERSION_10_3_AND_LATER_BUT_DEPRECATED_IN_MAC_OS_X_VERSION_10_5	
AVAILABLE_MAC_OS_X_VERSION_10_4_AND_LATER_BUT_DEPRECATED_IN_MAC_OS_X_VERSION_10_5	
BaseMediaType	
codecHighQuality	High-quality image reproduction. This value should correspond to the highest image quality that can be achieved with reasonable performance.

## 10.5 Symbol Changes

<code>codecLosslessQuality</code>	Lossless compression or decompression. This special value is valid only for components that can support lossless compression or decompression.
<code>codecLowQuality</code>	Low-quality image reproduction. This value should correspond to the lowest image quality that still results in acceptable display characteristics.
<code>codecMaxQuality</code>	The maximum standard value.
<code>codecMinQuality</code>	The minimum valid value.
<code>codecNormalQuality</code>	Image reproduction of normal quality.
<code>FlashMediaType</code>	
<code>graphicsModeComposition</code>	
<code>graphicsModePerComponentAlpha</code>	
<code>graphicsModePreBlackAlpha</code>	
<code>graphicsModePreMulColorAlpha</code>	
<code>graphicsModePreWhiteAlpha</code>	
<code>graphicsModeStraightAlpha</code>	
<code>graphicsModeStraightAlphaBlend</code>	
<code>HandleDataHandlerSubType</code>	
<code>k16GrayCodecType</code>	
<code>k32AlphaGrayCodecType</code>	
<code>k422YpCbCr10CodecType</code>	Component Y'CbCr 10-bit 4:2:2.
<code>k422YpCbCr16CodecType</code>	Component Y'CbCr 10,12,14,16-bit 4:2:2.
<code>k422YpCbCr8CodecType</code>	Component Y'CbCr 8-bit 4:2:2.
<code>k4444YpCbCrA8CodecType</code>	Component Y'CbCrA 8-bit 4:4:4:4.
<code>k4444YpCbCrA8RCodecType</code>	Component Y'CbCrA 8-bit 4:4:4:4, rendering format. full range alpha, zero biased YUV.
<code>k444YpCbCr10CodecType</code>	Component Y'CbCr 10-bit 4:4:4.
<code>k444YpCbCr8CodecType</code>	Component Y'CbCr 8-bit 4:4:4.

## 10.5 Symbol Changes

k48RGBCodecType	
k64ARGBCodecType	
kAnimationCodecType	
kAVRJPEGCodecType	
kBaseCodecType	
kBMPCoecType	
kCinepakCodecType	
kCloudCodecType	
kCMYKCodecType	
kComponentVideoCodecType	
kComponentVideoSigned	For historical reasons, 'yuvu' identifies the signed type.
kComponentVideoUnsigned	For historical reasons, 'yuvs' identifies the unsigned type.
kDVCNTSCCodecType	
kDVCPALCodecType	
kDVCPPro100NTSCCodecType	
kDVCPPro100PALCodecType	
kDVCPPro50NTSCCodecType	
kDVCPPro50PALCodecType	
kDVCPROHD1080i50CodecType	
kDVCPROHD1080i60CodecType	
kDVCPROHD720pCodecType	
kDVCPProPALCodecType	
kFireCodecType	
kFLCCodecType	
kGIFCodecType	
kGraphicsCodecType	
kH261CodecType	

## 10.5 Symbol Changes

kH263CodecType	
kH264CodecType	
kIndeo4CodecType	
kJPEG2000CodecType	
kJPEGCodecType	
kMacPaintCodecType	
kMicrosoftVideo1CodecType	
kMotionJPEGACodecType	
kMotionJPEGBCodecType	
kMPEG4VisualCodecType	
kMpegYUV420CodecType	
kOpenDMLJPEGCodecType	
kPhotoCDCodecType	
kPxlletCodecType	
kPlanarRGBCodecType	
kPNGCodecType	
kQFileType3DMF	
kQFileType3GP2	
kQFileType3GPP	
kQFileTypeAIFC	
kQFileTypeAIFF	
kQFileTypeAMC	
kQFileTypeAMR	
kQFileTypeAudioCDTrack	
kQFileTypeAVI	
kQFileTypeBMP	
kQFileTypeDVC	
kQFileTypeFlash	

## 10.5 Symbol Changes

kQTFileTypeFlashPix	
kQTFileTypeFLC	
kQTFileTypeGIF	
kQTFileTypeJFIF	
kQTFileTypeJPEG	
kQTFileTypeJPEG2000	
kQTFileTypeMacPaint	
kQTFileTypeMIDI	
kQTFileTypeMovie	
kQTFileTypeMP4	
kQTFileTypeMuLaw	
kQTFileTypePDF	
kQTFileTypePhotoShop	
kQTFileTypePICS	
kQTFileTypePicture	
kQTFileTypePNG	
kQTFileTypeQuickDrawGXPicture	
kQTFileTypeQuickTimeImage	
kQTFileTypeSDV	
kQTFileTypeSGIImage	
kQTFileTypeSoundDesignerII	
kQTFileTypeSystemSevenSound	
kQTFileTypeTargaImage	
kQTFileTypeText	
kQTFileTypeTIFF	
kQTFileTypeWave	
kQTQuartzComposerMediaType	
kQuickDrawCodecType	

## 10.5 Symbol Changes

kQuickDrawGXCodecType	
kRawCodecType	
kSGICodecType	
kSorenson3CodecType	Available in QuickTime 5 and later.
kSorensonCodecType	
kSorensonYUV9CodecType	
kTargaCodecType	
kTIFFCodecType	
kVectorCodecType	
kVideoCodecType	
kWaterRippleCodecType	
kWindowsRawCodecType	
kYUV420CodecType	
MAC_OS_X_VERSION_10_4	
MAC_OS_X_VERSION_10_5	
MovieMediaType	
MPEGMediaType	
MusicMediaType	
NSInteger	Used to describe an integer.
NSIntegerDefined	
NSIntegerMax	The maximum value for an NSInteger.
NSIntegerMin	The minimum value for an NSInteger.
NSUInteger	Used to describe an unsigned integer.
NSUIntegerMax	The maximum value for an NSUInteger.
NullDataHandlerSubType	
PointerDataHandlerSubType	
QTKIT_HIDDEN	



QTKIT_VERSION_7_0	
QTKIT_VERSION_7_2	
QTKIT_VERSION_MAX_ALLOWED	
QTKIT_VERSION_MIN_REQUIRED	
ResourceDataHandlerSubType	
SkinMediaType	
SoundMediaType	Sound channel.
SpriteMediaType	
TextMediaType	Text media.
ThreeDeeMediaType	
TimeCode64MediaType	
TimeCodeMediaType	
TweenMediaType	
URLDataHandlerSubType	
VideoMediaType	Video channel.
WiredActionHandlerType	

## QTMedia.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

QTMediaTypeMuxed	Multiplexed audio and video media.
QTMediaTypeQuartzComposer	Quartz Composer media.

## QTMovie.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

## 10.5 Symbol Changes

QTIncludeDynamicTypes	
QTMovieApertureModeAttribute	
QTMovieApertureModeClassic	
QTMovieApertureModeClean	
QTMovieApertureModeDidChangeNotification	Issued when the aperture mode of the target QTMovie object changes.
QTMovieApertureModeEncodedPixels	
QTMovieApertureModeProduction	
QTMovieChapterName	
QTMovieChapterStartTime	
QTMovieChapterTargetTrackAttribute	
QTMovieFrameImageDeinterlaceFields	
QTMovieFrameImageHighQuality	
QTMovieFrameImageOpenGLContext	
QTMovieFrameImagePixelFormat	
QTMovieFrameImageRepresentationsType	
QTMovieFrameImageSingleField	
QTMovieFrameImageSize	
QTMovieFrameImageType	
QTMovieFrameImageTypeCGImageRef	
QTMovieFrameImageTypeCIImage	
QTMovieFrameImageTypeCVOpenGLTextureRef	
QTMovieFrameImageTypeCVPixelFormatRef	
QTMovieFrameImageTypeNSImage	
QTMovieHasApertureModeDimensionsAttribute	
QTMovieLoadState	
QTMovieLoadStateComplete	
QTMovieLoadStateError	

QTMovieLoadStateLoaded	
QTMovieLoadStateLoading	
QTMovieLoadStatePlayable	
QTMovieLoadStatePlaythroughOK	

## QTSampleBuffer.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

QTSampleBufferAudioBufferList-OptionAssure16ByteAlignment	
QTSampleBufferAudioBufferListOptions	
QTSampleBufferDataRecordedAttribute	Returns the date on which the media in the buffer was originally recorded.
QTSampleBufferExplicitSceneChange	Indicates that a scene change was explicitly marked in the sample buffer's metadata.
QTSampleBufferHostTimeAttribute	Returns the buffer's host time, if the buffer is from a real time source.
QTSampleBufferSceneChangeTypeAttribute	If the buffer marks a scene change in the input content, returns a constant.
QTSampleBufferSMPTETimeAttribute	Returns the SMPTE timecode of the sample buffer, if it has one.
QTSampleBufferTimeStampDiscontinuitySceneChange	Indicates that the scene changed due to a discontinuity in time stamps between the current sample buffer and the previous sample buffer.

## QTTime.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>QTStringFromSMPTETime</code>	Returns a human-readable string from the <code>SMPTETime</code> . The returned string is of the form <code>hh:mm:ss.ff</code> .
------------------------------------	---

## QTTrack.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>QTTrackFormatSummaryAttribute</code>	An <code>NSString</code> that is a localized, human-readable string that summarizes a track's format; for example, "16-bit Integer (Big Endian), Stereo (L R), 48.000 kHz". This attribute is gettable but not settable.
<code>QTTrackHasApertureModeDimensionsAttribute</code>	The value to determine whether aperture mode dimensions have been set on a track, even if they are all identical to the classic dimensions (as is the case for content with square pixels and no edge-processing region).
<code>QTTrackIsChapterTrackAttribute</code>	

# 10.4 Symbol Changes

---

This article lists the symbols added to `QTKit.framework` in Mac OS X v10.4.

## C Symbols

All of the header files with new symbols are listed alphabetically, with their new symbols described.

### QTDataReference.h

---

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>QTDataReferenceTypeFile</code>	The file type for a <code>QTDataReference</code> object.
<code>QTDataReferenceTypeHandle</code>	The handle type for a <code>QTDataReference</code> object.
<code>QTDataReferenceTypePointer</code>	The pointer type for a <code>QTDataReference</code> object.
<code>QTDataReferenceTypeResource</code>	The resource type for a <code>QTDataReference</code> object.
<code>QTDataReferenceTypeURL</code>	The URL type for a <code>QTDataReference</code> object.

### QTMedia.h

---

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>QTMediaCharacteristicAudio</code>	The media has audio data.
<code>QTMediaCharacteristicCanSendVideo</code>	The media can send visual data to another track.
<code>QTMediaCharacteristicCanStep</code>	The media can step.
<code>QTMediaCharacteristicHasNoDuration</code>	The media has no duration.

## 10.4 Symbol Changes

QTMediaCharacteristicHasSkinData	The media has skin data.
QTMediaCharacteristicHasVideoFrameRate	The media has a video frame rate.
QTMediaCharacteristicNonLinear	The media is non-linear.
QTMediaCharacteristicProvidesActions	The media has actions.
QTMediaCharacteristicProvidesKeyFocus	Key events can be focused at the media.
QTMediaCharacteristicVisual	The media has video data.
QTMediaCreationTimeAttribute	The creation time. The value for this key is of type NSDate.
QTMediaDurationAttribute	The duration. The value for this key is of type NSValue, interpreted as a QTime.
QTMediaModificationTimeAttribute	The modification time. The value for this key is of type NSDate.
QTMediaQualityAttribute	The media quality. The value for this key is of type NSNumber, interpreted as a short.
QTMediaSampleCountAttribute	The media sample count. The value for this key is of type NSNumber, interpreted as a long.
QTMediaTimeScaleAttribute	The media time scale. The value for this key is of type NSNumber, interpreted as a long.
QTMediaType3D	3D media.
QTMediaTypeAttribute	The media type. The value for this key is of type NSString. See “Media Types” for the values this attribute can return.
QTMediaTypeBase	Base media.
QTMediaTypeFlash	Flash media.
QTMediaTypeHint	Hint media.
QTMediaTypeMovie	Movie media.
QTMediaTypeMPEG	MPEG media.
QTMediaTypeMusic	Music media.
QTMediaTypeQTVR	QuickTime VR media.
QTMediaTypeSkin	Skin media
QTMediaTypeSound	Sound media.
QTMediaTypeSprite	Sprite media.

QTMediaTypeStream	Stream media.
QTMediaTypeText	Text media.
QTMediaTypeTimeCode	Timecode media.
QTMediaTypeTween	Tween media.
QTMediaTypeVideo	Video media.

## QTMovie.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

QTAddImageCodecQuality	
QTAddImageCodecType	
QTMovieActiveSegmentAttribute	
QTMovieAskUnresolvedDataRefsAttribute	
QTMovieAutoAlternatesAttribute	
QTMovieChapterDidChangeNotification	Issued when the chapter associated with QTMovie changes.
QTMovieChapterListDidChangeNotification	Issued when the chapter list associated with QTMovie changes.
QTMovieCloseWindowRequestNotification	Sent when a request is made to close the movie's window.
QTMovieCopyrightAttribute	
QTMovieCreationTimeAttribute	
QTMovieCurrentSizeAttribute	
QTMovieCurrentTimeAttribute	
QTMovieDataAttribute	
QTMovieDataReferenceAttribute	
QTMovieDataSizeAttribute	
QTMovieDelegateAttribute	

## 10.4 Symbol Changes

QTMovieDidEndNotification	Sent when the movie is “done” or at its end.
QTMovieDisplayNameAttribute	
QTMovieDontInteractWithUserAttribute	
QTMovieDurationAttribute	
QTMovieEditabilityDidChangeNotification	Sent when the editable state of a movie has changed.
QTMovieEditableAttribute	
QTMovieEditedNotification	Sent when a movie has been edited.
QTMovieEnterFullScreenRequestNotification	Sent when a request is made to play back a movie in full screen mode.
QTMovieExitFullScreenRequestNotification	Sent when a request is made to play back a movie in normal windowed mode.
QTMovieExport	
QTMovieExportManufacturer	
QTMovieExportSettings	
QTMovieExportType	
QTMovieFileNameAttribute	
QTMovieFileOffsetAttribute	
QTMovieFlatten	
QTMovieHasAudioAttribute	
QTMovieHasDurationAttribute	
QTMovieHasVideoAttribute	
QTMovieIsActiveAttribute	
QTMovieIsInteractiveAttribute	
QTMovieIsLinearAttribute	
QTMovieIsSteppableAttribute	
QTMovieLoadStateAttribute	
QTMovieLoadStateDidChangeNotification	Sent when the load state of a movie has changed.



## 10.4 Symbol Changes

QTMovieLoopModeDidChangeNotification	Sent when a change is made in a movie's looping mode.
QTMovieLoopsAttribute	
QTMovieLoopsBackAndForthAttribute	
QTMovieMessageNotificationParameter	
QTMovieMessageStringPostedNotification	Sent when a movie message has been received by the movie controller.
QTMovieModificationTimeAttribute	
QTMovieMutedAttribute	
QTMovieNaturalSizeAttribute	
QTMovieOpenAsyncOKAttribute	
QTMoviePasteboardAttribute	
QTMoviePasteboardType	
QTMoviePlaysAllFramesAttribute	
QTMoviePlaysSelectionOnlyAttribute	
QTMoviePosterTimeAttribute	
QTMoviePreferredMutedAttribute	
QTMoviePreferredRateAttribute	
QTMoviePreferredVolumeAttribute	
QTMoviePreviewModeAttribute	
QTMoviePreviewRangeAttribute	
QTMovieRateAttribute	
QTMovieRateChangesPreservePitchAttribute	
QTMovieRateDidChangeNotification	Sent when the rate of a movie has changed.
QTMovieRateDidChangeNotificationParameter	
QTMovieResolveDataRefsAttribute	
QTMovieSelectionAttribute	
QTMovieSelectionDidChangeNotification	Sent when the selection of a movie has changed.

QTMovieSizeDidChangeNotification	Sent when the size of a movie has changed.
QTMovieStatusCodeNotificationParameter	
QTMovieStatusFlagsNotificationParameter	
QTMovieStatusStringNotificationParameter	
QTMovieStatusStringPostedNotification	Status messages can be sent by QuickTime's streaming components or by any code that wants to display a message in the movie controller bar status area.
QTMovieTargetIDNotificationParameter	
QTMovieTargetNameNotificationParameter	
QTMovieTimeDidChangeNotification	Sent when the time in a movie has changed to a value other than what it would be during normal playback.
QTMovieTimeScaleAttribute	
QTMovieUneditableException	
QTMovieURLAttribute	
QTMovieVolumeAttribute	
QTMovieVolumeDidChangeNotification	Sent when the volume of a movie has changed.

## QTMovieView.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

QTMovieViewControllerVisibleBinding	
QTMovieViewFillColorBinding	
QTMovieViewMovieBinding	
QTMovieViewPreservesAspectRatioBinding	

## QTime.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

QTGetTimeInterval	Returns the value of a QTime structure expressed as an NSTimeInterval.
QTGetTimeRecord	Returns the value of a QTime structure expressed as a TimeRecord.
QTMakeTime	Creates a QTime structure.
QTMakeTimeScaled	Returns a QTime structure.
QTMakeTimeWithTimeInterval	Creates a QTime structure.
QTMakeTimeWithTimeRecord	Creates a QTime structure.
QTStringFromTime	Returns a description of a QTime structure.
QTimeCompare	Returns a value of type NSComparisonResult.
QTimeDecrement	Subtracts one QTime from another.
QTimeFromString	Returns a QTime structure.
QTimeIncrement	Adds two QTime structures.
QTimeIsIndefinite	

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

QTIndefiniteTime	
QTZeroTime	

## QTimeRange.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

QTEqualTimeRanges	Returns YES if the specified time ranges are identical.
QTIntersectionTimeRange	Returns a QTTimeRange structure that represents the intersection of the two ranges.
QTMakeTimeRange	Returns a QTTimeRange structure initialized using the QTTime structures time and duration.
QTStringFromTimeRange	Returns a description of a QTTimeRange structure.
QTTimeInTimeRange	Returns YES if the specified time time lies in the time range range.
QTTimeRangeEnd	Returns a QTTime structure representing the end of the specified time range.
QTTimeRangeFromString	Returns a QTTimeRange structure
QTUnionTimeRange	Returns a QTTimeRange structure.

## QTTrack.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

QTTrackBoundsAttribute	The bounding rectangle of a QTTrack object; the value for this key is of type NSValue, interpreted as an NSRect.
QTTrackCreationTimeAttribute	The creation time of a QTTrack object; the value for this key is of type NSDate.
QTTrackDimensionsAttribute	The dimensions of a QTTrack object; the value for this key is of type NSValue, interpreted as an NSSize.
QTTrackDisplayNameAttribute	The display name of a QTTrack object; the value for this key is of type NSString.
QTTrackEnabledAttribute	The track enabled state of a QTTrack object; the value for this key is of type NSNumber, interpreted as a BOOL.
QTTrackIDAttribute	The track ID of a QTTrack object; the value for this key is of type NSNumber, interpreted as a long.
QTTrackLayerAttribute	The track layer of a QTTrack object; the value for this key is of type NSNumber, interpreted as a short.
QTTrackMediaTypeAttribute	The media type of a QTTrack object; the value for this key is of type NSString.

QTTrackModificationTimeAttribute	The modification time of a QTTrack object; the value for this key is of type NSDate.
QTTrackRangeAttribute	The range of time this track occupies; the value for this key is of type NSValue, interpreted as a QTTimeRange.
QTTrackTimeScaleAttribute	The track time scale; the value for this key is of type NSNumber, interpreted as a long.
QTTrackUsageInMovieAttribute	The movie usage setting; the value for this key is of type NSNumber, interpreted as a BOOL.
QTTrackUsageInPosterAttribute	The poster usage setting; the value for this key is of type NSNumber, interpreted as a BOOL.
QTTrackUsageInPreviewAttribute	The preview usage setting; the value for this key is of type NSNumber, interpreted as a BOOL.
QTTrackVolumeAttribute	The volume of a QTTrack object; the value for this key is of type NSNumber, interpreted as a float.

## QTUtilities.h

---

### Functions

---

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

QTOSTypeForString	Returns a four-character code representing the specified NSString.
QTStringForOSType	Returns an NSString representing the specified four-character code type.



# 10.3 Symbol Changes

---

This article lists the symbols added to `QTKit.framework` in Mac OS X v10.3.

## Classes

All of the classes with new symbols are listed alphabetically, with their new class, instance, and delegate methods described.

### NSCoder

---

Complete reference information is available in the `NSCoder` reference.

#### Instance Methods

---

<code>decodeQTimeForKey:</code>	Decodes a <code>QTime</code> structure.
<code>decodeQTimeRangeForKey:</code>	Decodes a <code>QTimeRange</code> structure.
<code>encodeQTime:forKey:</code>	Encodes a <code>QTime</code> structure.
<code>encodeQTimeRange:forKey:</code>	Encodes a <code>QTimeRange</code> structure range.

### NSObject

---

Complete reference information is available in the `NSObject` reference.

#### Instance Methods

---

<code>externalMovie:</code>	This method is called, if implemented by a <code>QTMovie</code> delegate object, when an external movie needs to be found (usually for a wired action targeted at an external movie).
<code>movieShouldLoadData:</code>	
<code>movieShouldTask:</code>	If a <code>QTMovie</code> object has a delegate and that delegate implements this method, that method will be called before <code>QTKit</code> performs the standard idle processing on a movie.

## NSValue

---

Complete reference information is available in the [NSValue](#) reference.

### Class Methods

---

<code>valueWithQTime:</code>	Creates an <code>NSValue</code> object that wraps the specified <code>QTime</code> structure.
<code>valueWithQTimeRange:</code>	Creates an <code>NSValue</code> object that wraps the specified <code>QTimeRange</code> structure.

### Instance Methods

---

<code>QTimeRangeValue</code>	Returns a <code>QTimeRange</code> structure that contains the range in an <code>NSValue</code> object.
<code>QTimeValue</code>	Returns a <code>QTime</code> structure that contains the time in an <code>NSValue</code> object.

## QTDataReference (New)

---

Complete reference information is available in the [QTDataReference](#) reference.

### Class Methods

---

<code>dataReferenceWithDataRef:type:</code>	Creates a <code>QTDataReference</code> object of type <code>type</code> initialized with data from <code>dataRef</code> .
<code>dataReferenceWithDataRefData:type:</code>	Creates a <code>QTDataReference</code> object of type <code>type</code> initialized with data from <code>dataRefData</code> .
<code>dataReferenceWithReferenceToData:</code>	Creates a <code>QTDataReference</code> object for the data block data.
<code>dataReferenceWithReferenceToData:name:MIMEType:</code>	Creates a <code>QTDataReference</code> object for the data block data.
<code>dataReferenceWithReferenceToFile:</code>	Creates a <code>QTDataReference</code> object for the file <code>fileName</code> .
<code>dataReferenceWithReferenceToURL:</code>	Creates a <code>QTDataReference</code> object for the URL <code>url</code> .



## Instance Methods

<code>dataRef</code>	Returns the QuickTime data reference associated with a <code>QTDataReference</code> object.
<code>dataRefData</code>	Returns the QuickTime data reference data associated with a <code>QTDataReference</code> object, stored in an <code>NSData</code> object.
<code>dataRefType</code>	Returns the type of the data reference associated with a <code>QTDataReference</code> object.
<code>initWithDataRef:type:</code>	Initializes a newly created <code>QTDataReference</code> object with data from <code>dataRef</code> .
<code>initWithDataRefData:type:</code>	Initializes a newly created <code>QTDataReference</code> object with data from <code>dataRefData</code> .
<code>initWithReferenceToData:</code>	Initializes a newly created <code>QTDataReference</code> object for the data block data.
<code>initWithReferenceToData:name:MIMETYPE:</code>	Initializes a newly created <code>QTDataReference</code> object for the data block data.
<code>initWithReferenceToFile:</code>	Initializes a newly created <code>QTDataReference</code> object for the file <code>fileName</code> .
<code>initWithReferenceToURL:</code>	Initializes a newly created <code>QTDataReference</code> object for the URL <code>url</code> .
<code>MIMETYPE</code>	Returns the type in a MIME type extension associated with a <code>QTDataReference</code> object.
<code>name</code>	Returns the name in a filenames extension associated with a <code>QTDataReference</code> object.
<code>referenceData</code>	Returns the reference data of a <code>QTDataReference</code> object, that is, the <code>NSData</code> object passed to <code>initWithReferenceToData</code> or <code>initWithReferenceToData:name:MIMETYPE</code> .
<code>referenceFile</code>	Returns the file name of the data reference associated with a <code>QTDataReference</code> object.
<code>referenceURL</code>	Returns the URL of the data reference associated with a <code>QTDataReference</code> object.
<code>setDataRef:</code>	Sets the data reference data of a <code>QTDataReference</code> object to <code>dataRef</code> .
<code>setDataRefType:</code>	Sets the data reference type of a <code>QTDataReference</code> object to <code>type</code> .

## QTMedia (New)

---

Complete reference information is available in the `QTMedia` reference.

### Class Methods

---

<code>initWithQuickTimeMedia:error:</code>	Creates a new <code>QTMedia</code> object with QuickTime media data.
--	--

### Instance Methods

---

<code>attributeForKey:</code>	Returns the value of the specified media attribute.
<code>hasCharacteristic:</code>	Returns whether the media has the specified characteristic.
<code>initWithQuickTimeMedia:error:</code>	Initializes a new <code>QTMedia</code> object with QuickTime media data.
<code>mediaAttributes</code>	Returns a dictionary containing all of the media's attributes.
<code>quickTimeMedia</code>	Returns the QuickTime media associated with the media object.
<code>setAttribute:forKey:</code>	Sets the value of the specified media attribute.
<code>setMediaAttributes:</code>	Sets the media's attributes using the values from the supplied dictionary.
<code>track</code>	Returns the <code>QTTrack</code> object that contains the media.

## QTMovie (New)

---

Complete reference information is available in the `QTMovie` reference.

### Class Methods

---

<code>canInitWithDataReference:</code>	Returns YES if the specified data reference can be used to initialize a <code>QTMovie</code> object.
<code>canInitWithFile:</code>	Returns YES if the contents of the specified file can be used to initialize a <code>QTMovie</code> object.
<code>canInitWithPasteboard:</code>	Returns YES if the contents of the specified pasteboard can be used to initialize a <code>QTMovie</code> object.
<code>canInitWithURL:</code>	Returns YES if the contents of the specified URL can be used to initialize a <code>QTMovie</code> object.

<code>movie</code>	Creates an empty QTMovie object.
<code>movieFileTypes:</code>	Returns an array of file types that can be opened as QuickTime movies.
<code>movieNamed:error:</code>	Creates a QTMovie object initialized with the data from the QuickTime movie of the specified name in the application's bundle.
<code>movieUnfilteredFileTypes</code>	Returns an array of file types that can be used to initialize a QTMovie object.
<code>movieUnfilteredPasteboardTypes</code>	Returns an array of pasteboard types that can be used to initialize a QTMovie object.
<code>movieWithAttributes:error:</code>	Creates a QTMovie object initialized with the attributes specified in attributes.
<code>movieWithData:error:</code>	Creates a QTMovie object initialized with the data specified by data.
<code>movieWithDataReference:error:</code>	Creates a QTMovie object initialized with the data specified by the data reference dataReference.
<code>movieWithFile:error:</code>	Creates a QTMovie object initialized with the data in the file specified by the name fileName.
<code>movieWithPasteboard:error:</code>	Creates a QTMovie object initialized with the contents of the pasteboard specified by pasteboard.
<code>movieWithQuickTimeMovie:disposeWhenDone:error:</code>	Creates a QTMovie object initialized with the data from an existing QuickTime movie movie.
<code>movieWithURL:error:</code>	Creates a QTMovie object initialized with the data in the URL specified by url.

### Instance Methods

---

<code>addImage:forDuration:withAttributes:</code>	Adds an image for the specified duration to the receiver, using attributes specified in the attributes dictionary.
<code>appendSelectionFromMovie:</code>	Appends to a QTMovie the current selection in movie.
<code>attributeForKey:</code>	Returns the current value of the movie attribute attributeKey.

<code>canUpdateMovieFile</code>	Indicates whether a movie file can be updated with changes made to the movie object.
<code>currentFrameImage</code>	Returns an <code>NSImage</code> for the frame at the current time in a <code>QTMovie</code> .
<code>currentTime</code>	Returns the current time of a <code>QTMovie</code> object as a structure of type <code>QTTime</code> .
<code>delegate</code>	Returns the delegate of a <code>QTMovie</code> object.
<code>deleteSegment:</code>	Deletes from a <code>QTMovie</code> the segment delimited by segment.
<code>duration</code>	Returns the duration of a <code>QTMovie</code> object as a structure of type <code>QTTime</code> .
<code>frameImageAtTime:</code>	Returns an <code>NSImage</code> for the frame at the time time in a <code>QTMovie</code> .
<code>generateApertureModeDimensions</code>	Adds information to a <code>QTMovie</code> needed to support aperture modes for tracks created with applications and/or versions of QuickTime that did not support aperture mode dimensions.
<code>gotoBeginning</code>	Repositions the play position to the beginning of the movie.
<code>gotoEnd</code>	Repositions the play position to the end of the movie.
<code>gotoNextSelectionPoint</code>	Repositions the movie to the next selection point.
<code>gotoPosterTime</code>	Repositions the play position to the movie's poster time.
<code>gotoPreviousSelectionPoint</code>	Repositions the movie to the previous selection point.
<code>initWithAttributes:error:</code>	Initializes a <code>QTMovie</code> object with the attributes specified in attributes.
<code>initWithData:error:</code>	Initializes a <code>QTMovie</code> object with the data specified by data.
<code>initWithDataReference:error:</code>	Initializes a <code>QTMovie</code> object with the data reference setting specified by dataReference.
<code>initWithFile:error:</code>	Initializes a <code>QTMovie</code> object with the data in the file specified by the name fileName.

<code>initWithMovie:timeRange:error:</code>	Initializes a QTMovie object with some or all of the data from an existing QTMovie object movie.
<code>initWithPasteboard:error:</code>	Initializes a QTMovie object with the contents of the pasteboard specified by pasteboard.
<code>initWithQuickTimeMovie:disposeWhenDone:error:</code>	Initializes a QTMovie object with the data from an existing QuickTime movie movie.
<code>initWithURL:error:</code>	Initializes a QTMovie object with the data in the URL specified by url.
<code>insertEmptySegmentAt:</code>	inserts into a QTMovie an empty segment delimited by the range range.
<code>insertSegmentOfMovie:fromRange:scaledToRange:</code>	Inserts the specified segment from the movie into the receiver, scaled to the range dstRange.
<code>insertSegmentOfMovie:timeRange:atTime:</code>	Inserts into a QTMovie at time time the selection in movie delimited by the time range range.
<code>movieAttributes</code>	Returns a dictionary containing the current values of all defined movie attributes.
<code>movieFormatRepresentation</code>	Returns the movie's data in an NSData object.
<code>movieWithTimeRange:error:</code>	Returns a QTMovie object whose data is the data in the specified time range.
<code>muted</code>	Returns the movie's mute setting.
<code>play</code>	Plays the movie.
<code>posterImage</code>	Returns an NSImage for the poster frame of a QTMovie.
<code>quickTimeMovie</code>	Returns the QuickTime movie associated with a QTMovie object.
<code>quickTimeMovieController</code>	Returns the QuickTime movie controller associated with a QTMovie object.
<code>rate</code>	Returns the current rate of a QTMovie object.
<code>removeApertureModeDimensions</code>	Removes aperture mode dimension information from a movie's tracks.
<code>replaceSelectionWithSelectionFromMovie:</code>	Replaces the current selection in a QTMovie with the current selection in movie.

<code>scaleSegment:newDuration:</code>	Scales the QTMovie segment delimited by the segment segment so that it will have the new duration newDuration.
<code>selectionDuration</code>	Returns the duration of the movie's current selection as a QTTime structure.
<code>selectionEnd</code>	Returns the end point of the movie's current selection as a QTTime structure.
<code>selectionStart</code>	Returns the start time of the movie's current selection as a QTTime structure.
<code>setAttribute:forKey:</code>	Set the movie attribute attributeKey to the value specified by the value parameter.
<code>setCurrentTime:</code>	Sets the movie's current time setting to time.
<code>setDelegate:</code>	Sets the movie's delegate to delegate.
<code>setMovieAttributes:</code>	Set the movie attributes using the key-value pairs specified in the dictionary attributes.
<code>setMuted:</code>	Sets the movie's mute setting to mute.
<code>setRate:</code>	Sets the movie's rate to rate.
<code>setSelection:</code>	Sets the movie's selection to selection.
<code>setVolume:</code>	Sets the movie's volume to volume.
<code>stepBackward</code>	Sets the movie backward a single frame.
<code>stepForward</code>	Sets the movie forward a single frame.
<code>stop</code>	Stops the movie playing.
<code>tracks</code>	Returns an array of QTTrack objects associated with the receiver.
<code>tracksOfMediaType:</code>	Returns an array of tracks with the specified media type.
<code>updateMovieFile</code>	Updates the movie file of a QTMovie.
<code>volume</code>	Returns the movie's volume as a scalar value of type float.
<code>writeToFile:withAttributes:</code>	Returns YES if the movie file was successfully created and NO otherwise.

### Delegate Methods

---

<code>movie:linkToURL:</code>	Called to handle the mcAction mcActionLinkToURL.
-------------------------------	--

<code>movie:shouldContinueOperation:withPhase:atPercent:withAttributes:</code>	If implemented, this method is called periodically during lengthy operations (such as exporting a movie).
--	---

## QTMovieView (New)

Complete reference information is available in the `QTMovieView` reference.

### Instance Methods

<code>add:</code>	This action method adds the contents of the clipboard to the movie at the current movie time.
<code>addScaled:</code>	This action method adds the contents of the clipboard to the movie, scaled to fit into the current movie selection.
<code>controllerBarHeight</code>	Returns the height of the controller bar.
<code>copy:</code>	This action method copies the current movie selection onto the clipboard.
<code>cut:</code>	This action method deletes the current movie selection from the movie, placing it on the clipboard.
<code>delete:</code>	This action method deletes the current movie selection from the movie, placing it on the clipboard.
<code>fillColor</code>	Returns the fill color of the <code>QTMovieView</code> .
<code>gotoBeginning:</code>	This action method sets the current movie time to the beginning of the movie.
<code>gotoEnd:</code>	This action method sets the current movie time to the end of the movie.
<code>gotoNextSelectionPoint:</code>	This action method sets the current movie time to the next selection point.
<code>gotoPosterFrame:</code>	This action method sets the current movie time to the movie poster frame.
<code>gotoPreviousSelectionPoint:</code>	This action method sets the current movie time to the previous selection point.
<code>initWithFrame:</code>	Initializes a newly allocated <code>QTMovieView</code> with <code>frame</code> as its frame rectangle.
<code>isControllerVisible</code>	Returns YES if the movie controller bar of the <code>QTMovieView</code> object is visible.
<code>isEditable</code>	Returns YES if the <code>QTMovieView</code> object is editable.

<code>movie</code>	Returns the <code>QTMovie</code> object associated with the <code>QTMovieView</code> .
<code>movieBounds</code>	Returns the rectangle currently occupied by the movie in a <code>QTMovieView</code> .
<code>movieControllerBounds</code>	Returns the rectangle currently occupied by the movie controller bar (if it's visible) in a <code>QTMovieView</code> .
<code>paste:</code>	This action method inserts the contents of the clipboard (if it contains a movie clip) into the movie at the current play position.
<code>pause:</code>	This action method pauses the movie playback.
<code>play:</code>	This action method starts the movie playing at its current location.
<code>preservesAspectRatio</code>	Returns YES if the <code>QTMovieView</code> object maintains the aspect ratio of the movie when drawing it in the view.
<code>replace:</code>	This action method replaces the current movie selection with the contents of the clipboard.
<code>selectAll:</code>	This action method selects the entire movie.
<code>selectNone:</code>	This action method selects nothing.
<code>setControllerVisible:</code>	Sets the visibility state of the movie controller bar in a <code>QTMovieView</code> to <code>controllerVisible</code> .
<code>setEditable:</code>	Sets the edit state of a <code>QTMovieView</code> to <code>editable</code> .
<code>setFillColor:</code>	Sets the fill color of a <code>QTMovieView</code> to <code>fillColor</code> .
<code>setMovie:</code>	Sets the <code>QTMovie</code> object in a <code>QTMovieView</code> to <code>movie</code> .
<code>setPreservesAspectRatio:</code>	Sets the aspect ratio state of a <code>QTMovieView</code> to <code>preservesAspectRatio</code> .
<code>setShowsResizeIndicator:</code>	Shows or hides the movie controller grow box.
<code>stepBackward:</code>	This action method steps the movie backward one frame.
<code>stepForward:</code>	This action method steps the movie forward one frame.
<code>trim:</code>	This action method trims the movie to the current movie selection.

## QTTrack (New)

---

Complete reference information is available in the `QTTrack` reference.

### Class Methods

---

<code>trackWithQuickTimeTrack:error:</code>	Creates a <code>QTTrack</code> object with data from the QuickTime track track.
---	---



## Instance Methods

<code>addImage:forDuration:withAttributes:</code>	Adds an image for the specified duration to the receiver, using attributes specified in the attributes dictionary.
<code>apertureModeDimensionsForMode:</code>	Returns an NSSize value that indicates the dimensions of the target track for the specified movie aperture mode.
<code>attributeForKey:</code>	Returns the current value of the track attribute <code>attributeKey</code> .
<code>deleteSegment:</code>	Deletes from a QTTrack the segment delimited by <code>segment</code> .
<code>generateApertureModeDimensions</code>	Adds information to a QTTrack needed to support aperture modes for tracks created with applications and/or versions of QuickTime that did not support aperture mode dimensions.
<code>initWithQuickTimeTrack:error:</code>	If a QTTrack object cannot be created, an NSError object is returned in the location pointed to by <code>errorPtr</code> .
<code>insertEmptySegmentAt:</code>	Inserts into a QTTrack an empty segment delimited by the range <code>range</code> .
<code>insertSegmentOfTrack:fromRange:scaledToRange:</code>	Inserts the specified segment from the track into the receiver, scaled to the range <code>dstRange</code> .
<code>insertSegmentOfTrack:timeRange:atTime:</code>	Inserts into a QTTrack at time <code>time</code> the selection in movie delimited by the time range <code>range</code> .
<code>isEnabled</code>	Returns YES if the QTTrack object is currently enabled, NO otherwise.
<code>media</code>	Returns the media associated with a QTTrack object.
<code>movie</code>	Returns the movie that contains a QTTrack object.
<code>quickTimeTrack</code>	Returns the QuickTime track associated with a QTTrack object.
<code>removeApertureModeDimensions</code>	Removes aperture mode dimension information from the target track.

<code>scaleSegment:newDuration:</code>	Scales the QTTrack segment delimited by the segment <code>segment</code> so that it will have the new duration <code>newDuration</code> .
<code>setApertureModeDimensions:forMode:</code>	Sets the dimensions of the target track for the specified movie aperture mode.
<code>setAttribute:forKey:</code>	Set the track attribute <code>attributeKey</code> to the value specified by the <code>value</code> parameter.
<code>setEnabled:</code>	Sets the enabled state of a QTTrack to enabled.
<code>setTrackAttributes:</code>	Set the track attributes using the key-value pairs specified in the dictionary <code>attributes</code> .
<code>setVolume:</code>	Sets the volume of a QTTrack to <code>volume</code> .
<code>trackAttributes</code>	Returns a dictionary containing the current values of all defined track attributes.
<code>volume</code>	Returns the volume of a QTTrack object.

## C Symbols

All of the header files with new symbols are listed alphabetically, with their new symbols described.

### QTKit.h

---

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>AVAILABLE_MAC_OS_X_VERSION_10_5_AND_LATER</code>	
--	--

### QTKitDefines.h

---

#### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

<code>QTKIT_EXTERN</code>	
---------------------------	--

## QTMovie.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

QTIncludeAggressiveTypes	Available in Mac OS X v10.3 and later.
QTIncludeAllTypes	Available in Mac OS X v10.3 and later.
QTIncludeCommonTypes	Available in Mac OS X v10.3 and later.
QTIncludeStillImageTypes	Available in Mac OS X v10.3 and later.
QTIncludeTranslatableTypes	Available in Mac OS X v10.3 and later.
QTMovieFileTypeOptions	
QTMovieOperationBeginPhase	
QTMovieOperationEndPhase	
QTMovieOperationPhase	
QTMovieOperationUpdatePercentPhase	

## QTTime.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

kQTTimeIsIndefinite	
QTTime	

## QTTimeRange.h

---

### Data Types & Constants

---

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

QTTimeRange	Defines a range of time.
-------------	--------------------------



# Document Revision History

---

This table describes the changes to *QTKit Reference Update*.

Date	Notes
2007-07-18	Updated with the symbols added to the QuickTime Kit framework in Mac OS X v10.5.
2005-04-29	New document that summarizes the symbols added to the QuickTime Kit framework in Mac OS X v10.4.

