QuickTime Reference Update

QuickTime

2007-07-18

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

Simultaneously published in the United States and Canada.

Contents

Introduction to QuickTime Reference Update

This document summarizes the symbols that have been added to the QuickTime 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 *QuickTime Framework Reference*.

10.5 Symbol Changes

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

C Symbols

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

ImageCompression.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.

`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`	

MediaHandlers.h

Functions

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

`MediaGenerateApertureModeDimensions`	Examines media and sets up track aperture mode dimensions.
`MediaGetApertureModeClipRect-ForSampleDescriptionIndex`	Calculates a source clip rectangle appropriate for the current aperture mode and the given sample description.
`MediaGetApertureModeMatrix-ForSampleDescriptionIndex`	Calculates a matrix appropriate for the current aperture mode and the given sample description.

`MediaSetTrackApertureModeDimensions-UsingSampleDescription`	Sets the three aperture mode dimension properties on the track, calculating the values using the provided sample description.

MediaHandlers.k.h

Functions

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

`SetTrackApertureModeDimensions-UsingSampleDescription`	Sets a track's aperture mode dimensions using values calculated using a sample description.

Movies.h

Functions

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

`GenerateMovieApertureModeDimensions`	Examines a movie and sets up track aperture mode dimensions.
`GenerateTrackApertureModeDimensions`	Examines a track and sets up aperture mode dimensions.
`QTAudioContextRegisterInsert`	
`QTMetaDataCreateFromBuffer`	
`QTMetaDataGetBytes`	
`RemoveMovieApertureModeDimensions`	Removes aperture mode dimension information from a movie.
`RemoveTrackApertureModeDimensions`	Removes aperture mode dimension information from a track.

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.

`AudioContextInsertFinalizeCallback`	

AudioContextInsertProcessDataCallback	
AudioContextInsertResetCallback	
kQTAudioPropertyID_RegisterAudioContextInsert	
kQTAudioRenderQuality_High	
kQTAudioRenderQuality_Low	
kQTAudioRenderQuality_Max	
kQTAudioRenderQuality_Medium	
kQTAudioRenderQuality_Min	
kQTMetaDataTypeQuickTimeMetaData	
kQTMovieAudioExtractionAudioPropertyID_RegisterMovieInsert	
kQTMovieAudioExtractionAudioPropertyID_RegisterTrackInsert	
kQTMovieAudioExtractionAudioPropertyID_RemainingAudioDuration	
kQTMovieAudioExtractionMoviePropertyID_RegisterInsert	
kQTMovieImporterPropertyID_AllowMediaOptimization	
kQTMovieInstantiationPropertyID_AllowMediaOptimization	
kQTPerformancePropertyID_AudioIOOverloadCount	
kQTPerformancePropertyID_MediaStallCount	
kQTPropertyClass_MovieImporter	
kQTPropertyClass_Performance	
QTAudioContextInsertRegistryInfo	
QTAudioContextInsertRegistryInfoRef	
QTAudioTrackInsertRegistryInfo	
QTAudioTrackInsertRegistryInfoRef	
TimeCode64MediaType	

MoviesFormat.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.

kQTSampleDependency_DependsOnOthers_Corrected	
kQTSampleDependency_DoesNotDependOnOthers_Corrected	
kQTSampleDependency_HasNoRedundantCoding_Corrected	
kQTSampleDependency_HasRedundantCoding_Corrected	
kQTSampleDependency_IsDependedOnByOthers_Corrected	
kQTSampleDependency_IsNotDependedOnByOthers_Corrected	

QuickTimeComponents.h

Functions

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

SCAudioFillBuffer	Used is used to pull compressed frames from the StdAudio component in kQTSCAudioPropertyID_BasicDescription format.
SCAudioReset	
TCFrameNumberToTimeCodeCounter	
TCFrameNumberToTimeCodeTime	
TCGetCurrentFrameAndTimeCodeDef	
TCGetFrameAndTimeCodeDefAtTime	
TCTimeCodeCounterToFrameNumber	
TCTimeCodeCounterToString	
TCTimeCodeTimeToFrameNumber	
TCTimeCodeTimeToString	

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.

kQTSCAudioPropertyID_PreviewSourceMovie	
kQTSCAudioPropertyID_PreviewSourceTrack	
kQTSGAudioPropertyID_CodecSpecificSettingsArray	
kTCFrameNumberToTimeCodeCounterSelect	
kTCFrameNumberToTimeCodeTimeSelect	
kTCGetCurrentFrameAndTimeCodeDefSelect	
kTCGetFrameAndTimeCodeDefAtTimeSelect	
kTCTimeCodeCounterToFrameNumberSelect	
kTCTimeCodeCounterToStringSelect	
kTCTimeCodeTimeToFrameNumberSelect	
kTCTimeCodeTimeToStringSelect	
sgcAudioCodecSpecificSettingsArray	
TimeCode64Counter	

QuickTimeErrors.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.

vdDontHaveThatUniqueIDErr	
vdHardwareGoneErr	
vdImageDescStaleErr	

10.4 Symbol Changes

This article lists the symbols added to `QuickTime.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.

HIMovieView.h

Functions

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

`HIMovieViewChangeAttributes`	Changes the views attributes.
`HIMovieViewCreate`	Creates an HIMovieView object.
`HIMovieViewGetAttributes`	Returns the view's current attributes.
`HIMovieViewGetControllerBarSize`	Returns the size of the visible movie controller bar.
`HIMovieViewGetMovie`	Returns the view's current movie.
`HIMovieViewGetMovieController`	Returns the view's current movie controller.
`HIMovieViewPause`	Pauses the view's current movie.
`HIMovieViewPlay`	Plays the view's current movie.
`HIMovieViewSetMovie`	Sets the view's current movie.

ImageCodec.h

Functions

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

ImageCodecBeginPass	Notifies the compressor that it should operate in multipass mode and use the given multipass storage.
ImageCodecCompleteFrame	Directs the compressor to finish with a queued source frame, either emitting or dropping it.
ImageCodecDecodeBand	Returns an ImageSubCodecDecompressRecord structure for an image codec component.
ImageCodecEncodeFrame	Presents the compressor with a frame to encode.
ImageCodecEndPass	
ImageCodecGetPreferredChunkSizeAndAlignment	
ImageCodecPrepareToCompressFrames	Prepares the compressor to receive frames.
ImageCodecProcessBetweenPasses	Provides the compressor with an opportunity to perform processing between passes.

ImageCompression.h

Functions

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

GetDSequenceNonScheduledDisplayDirection	Returns the display direction for a decompress sequence.
GetDSequenceNonScheduledDisplayTime	Gets the display time for a decompression sequence.
ICMCompressionFrameOptionsCreate	Creates a frame compression options object.
ICMCompressionFrameOptionsCreateCopy	Copies a frame compression options object.
ICMCompressionFrameOptionsGetForceKeyFrame	Retrieves the force key frame flag.
ICMCompressionFrameOptionsGetFrameType	Retrieves the frame type setting.
ICMCompressionFrameOptionsGetProperty	Retrieves the value of a specific property of a compression frame options object.
ICMCompressionFrameOptionsGetPropertyInfo	Retrieves information about properties of a compression frame options object.

ICMCompressionFrameOptionsGetTypeID	Returns the type ID for the current frame compression options object.
ICMCompressionFrameOptionsRelease	Decrements the retain count of a frame compression options object.
ICMCompressionFrameOptionsRetain	Increments the retain count of a frame compression options object.
ICMCompressionFrameOptionsSetForceKeyFrame	Forces frames to be compressed as key frames.
ICMCompressionFrameOptionsSetFrameType	Requests a frame be compressed as a particular frame type.
ICMCompressionFrameOptionsSetProperty	Sets the value of a specific property of a compression frame options object.
ICMCompressionSessionBeginPass	Announces the start of a specific compression pass.
ICMCompressionSessionCompleteFrames	Forces a compression session to complete encoding frames.
ICMCompressionSessionCreate	Creates a compression session for a specified codec type.
ICMCompressionSessionEncodeFrame	Presents video frames to a compression session.
ICMCompressionSessionEndPass	Announces the end of a pass.
ICMCompressionSessionGetImageDescription	Retrieves the image description for a video compression session.
ICMCompressionSessionGetPixelBufferPool	Returns a pool that can provide ideal source pixel buffers for a compression session.
ICMCompressionSessionGetProperty	Retrieves the value of a specific property of a compression session.
ICMCompressionSessionGetPropertyInfo	Retrieves information about properties of a compression session.
ICMCompressionSessionGetTimeScale	Retrieves the time scale for a compression session.
ICMCompressionSessionGetTypeID	Returns the type ID for the current compression session.
ICMCompressionSessionOptionsCreate	Creates a compression session options object.

ICMCompressionSessionOptionsCreateCopy	Copies a compression session options object.
ICMCompressionSessionOptionsGetAllowFrameReordering	Retrieves the allow frame reordering flag.
ICMCompressionSessionOptionsGetAllowFrameTimeChanges	Retrieves the allow frame time changes flag.
ICMCompressionSessionOptions-GetAllowTemporalCompression	Retrieves the allow temporal compression flag.
ICMCompressionSessionOptionsGetDurationsNeeded	Retrieves the durations needed flag.
ICMCompressionSessionOptionsGetMaxKeyFrameInterval	Retrieves the maximum key frame interval.
ICMCompressionSessionOptionsGetProperty	Retrieves the value of a specific property of a compression session options object.
ICMCompressionSessionOptionsGetPropertyInfo	Retrieves information about properties of a compression session options object.
ICMCompressionSessionOptionsGetTypeID	Returns the type ID for the current compression session options object.
ICMCompressionSessionOptionsRelease	Decrements the retain count of a compression session options object.
ICMCompressionSessionOptionsRetain	Increments the retain count of a compression session options object.
ICMCompressionSessionOptionsSetAllowFrameReordering	Enables frame reordering.
ICMCompressionSessionOptionsSetAllowFrameTimeChanges	Allows the compressor to modify frame times.
ICMCompressionSessionOptions-SetAllowTemporalCompression	Enables temporal compression.
ICMCompressionSessionOptionsSetDurationsNeeded	Indicates that the durations of outputted frames must be calculated.
ICMCompressionSessionOptionsSetMaxKeyFrameInterval	Sets the maximum interval between key frames.
ICMCompressionSessionOptionsSetProperty	Sets the value of a specific property of a compression session options object.
ICMCompressionSessionProcessBetweenPasses	Lets the compressor perform processing between passes.

`ICMCompressionSessionRelease`	Decrements the retain count of a compression session.
`ICMCompressionSessionRetain`	Increments the retain count of a compression session.
`ICMCompressionSessionSetProperty`	Sets the value of a specific property of a compression session.
`ICMCompressionSessionSupportsMultiPassEncoding`	Queries whether a compression session supports multipass encoding.
`ICMCompressorSessionDropFrame`	Called by a compressor to notify the ICM that a source frame has been dropped and will not contribute to any encoded frames.
`ICMCompressorSessionEmitEncodedFrame`	Called by a compressor to output an encoded frame corresponding to one or more source frames.
`ICMCompressorSourceFrameDetachPixelBuffer`	
`ICMCompressorSourceFrameGetDisplayNumber`	Retrieves a source frames display number.
`ICMCompressorSourceFrameGetDisplay-TimeStampAndDuration`	Retrieves the display time stamp and duration of a source frame.
`ICMCompressorSourceFrameGetFrameOptions`	Retrieves the frame compression options for a source frame.
`ICMCompressorSourceFrameGetPixelBuffer`	Retrieves a source frames pixel buffer.
`ICMCompressorSourceFrameGetTypeID`	Returns the type ID for the current source frame object.
`ICMCompressorSourceFrameRelease`	Decrements the retain count of a source frame object.
`ICMCompressorSourceFrameRetain`	Increments the retain count of a source frame object.
`ICMDecompressionFrameOptionsCreate`	Creates a frame decompression options object.
`ICMDecompressionFrameOptionsCreateCopy`	Copies a frame decompression options object.
`ICMDecompressionFrameOptionsGetProperty`	Retrieves the value of a specific property of a decompression frame options object.

ICMDecompressionFrameOptionsGetPropertyInfo	Retrieves information about properties of a decompression frame options object.
ICMDecompressionFrameOptionsGetTypeID	Returns the type ID for the current frame decompression options object.
ICMDecompressionFrameOptionsRelease	Decrements the retain count of a frame decompression options object.
ICMDecompressionFrameOptionsRetain	Increments the retain count of a frame decompression options object.
ICMDecompressionFrameOptionsSetProperty	Sets the value of a specific property of a decompression frame options object.
ICMDecompressionSessionCreate	Creates a session for decompressing video frames.
ICMDecompressionSessionCreateForVisualContext	Creates a session for decompressing video frames.
ICMDecompressionSessionDecodeFrame	Queues a frame for decompression.
ICMDecompressionSessionFlush	Flushes the frames queued for a decompression session.
ICMDecompressionSessionGetProperty	Retrieves the value of a specific property of a decompression session.
ICMDecompressionSessionGetPropertyInfo	Retrieves information about the properties of a decompression session.
ICMDecompressionSessionGetTypeID	Returns the type ID for the current decompression session.
ICMDecompressionSessionOptionsCreate	Creates a decompression session options object.
ICMDecompressionSessionOptionsCreateCopy	Copies a decompression session options object.
ICMDecompressionSessionOptionsGetProperty	Retrieves the value of a specific property of a decompression session options object.
ICMDecompressionSessionOptionsGetPropertyInfo	Retrieves information about properties of a decompression session options object.
ICMDecompressionSessionOptionsGetTypeID	Returns the type ID for the current decompression session options object.
ICMDecompressionSessionOptionsRelease	Decrements the retain count of a decompression session options object.

ICMDecompressionSessionOptionsRetain	Increments the retain count of a decompression session options object.
ICMDecompressionSessionOptionsSetProperty	Sets the value of a specific property of a decompression session options object.
ICMDecompressionSessionRelease	Decrements the retain count of a decompression session.
ICMDecompressionSessionRetain	Increments the retain count of a decompression session.
ICMDecompressionSessionSetNon-ScheduledDisplayDirection	Sets the direction for non-scheduled display time.
ICMDecompressionSessionSetNonScheduledDisplayTime	Sets the display time for a decompression session, and requests display of the non-scheduled queued frame at that display time, if there is one.
ICMDecompressionSessionSetProperty	Sets the value of a specific property of a decompression session.
ICMEncodedFrameCreateMutable	Called by a compressor to create an encoded-frame token corresponding to a given source frame.
ICMEncodedFrameGetBufferSize	Gets the size of an encoded frame's data buffer.
ICMEncodedFrameGetDataPtr	Gets the data buffer for an encoded frame.
ICMEncodedFrameGetDataSize	Gets the data size of the compressed frame in an encoded frame's buffer.
ICMEncodedFrameGetDecodeDuration	Retrieves an encoded frame's decode duration.
ICMEncodedFrameGetDecodeNumber	Retrieves the decode number of an encoded frame.
ICMEncodedFrameGetDecodeTimeStamp	Retrieves an encoded frame's decode time stamp.
ICMEncodedFrameGetDisplayDuration	Retrieves an encoded frame's display duration.
ICMEncodedFrameGetDisplayOffset	Retrieves an encoded frame's display offset.
ICMEncodedFrameGetDisplayTimeStamp	Retrieves an encoded frame's display time stamp.

`ICMEncodedFrameGetFrameType`	Retrieves the frame type for an encoded frame.
`ICMEncodedFrameGetImageDescription`	Retrieves the image description of an encoded frame.
`ICMEncodedFrameGetMediaSampleFlags`	Retrieves the media sample flags for an encoded frame.
`ICMEncodedFrameGetSimilarity`	Retrieves the similarity value for an encoded frame.
`ICMEncodedFrameGetSourceFrameRefCon`	Retrieves the reference value of an encoded frame's source frame.
`ICMEncodedFrameGetTimeScale`	Retrieves the timescale of an encoded frame.
`ICMEncodedFrameGetTypeID`	Returns the type ID for the current encoded frame object.
`ICMEncodedFrameGetValidTimeFlags`	Retrieves an encoded frame's flags indicating which of its time stamps and durations are valid.
`ICMEncodedFrameRelease`	Decrements the retain count of an encoded frame object.
`ICMEncodedFrameRetain`	Increments the retain count of an encoded frame object.
`ICMEncodedFrameSetDataSize`	Sets the data size of the compressed frame in an encoded frame's buffer.
`ICMEncodedFrameSetDecodeDuration`	Sets an encoded frame's decode duration.
`ICMEncodedFrameSetDecodeTimeStamp`	Sets an encoded frame's decode time stamp.
`ICMEncodedFrameSetDisplayDuration`	Sets an encoded frame's display duration.
`ICMEncodedFrameSetDisplayTimeStamp`	Sets an encoded frame's display time stamp.
`ICMEncodedFrameSetFrameType`	Sets the frame type for an encoded frame.
`ICMEncodedFrameSetMediaSampleFlags`	Sets the media sample flags for an encoded frame.
`ICMEncodedFrameSetSimilarity`	Sets the similarity for an encoded frame.

ICMEncodedFrameSetValidTimeFlags	Sets an encoded frame's flags that indicate which of its time stamps and durations are valid.
ICMImageDescriptionGetProperty	Returns a particular property of a image description handle.
ICMImageDescriptionGetPropertyInfo	Returns information about a particular property of a image description.
ICMImageDescriptionSetProperty	Sets a particular property of a image description handle.
ICMMultiPassStorageCopyDataAtTimeStamp	Called by a multipass-capable compressor to retrieve data at a given time stamp.
ICMMultiPassStorageCreateWithCallbacks	Assembles a multipass storage mechanism from callbacks.
ICMMultiPassStorageCreateWithTemporaryFile	Creates multipass storage using a temporary file.
ICMMultiPassStorageGetTimeStamp	Called by a multipass-capable compressor to retrieve a time stamp for which a value is stored.
ICMMultiPassStorageGetTypeID	Returns the type ID for the current multipass storage object.
ICMMultiPassStorageRelease	Decrements the retain count of a multipass storage object.
ICMMultiPassStorageRetain	Increments the retain count of a multipass storage object.
ICMMultiPassStorageSetDataAtTimeStamp	Called by a multipass-capable compressor to store data at a given time stamp.
QTOpenGLTextureContextCreate	Creates a new OpenGL texture context for a specified OpenGL context and pixel format.
QTPixelBufferContextCreate	Creates a new pixel buffer context with the given attributes.
QTVisualContextCopyImageForTime	Retrieves an image buffer from the visual context, indexed by the provided time.
QTVisualContextGetAttribute	Returns a visual context attribute.
QTVisualContextGetTypeID	Returns the CFTypeID for QTVisualContextRef.

QTVisualContextIsNewImageAvailable	Queries whether a new image is available for a given time.
QTVisualContextRelease	Releases a visual context object.
QTVisualContextRetain	Retains a visual context object.
QTVisualContextSetAttribute	Sets a visual context attribute.
QTVisualContextSetImageAvailableCallback	Installs a user-defined callback to receive notifications when a new image becomes available.
QTVisualContextTask	Causes visual context to release internally held resources for later re-use.
SetDSequenceNonScheduledDisplayDirection	Sets the display direction for a decompress sequence.
SetDSequenceNonScheduledDisplayTime	Sets the display time for a decompression sequence.

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.

kQTVisualContextExpectedReadAheadKey	A CFNumberRef that defines the number of seconds ahead of real time that the client expects to pull images out of a visual context.
kQTVisualContextOutputColorSpaceKey	
kQTVisualContextPixelBufferAttributesKey	A CFDictionaryRef that defines the dictionary containing pixel buffer attributes. See kICMCompressionSessionPropertyID_-PixelBufferPool in ICM Compression Session Properties.
kQTVisualContextTargetDimensions_HeightKey	A CFNumberRef that defines the height, in pixels, of the rendering target.
kQTVisualContextTargetDimensions_WidthKey	A CFNumberRef that defines the width, in pixels, of the rendering target.
kQTVisualContextTargetDimensionsKey	A CFDictionaryRef that defines the dictionary containing kQTVisualContextTargetDimensions_WidthKey and kQTVisualContextTargetDimensions_-HeightKey values (see below).

kQTVisualContextType_OpenGLTexture	The value of kQTVisualContextTypeKey for OpenGL texture visual contexts.
kQTVisualContextType_PixelBuffer	The value of kQTVisualContextTypeKey for pixel buffer visual contexts.
kQTVisualContextTypeKey	
kQTVisualContextWorkingColorSpaceKey	

Movies.h

Functions

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

AddMediaSample2	Adds sample data and a description to a media.
AddMediaSampleFromEncodedFrame	Adds sample data and description from an encoded frame to a media.
AddSampleTableToMedia	Adds a sample table to a media.
CopyMediaMutableSampleTable	Obtains information about sample references in a media in the form of a sample table.
DisposeQTTrackPropertyListenerUPP	Disposes a track property listener UPP.
ExtendMediaDecodeDurationToDisplayEndTime	Prepares a media for the addition of a completely new sequence of samples by ensuring that the media display end time is not later than the media decode end time.
GetMediaAdvanceDecodeTime	Returns the advance decode time of a media.
GetMediaDataSizeTime64	Determines the size, in bytes, of the sample data in a media segment.
GetMediaDecodeDuration	Returns the decode duration of a media.
GetMediaDisplayDuration	Returns the display duration of a media.
GetMediaDisplayEndTime	Returns the display end time of a media.
GetMediaDisplayStartTime	Returns the display start time of a media.
GetMediaNextInterestingDecodeTime	Searches for decode times of interest in a media.

GetMediaNextInterestingDisplayTime	Searches for display times of interest in a media.
GetMediaSample2	Retrieves sample data from a media file.
GetMovieAudioBalance	Returns the balance value for the audio mix of a movie currently playing.
GetMovieAudioContext	Returns the current audio context for a movie.
GetMovieAudioFrequencyLevels	Returns the current frequency meter levels of a movie mix.
GetMovieAudioFrequencyMeteringBandFrequencies	Returns the chosen middle frequency for each band in the configured frequency metering of a particular movie mix.
GetMovieAudioFrequencyMeteringNumBands	Returns the number of frequency bands being metered for a movie's specified audio mix.
GetMovieAudioGain	Returns the gain value for the audio mix of a movie currently playing.
GetMovieAudioMute	Returns the mute value for the audio mix of a movie currently playing.
GetMovieAudioVolumeLevels	Returns the current volume meter levels of a movie.
GetMovieAudioVolumeMeteringEnabled	Returns the enabled or disabled status of volume metering of a particular audio mix of a movie.
GetMovieVisualBrightness	Returns the brightness adjustment for the movie.
GetMovieVisualContext	Returns the current visual context for a movie.
GetMovieVisualContrast	Returns the contrast adjustment for the movie.
GetMovieVisualHue	Returns the hue adjustment for the movie.
GetMovieVisualSaturation	Returns the color saturation adjustment for the movie.
GetTrackAudioGain	Returns the gain value for the audio mix of a track currently playing.
GetTrackAudioMute	Returns the mute value for the audio mix of a track currently playing.

GetTrackAudioVolumeLevels	
GetTrackAudioVolumeMeteringEnabled	
GetTrackEditRate64	Returns the rate of the track edit of a specified track at an indicated time.
InvokeQTTrackPropertyListenerUPP	Invokes the specified property listener of a track.
MediaContainsDisplayOffsets	Tests whether a media contains display offsets.
MediaDecodeTimeToSampleNum	Finds the sample for a specified decode time.
MediaDisplayTimeToSampleNum	Finds the sample number for a specified display time.
MovieAudioExtractionBegin	Begins a movie audio extraction session.
MovieAudioExtractionEnd	Ends a movie audio extraction session.
MovieAudioExtractionFillBuffer	Extracts audio from a movie.
MovieAudioExtractionGetProperty	Gets a property of a movie audio extraction session.
MovieAudioExtractionGetPropertyInfo	Gets information about a property of a movie audio extraction session.
MovieAudioExtractionSetProperty	Sets a property of a movie audio extraction session.
NewMovieFromProperties	Creates a new movie using movie properties.
NewQTTrackPropertyListenerUPP	Creates a new callback to monitor a track property.
QTAddTrackPropertyListener	Installs a callback to monitor a track property.
QTAudioContextCreateForAudioDevice	Creates a QTAudioContext object that encapsulates a connection to a CoreAudio output device.
QTAudioContextRelease	
QTAudioContextRetain	
QTCopyMediaMetaData	Retains a media's metadata object and returns it.
QTCopyMovieMetaData	Retains a movie's metadata object and returns it.

QTCopyTrackMetaData	Retains a track's metadata object and returns it.
QTGetTrackProperty	Returns the value of a specific track property.
QTGetTrackPropertyInfo	Returns information about the properties of a track.
QTMetaDataAddItem	Adds an inline metadata item to the metadata storage format.
QTMetaDataGetItemCountWithKey	Returns the number of items in a metadata storage format with a certain key.
QTMetaDataGetItemProperty	Returns a property of a metadata item.
QTMetaDataGetItemPropertyInfo	Returns information about a property of a metadata item.
QTMetaDataGetItemValue	Returns the value of a metadata item from an item identifier.
QTMetaDataGetNextItem	Returns the next metadata item corresponding to a specified key.
QTMetaDataGetProperty	Returns a property of a metadata object.
QTMetaDataGetPropertyInfo	Returns information about a property of a metadata object.
QTMetaDataRelease	Decrements the retain count of a metadata object.
QTMetaDataRemoveItem	Removes a metadata item from a storage format.
QTMetaDataRemoveItemsWithKey	Removes metadata items with a specific key from the storage format.
QTMetaDataRetain	Increments the retain count of a metadata object.
QTMetaDataSetItem	Sets the value of the metadata item from the item identifier.
QTMetaDataSetItemProperty	Sets a property of a metadata item.
QTMetaDataSetProperty	Sets a property of a metadata object.
QTRemoveTrackPropertyListener	Removes a track property monitoring callback
QTSampleTableAddSampleDescription	Adds a sample description to a sample table, returning a sample description ID that can be used to refer to it.

QTSampleTableAddSampleReferences	Adds sample references to a sample table.
QTSampleTableCopySampleDescription	Retrieves a sample description from a sample table.
QTSampleTableCreateMutable	Creates a new, empty sample table.
QTSampleTableCreateMutableCopy	Copies a sample table.
QTSampleTableGetDataOffset	Returns the data offset of a sample.
QTSampleTableGetDataSizePerSample	Returns the data size of a sample.
QTSampleTableGetDecodeDuration	Returns the decode duration of a sample.
QTSampleTableGetDisplayOffset	Returns the offset from decode time to display time of a sample.
QTSampleTableGetNextAttributeChange	Finds the next sample number at which one or more of a set of given sample attributes change.
QTSampleTableGetNumberOfSamples	Returns the number of samples in a sample table.
QTSampleTableGetProperty	Returns the value of a specific sample table property.
QTSampleTableGetPropertyInfo	Returns information about the properties of a sample table.
QTSampleTableGetSampleDescriptionID	Returns the sample description ID of a sample.
QTSampleTableGetSampleFlags	Returns the media sample flags of a sample.
QTSampleTableGetTimeScale	Returns the timescale of a sample table.
QTSampleTableGetTypeID	Returns the CFTypeID value for the current sample table.
QTSampleTableRelease	Decrements the retain count of a sample table.
QTSampleTableReplaceRange	Replaces a range of samples in a sample table with a range of samples from another sample table.
QTSampleTableRetain	Increments the retain count of a sample table.
QTSampleTableSetProperty	Sets the value of a specific sample table property.
QTSampleTableSetTimeScale	Changes the timescale of a sample table.

QTSetTrackProperty	Sets the value of a specific track property.
QTSoundDescriptionConvert	Converts a sound description from one version to another.
QTSoundDescriptionCreate	Creates a sound description structure of the requested kind from an AudioStreamBasicDescription, optional audio channel layout, and optional magic cookie.
QTSoundDescriptionGetProperty	Gets a particular property of a sound description.
QTSoundDescriptionGetPropertyInfo	Gets information about a particular property of a sound description.
QTSoundDescriptionSetProperty	Sets a particular property of a sound description.
SampleNumToMediaDecodeTime	Finds the decode time for a specified sample.
SampleNumToMediaDisplayTime	Finds the display time for a specified sample.
SetMovieAudioBalance	Sets the balance level for the mixed audio output of a movie.
SetMovieAudioContext	Targets a movie to render into an audio context.
SetMovieAudioFrequencyMeteringNumBands	Configures frequency metering for a particular audio mix in a movie.
SetMovieAudioGain	Sets the audio gain level for the mixed audio output of a movie, altering the perceived volume of the movie's playback.
SetMovieAudioMute	Sets the mute value for the audio mix of a movie currently playing.
SetMovieAudioVolumeMeteringEnabled	Enables or disables volume metering of a particular audio mix of a movie.
SetMovieVisualBrightness	Sets the brightness adjustment for the movie.
SetMovieVisualContext	Targets a movie to render into a visual context.
SetMovieVisualContrast	Sets the contrast adjustment for the movie.
SetMovieVisualHue	Sets the hue adjustment for the movie.
SetMovieVisualSaturation	Sets the color saturation adjustment for the movie.

`SetTrackAudioGain`	Sets the audio gain level for the audio output of a track, altering the perceived volume of the track's playback.
`SetTrackAudioMute`	Mutes or unmutes the audio output of a track.
`SetTrackAudioVolumeMeteringEnabled`	
`TrackTimeToMediaDisplayTime`	Converts a track's time value to a display time value that is appropriate to the track's media, using the track's edit list.

QuickTimeComponents.h

Functions

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

`DataHGetAvailableFileSize64`	
`DataHGetDataAvailability64`	
`SCAudioInvokeLegacyCodecOptionsDialog`	Invokes the legacy code options dialog of an audio codec component.
`SCCopyCompressionSessionOptions`	Creates a compression session options object based upon the settings in the Standard Compression component.

10.3 Symbol Changes

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

C Symbols

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

HIMovieView.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.

`kEventClassMovieView`	
`kEventMovieViewOptimalBoundsChanged`	
`kHIMovieViewAcceptsFocusAttribute`	
`kHIMovieViewAutoIdlingAttribute`	
`kHIMovieViewClassID`	Class ID for HIMovieView.
`kHIMovieViewControllerVisibleAttribute`	
`kHIMovieViewEditableAttribute`	
`kHIMovieViewHandleEditingHIAttribute`	
`kHIMovieViewNoAttributes`	
`kHIMovieViewStandardAttributes`	

ImageCodec.h

Functions

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

`ImageCodecLockBits`	
`ImageCodecUnlockBits`	

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.

`codecSupportsOutOfOrderDisplayTimes`	Codec supports frames queued in one order for display in a different order, for example IPB content.
`codecSupportsScheduledBackwards-PlaybackWithDifferenceFrames`	Codec can use additional buffers to minimize redecoding during backwards playback.
`ICMFrameTimeRecord_QT3`	
`kDVCompressorPropertyID_AspectRatio16x9`	If set, indicates that the compressor should use a 16:9 picture aspect ratio. If clear, the compressor will use the default 4:3 picture aspect ratio.
`kDVCompressorPropertyID_ProgressiveScan`	If set, indicates that the compressed frames should be marked as progressive-scan. By default, this flag is clear, meaning that frames should be marked as interlaced.
`kICMDeinterlaceFields`	
`kICMImageCompressorPropertyID_EnforcedCleanAperture`	The clean aperture enforced for compressed frames.
`kICMImageCompressorPropertyID_EnforcedEncodedHeight`	The encoded height enforced for compressed frames.
`kICMImageCompressorPropertyID_EnforcedEncodedWidth`	The encoded width enforced for compressed frames.
`kICMImageCompressorPropertyID_EnforcedFieldInfo`	The number and order of fields enforced for compressed frames.

kICMImageCompressorPropertyID_- EnforcedPixelAspectRatio	The pixel aspect ratio enforced for compressed frames.
kICMNoDeinterlacing	
kImageCodecBeginPassSelect	
kImageCodecCompleteFrameSelect	
kImageCodecDecodeBandSelect	
kImageCodecEncodeFrameSelect	
kImageCodecEndPassSelect	
kImageCodecGetPreferredChunkSizeAndAlignmentSelect	
kImageCodecLockBitsSelect	
kImageCodecPrepareToCompressFramesSelect	
kImageCodecProcessBetweenPassesSelect	
kImageCodecUnlockBitsSelect	
kParameterProperty	
kQTPropertyClass_DVCompressor	The property class for DV compressors. (Applicable to DV25, DV50, NTSC, PAL, PROPAL.)
kQTPropertyClass_ImageCompressor	Property class for image compressor components.
kQTRowBytesImageDescriptionExtension	Big-endian SInt32.
ParameterProperty	

ImageCompression.h

Functions

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

DisposeQTComponentPropertyListenerFilterUPP	
DisposeQTComponentPropertyListenerUPP	
GraphicsExportGetFlags	

GraphicsExportGetInputCGBitmapContext	Retrieves the CGBitmapContext that the graphics exporter is using as its input image.
GraphicsExportGetInputCGImage	Determines which Core Graphics CGImage is the source for a graphics export operation.
GraphicsExportSetFlags	
GraphicsExportSetInputCGBitmapContext	Sets the CGBitmapContext that the graphics exporter will use as its input image.
GraphicsExportSetInputCGImage	Specifies a Core Graphics CGImage as the source for a graphics export operation.
GraphicsImportCreateCGImage	Imports an image as a Core Graphics CGImage.
GraphicsImportDoExportImageFileToDataRefDialog	Presents a dialog box that lets the user save an imported image in a foreign file format.
GraphicsImportExportImageFileToDataRef	Saves an imported image in a foreign file format.
GraphicsImportGetDestinationColorSyncProfileRef	Retrieves a ColorSync profile from a graphics importer component.
GraphicsImportGetGenericColorSyncProfile	Retrieves the generic colorsync profile for a graphics importer component.
GraphicsImportGetOverrideSourceColorSyncProfileRef	Retrieves the override ColorSync profile for a graphics importer component.
GraphicsImportGetReturnGenericColorSyncProfile	
GraphicsImportSaveAsPictureToDataRef	Creates a storage location that contains a QuickDraw picture for an imported image.
GraphicsImportSaveAsQuickTimeImageFileToDataRef	Creates a storage location that contains a QuickTime image of an imported image.
GraphicsImportSetDestinationColorSyncProfileRef	Sets the ColorSync profile for a graphics importer component.
GraphicsImportSetOverrideSourceColorSyncProfileRef	Sets the override ColorSync profile for a graphics importer component.
GraphicsImportSetReturnGenericColorSyncProfile	

GraphicsImportWillUseColorMatching	Asks whether GraphicsImportDraw will use color matching if called with the current importer settings.
InvokeQTComponentPropertyListenerFilterUPP	
InvokeQTComponentPropertyListenerUPP	
NewQTComponentPropertyListenerFilterUPP	
NewQTComponentPropertyListenerUPP	
QTAddComponentPropertyListener	Installs a callback to monitor a component property.
QTComponentPropertyListenerCollectionAddListener	Adds a listener callback for a specified property class and ID to a property listener collection.
QTComponentPropertyListenerCollectionCreate	Creates a collection of component property monitors.
QTComponentPropertyListenerCollection-HasListenersForProperty	Determines if there are any listeners in a component property listener collection registered for a specified property class and ID.
QTComponentPropertyListenerCollectionIsEmpty	Determines if a listener collection is empty.
QTComponentPropertyListenerCollectionNotifyListeners	Calls all listener callbacks in a component property listener collection registered for a specified property class and ID.
QTComponentPropertyListenerCollectionRemoveListener	Removes a listener callback with a specified property class and ID from a property listener collection.
QTGetComponentProperty	Returns the value of a specific component property.
QTGetComponentPropertyInfo	Returns information about the properties of a component.
QTRemoveComponentPropertyListener	Removes a component property monitoring callback.
QTSetComponentProperty	Sets the value of a specific component property.

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.

AVAILABLE_MAC_OS_X_VERSION_10_5_AND_LATER	
codecCompletionDecoded	Codec has decoded this frame; if it is cancelled and rescheduled, set icmFrameAlreadyDecoded in ICMFrameTimeRecord.flags.
codecCompletionNotDisplayable	The frame may still be scheduled for decode, but will not be able to be displayed because the buffer containing it will need to be recycled to display earlier frames..
codecCompletionNotDrawn	Set in conjunction with codecCompletionDest to indicate that the frame was not drawn.
codecDSequenceDeinterlaceFields	
codecInfoDoesMultiPass	Codec can perform multi-pass compression.
codecInfoDoesReorder	Codec can rearrange frames during compression.
ComponentPropertyClass	
ComponentPropertyID	
ComponentPropertyInfo	
ComponentValuePtr	
ComponentValueType	
ConstComponentValuePtr	
graphicsImporterCanValidateFile	
graphicsImporterHasMIMEList	
graphicsImporterIsBaseImporter	
graphicsImporterSubTypeIsFileExtension	
graphicsModePerComponentAlpha	
ICMCompressionFrameOptionsRef	
ICMCompressionPassModeFlags	
ICMCompressionSessionOptionsRef	

ICMCompressionSessionRef	
ICMCompressorSessionRef	
ICMCompressorSourceFrameRef	
ICMDataRateLimit	
ICMDecompressionFrameOptionsRef	
ICMDecompressionSessionOptionsRef	
ICMDecompressionSessionRef	
ICMDecompressionTrackingCallback	
ICMDecompressionTrackingCallbackRecord	
ICMDecompressionTrackingFlags	
ICMEncodedFrameOutputCallback	
ICMEncodedFrameOutputRecord	
ICMEncodedFrameRef	
ICMFieldMode	
icmFrameAlreadyDecoded	
icmFrameTimeDecodeImmediately	
icmFrameTimeDoNotDisplay	
icmFrameTimeHasDecodeTime	Indicates that decodeTime is valid.
icmFrameTimeIsNonScheduledDisplayTime	
ICMFrameType	
ICMMultiPassCopyDataAtTimeStampCallback	
ICMMultiPassGetTimeStampCallback	
ICMMultiPassReleaseCallback	
ICMMultiPassSetDataAtTimeStampCallback	
ICMMultiPassStorageCallbacks	
ICMMultiPassStorageCreationFlags	
ICMMultiPassStorageRef	
ICMMultiPassStorageStep	

ICMMutableEncodedFrameRef	
ICMNonScheduledDisplayTime	
ICMSimpleBoundaryConditions	
ICMSourceTrackingCallback	
ICMSourceTrackingCallbackRecord	
ICMSourceTrackingFlags	
ICMValidTimeFlags	
kComponentPropertyCacheFlagIsDynamic	Property metadata should not be cached at all.
kComponentPropertyCacheFlagNotPersistent	Property metadata should not be saved in persistent cache.
kComponentPropertyCacheFlags	One of the following two flags: Also 'flgs'.
kComponentPropertyCacheSeed	A component property cache seed value. Also 'seed'.
kComponentPropertyClassPropertyInfo	A QTComponentPropertyInfo structure that defines a property information class. Also 'pnfo'.
kComponentPropertyExtendedInfo	
kComponentPropertyFlagCanGetLater	
kComponentPropertyFlagCanGetNow	
kComponentPropertyFlagCanSetLater	
kComponentPropertyFlagCanSetNow	
kComponentPropertyFlagGetBufferMustBeInitialized	
kComponentPropertyFlagHasExtendedInfo	
kComponentPropertyFlagValueIsCFTypeRef	
kComponentPropertyFlagValueMustBeReleased	
kComponentPropertyFlagWillNotifyListeners	
kComponentPropertyInfoList	An array of QTComponentPropertyInfo structures, one for each property. Also 'list'.
kDVCPro100NTSCCodecType	
kDVCPro100PALCodecType	

kDVCPro50NTSCCodecType	
kDVCPro50PALCodecType	
kDVCPROHD1080i50CodecType	
kDVCPROHD1080i60CodecType	
kDVCPROHD720pCodecType	
kGraphicsExportGetFlagsSelect	
kGraphicsExportGetInputCGBitmapContextSelect	
kGraphicsExportGetInputCGImageSelect	
kGraphicsExportSetFlagsSelect	
kGraphicsExportSetInputCGBitmapContextSelect	
kGraphicsExportSetInputCGImageSelect	
kGraphicsImportCreateCGImageSelect	
kGraphicsImportCreateCGImageUsingCurrentSettings	Use the current settings.
kGraphicsImportDoExportImage-FileToDataRefDialogSelect	
kGraphicsImporterDontUseColorMatching	Set this flag (before calling GraphicsImportGetColorSyncProfile) if you do matching yourself.
kGraphicsImportExportImageFileToDataRefSelect	
kGraphicsImportGetDestination-ColorSyncProfileRefSelect	
kGraphicsImportGetGenericColorSyncProfileSelect	
kGraphicsImportGetOverrideSource-ColorSyncProfileRefSelect	
kGraphicsImportGetReturnGeneric-ColorSyncProfileSelect	
kGraphicsImportSaveAsPictureToDataRefSelect	
kGraphicsImportSaveAsQuickTime-ImageFileToDataRefSelect	
kGraphicsImportSetDestination-ColorSyncProfileRefSelect	
kGraphicsImportSetOverrideSource-ColorSyncProfileRefSelect	

kGraphicsImportSetReturnGeneric-ColorSyncProfileSelect	
kGraphicsImportWillUseColorMatchingSelect	
kH264CodecType	
kICMCompressionFrameOptionsPropertyID_ForceKeyFrame	Boolean, ReadWrite.
kICMCompressionFrameOptionsPropertyID_FrameType	ICMFrameType, ReadWrite.
kICMCompressionPassMode_NoSourceFrames	
kICMCompressionPassMode_-NotReadyToOutputEncodedFrames	
kICMCompressionPassMode_OutputEncodedFrames	
kICMCompressionPassMode_ReadFromMultiPassStorage	
kICMCompressionPassMode_WriteToMultiPassStorage	
kICMCompressionSessionOptionsPropertyID_-AllowAsyncCompletion	Enables the compressor to call the encoded-frame callback from a different thread. By default this option is FALSE, which means that the compressor must call the encoded-frame callback from the same thread as ICMCompressionSessionEncodeFrame and ICMCompressionSessionCompleteFrames. Also 'asok'.
kICMCompressionSessionOptionsPropertyID_-AllowFrameReordering	Enables frame reordering. To encode B-frames a compressor must reorder frames, which may mean that the order in which they are emitted and stored (the decode order) may be different from the order in which they are presented to the compressor (the display order). By default, frame reordering is disabled. To encode using B-frames, you must enable frame reordering by passing TRUE in this property. Also 'b ok'.
kICMCompressionSessionOptionsPropertyID_-AllowFrameTimeChanges	Enables the compressor to modify frame times, improving its performance. Some compressors are able to identify and coalesce runs of identical frames and emit single frames with longer duration, or emit frames at a different frame rate from the original. By default, this flag is set to FALSE, which forces the compressor to emit one encoded frame for every source frame and to preserve frame display times. This option replaces the practice of having compressors return special high similarity values to indicate that frames can be dropped. Also '+ ok'.

kICMCompressionSessionOptionsPropertyID_-AllowTemporalCompression	Enables temporal compression of P-frames and B-frames. By default, temporal compression is disabled. Also 'p ok'.
kICMCompressionSessionOptionsPropertyID_-AverageDataRate	The long-term desired average data rate in bytes per second. This is not an absolute limit. The default data rate is zero, indicating that the setting of kICMCompressionSessionOptionsPropertyID_-Quality should determine the size of compressed data. Data rate settings have effect only when timing information is provided for source frames. Some codecs do not accept limiting to specified data rates. Also 'aver'.
kICMCompressionSessionOptionsPropertyID_-CleanAperture	Native-endian CleanApertureImageDescriptionExtension, ReadWrite.
kICMCompressionSessionOptionsPropertyID_ColorTable	The color table for compression, used with indexed-color depths. Clients who are passed this property are responsible for disposing the returned CTabHandle. Also 'clut'.
kICMCompressionSessionOptionsPropertyID_-CompressorComponent	Sets a specific compressor component or component instance to be used, or passes one of the wildcards anyCodec, bestSpeedCodec, bestFidelityCodec, or bestCompressionCodec. Pass this option to force the Image Compression Manager to use a specific compressor component or compressor component instance. To allow the Image Compression Manager to choose the compressor component, set the compressorComponent to anyCodec (the default), bestSpeedCodec, bestFidelityCodec, or bestCompressionCodec. If you pass in a component instance that you opened, the ICM will not close that instance; you must do so after the compression session is released. Also 'imco'.
kICMCompressionSessionOptionsPropertyID_-CompressorSettings	A handle containing compressor settings. The compressor will be configured with these settings (by a call to ImageCodecSetSettings) during the ICMCompressionSessionCreate process. Also 'cost'.
kICMCompressionSessionOptionsPropertyID_-CPUTimeBudget	UInt32, ReadWrite.
kICMCompressionSessionOptionsPropertyID_-DataRateLimitCount	UInt32, Read.

kICMCompressionSessionOptionsPropertyID_-DataRateLimits	Zero, one, or two hard limits on data rate. Each hard limit is described by a data size in bytes and a duration in seconds. It requires that the total size of compressed data for any contiguous segment of that duration (in decode time) must not exceed the data size. By default, no data rate limits are set. When setting this property, the inPropValueSize parameter should be the number of data rate limits multiplied by sizeof(ICMDataRateLimit). Data rate settings have an effect only when timing information is provided for source frames. Some codecs do not accept limiting to specified data rates. Also 'hard'.
kICMCompressionSessionOptionsPropertyID_Depth	UInt32, ReadWrite.
kICMCompressionSessionOptionsPropertyID_-DurationsNeeded	Indicates that durations of emitted frames are needed. If this option is set and source frames are provided with times but not durations, then frames will be delayed so that durations can be calculated as the difference between one frame's time stamp and the next frame's time stamp. By default, this flag is FALSE, so frames will not be delayed in order to calculate durations. If you pass encoded frames to AddMediaSampleFromEncodedFrame, you must set this flag to TRUE. Also 'need'.
kICMCompressionSessionOptionsPropertyID_-ExpectedFrameRate	Fixed, ReadWrite.
kICMCompressionSessionOptionsPropertyID_-ExtraAspectRatioStretchFactor	Requests additional distortion to be applied to the aspect ratio in the kICMScalingMode_Letterbox and kICMScalingMode_Trim scaling modes. Values greater than fixed1 mean wider, values less than fixed1 mean narrower. For example, a value of X2Fix(2.0) would make the picture aspect ratio twice as wide.
kICMCompressionSessionOptionsPropertyID_FieldInfo	FieldInfoImageDescriptionExtension2, ReadWrite.
kICMCompressionSessionOptionsPropertyID_-MaxDataRateLimits	UInt32, Read.

kICMCompressionSessionOptionsPropertyID_-MaxFrameDelayCount	The maximum frame delay count is the maximum number of frames that a compressor is allowed to hold before it must output a compressed frame. This value limits the number of frames that may be held in the compression window. If the maximum frame delay count is M, then before the call to encode frame N returns, frame N-M must have been emitted. The default value is kICMUnlimitedFrameDelayCount, which sets no limit on the compression window. Also 'cwin'.
kICMCompressionSessionOptionsPropertyID_-MaxFrameDelayTime	TimeValue64, ReadWrite.
kICMCompressionSessionOptionsPropertyID_-MaxKeyFrameInterval	The maximum interval between key frames, also known as the key frame rate. Compressors are allowed to generate key frames more frequently if this would result in more efficient compression. The default key frame interval is 0, which indicates that the compressor should choose where to place all key frames. This differs from previous practice, in which a key frame rate of zero disabled temporal compression. Also 'kyfr'.
kICMCompressionSessionOptionsPropertyID_-MaxPartialSyncFrameInterval	SInt32, ReadWrite.
kICMCompressionSessionOptionsPropertyID_-MultiPassStorage	A multipass compression client must provide a storage location for multipass data. Pass ICMMultiPassStorageCreateWithTemporaryFile to make the ICM store multipass data in a temporary file. Pass ICMMultiPassStorageCreateWithCallbacks to manage the storage yourself. Note that the amount of multipass data to be stored can be substantial; it could be greater than the size of the output movie file. If this property is not NULL, the client must call ICMCompressionSessionBeginPass and ICMCompressionSessionEndPass around groups of calls to ICMCompressionSessionEncodeFrame. By default, this property is NULL and multipass compression is not enabled. The compression session options object retains the multipass storage object when one is set. Also 'imps'.
kICMCompressionSessionOptionsPropertyID_-PixelAspectRatio	Native-endian PixelAspectRatioImageDescriptionExtension, ReadWrite.

kICMCompressionSessionOptionsPropertyID_Quality	The compression quality. This value is always used to set the spatial quality; if temporal compression is enabled, it is also used to set temporal quality. The default quality is codecNormalQuality. Also 'qual'.
kICMCompressionSessionOptionsPropertyID_ScalingMode	OSType, ReadWrite.
kICMCompressionSessionOptionsPropertyID_-SimpleBoundaryConditions	
kICMCompressionSessionOptionsPropertyID_-SourceFrameCount	UInt64, ReadWrite.
kICMCompressionSessionOptionsPropertyID_-WasCompressed	Indicates that the source was previously compressed. This property is an optional information hint to the compressor; by default it is FALSE. Also 'wasc'.
kICMCompressionSessionPropertyID_-CompressorPixelBufferAttributes	The compressor's pixel buffer attributes for the compression session. You can use these to create a pixel buffer pool for source pixel buffers. This is not the same as the sourcePixelBufferAttributes property passed to ICMCompressionSessionCreate. Getting this property does not change its retain count. Also 'batt'.
kICMCompressionSessionPropertyID_ImageDescription	The image description for a compression session. For some codecs, the image description may not be available before the first frame is compressed. Multiple calls to retrieve this property will return the same handle. The ICM will dispose of this handle when the compression session is disposed; the caller must not dispose of it. Also 'idsc'.
kICMCompressionSessionPropertyID_PixelBufferPool	A pool that can provide ideal source pixel buffers for a compression session. The compression session creates this pixel buffer pool based on the compressor's pixel buffer attributes and any pixel buffer attributes passed in to ICMCompressionSessionCreate. If the source pixel buffer attributes and the compressor pixel buffer attributes can not be reconciled, the pool is based on the source pixel buffer attributes and the ICM converts each CVPixelBuffer internally. Also 'pool'.
kICMCompressionSessionPropertyID_TimeScale	The time scale for the compression session. Also 'tscl'.
kICMDecompressionFrameOptionsPropertyID_-DestinationPixelBuffer	CVPixelBufferRef, ReadWrite.

kICMDecompressionSessionOptionsPropertyID_Accuracy	CodecQ, ReadWrite.
kICMDecompressionSessionOptionsPropertyID_-ApertureMode	You can set the aperture mode property on a decompression session options object to indicate whether pixel buffers should be tagged to enable aspect ratio and clean aperture correction. The default aperture mode for a decompression session is clean aperture mode.
kICMDecompressionSessionOptionsPropertyID_-DecompressorComponent	DecompressorComponent, ReadWrite.
kICMDecompressionSessionOptionsPropertyID_-DisplayOrderRequired	Boolean, ReadWrite.
kICMDecompressionSessionOptionsPropertyID_FieldMode	ICMFieldMode, ReadWrite.
kICMDecompressionSessionOptionsPropertyID_-MaxBufferCount	UInt32, ReadWrite.
kICMDecompressionSessionOptionsPropertyID_-OutputAheadTime	TimeRecord, ReadWrite.
kICMDecompressionSessionPropertyID_-NonScheduledDisplayDirection	Fixed, ReadWrite.
kICMDecompressionSessionPropertyID_-NonScheduledDisplayTime	ICMNonScheduledDisplayTime, ReadWrite.
kICMDecompressionSessionPropertyID_PixelBufferPool	CVPixelBufferPoolRef, Read.
kICMDecompressionSessionPropertyID_-PixelBufferPoolIsShared	Boolean, Read.
kICMDecompressionTracking_EmittingFrame	
kICMDecompressionTracking_FrameDecoded	
kICMDecompressionTracking_FrameDropped	
kICMDecompressionTracking_FrameNeedsRequeueing	
kICMDecompressionTracking_LastCall	
kICMDecompressionTracking_ReleaseSourceData	
kICMFieldMode_BothFields	
kICMFieldMode_BottomFieldOnly	
kICMFieldMode_DeinterlaceFields	
kICMFieldMode_TopFieldOnly	

kICMFrameType_B	
kICMFrameType_I	
kICMFrameType_P	
kICMFrameType_Unknown	
kICMImageDescriptionPropertyID_CGColorSpace	A CGColorSpaceRef for the colorspace described by the image description, constructed from video color information or ICC Profile. It is important to note that the YCbCr matrix from the video color info is not represented in the CGColorSpaceRef. The caller of GetProperty is responsible for releasing this, for example, by calling CGColorSpaceRelease. Only supported on Mac OS X.
kICMImageDescriptionPropertyID_ClassicTrackHeight	Fixed, Read.
kICMImageDescriptionPropertyID_ClassicTrackWidth	Fixed, Read.
kICMImageDescriptionPropertyID_CleanAperture	Native-endian CleanApertureImageDescriptionExtension, ReadWrite.
kICMImageDescriptionPropertyID_CleanApertureClipRect	The clean aperture as a FixedRect in source coordinates, within the rectangle defined by the image description width and height, suitable for use as a source rectangle in a decompression sequence. For historical reasons, the DVCPROHD codecs store the production aperture display dimensions in the image description width and height; the actual encoded dimensions are smaller. For DVCPROHD, the clip rect will be relative to the image description width and height, not the encoded dimensions.
kICMImageDescriptionPropertyID_-CleanApertureDisplayDimensions	The dimensions at which the image can be displayed on a square-pixel display, generally calculated using the clean aperture and pixel aspect ratio. Note that this value is returned as a FixedPoint; the width and height can also be read separately as rounded SInt32s via kICMImageDescriptionPropertyID_-CleanApertureDisplayWidth and kICMImageDescriptionPropertyID_-CleanApertureDisplayHeight.
kICMImageDescriptionPropertyID_-CleanApertureDisplayHeight	A height at which the buffer's image could be displayed on a square-pixel display, possibly calculated using the clean aperture and pixel aspect ratio.

kICMImageDescriptionPropertyID_-CleanApertureDisplayWidth	A width at which the buffer's image could be displayed on a square-pixel display, possibly calculated using the clean aperture and pixel aspect ratio.
kICMImageDescriptionPropertyID_CleanApertureMatrix	A matrix transforming the clean aperture clip rect to the origin, scaled to the clean aperture display dimensions. For historical reasons, the DVCPROHD codecs store the production aperture display dimensions in the image description width and height; the actual encoded dimensions are smaller. For DVCPROHD, the matrix will be relative to the image description width and height, not the encoded dimensions.
kICMImageDescriptionPropertyID_DisplayHeight	SInt32, Read.
kICMImageDescriptionPropertyID_DisplayWidth	SInt32, Read.
kICMImageDescriptionPropertyID_EncodedHeight	SInt32, ReadWrite.
kICMImageDescriptionPropertyID_-EncodedPixelsDimensions	Describes the dimensions of the encoded image. Note that this value is returned as a FixedPoint for convenience; the width and height can also be read separately as SInt32s via kICMImageDescriptionPropertyID_-EncodedWidth and kICMImageDescriptionPropertyID_-EncodedHeight.
kICMImageDescriptionPropertyID_EncodedWidth	SInt32, ReadWrite.
kICMImageDescriptionPropertyID_FieldInfo	FieldInfoImageDescriptionExtension2, ReadWrite.
kICMImageDescriptionPropertyID_GammaLevel	Fixed, ReadWrite.
kICMImageDescriptionPropertyID_ICCProfile	A CFDataRef containing the serialized ICC profile described by the image description. The caller of GetProperty is responsible for releasing this, for example, by calling CFRelease.
kICMImageDescriptionPropertyID_NCLCColorInfo	Native-endian NCLCColorInfoImageDescriptionExtension, ReadWrite.
kICMImageDescriptionPropertyID_PixelAspectRatio	Native-endian PixelAspectRatioImageDescriptionExtension, ReadWrite.

kICMImageDescriptionPropertyID_-ProductionApertureDisplayDimensions	The dimensions at which the image could be displayed on a square-pixel display, disregarding any clean aperture but honoring the pixel aspect ratio. This may be useful for authoring applications that want to expose the edge-processing region. For general viewing, use kICMImageDescriptionPropertyID_-CleanApertureDimensions instead. Note that this value is returned as a FixedPoint; the width and height can also be read separately as rounded SInt32s via kICMImageDescriptionPropertyID_-ProductionApertureDisplayWidth and kICMImageDescriptionPropertyID_-ProductionApertureDisplayHeight.
kICMImageDescriptionPropertyID_-ProductionApertureDisplayHeight	A height at which the image could be displayed on a square-pixel display, disregarding any clean aperture but honoring the pixel aspect ratio. This may be useful for authoring applications that want to expose the edge processing region. For general viewing, use kICMImageDescriptionPropertyID_-DisplayHeight instead.
kICMImageDescriptionPropertyID_-ProductionApertureDisplayWidth	A width at which the image could be displayed on a square-pixel display, disregarding any clean aperture but honoring the pixel aspect ratio. This may be useful for authoring applications that want to expose the edge processing region. For general viewing, use kICMImageDescriptionPropertyID_-DisplayWidth instead.
kICMImageDescriptionPropertyID_-ProductionApertureMatrix	A matrix transforming the image to the origin, scaled to the production aperture display dimensions. For historical reasons, the DVCPROHD codecs store the production aperture display dimensions in the image description width and height; the actual encoded dimensions are smaller. For DVCPROHD, the matrix will be relative to the image description width and height, not the encoded dimensions.
kICMImageDescriptionPropertyID_-ProductionDisplayHeight	SInt32, Read.
kICMImageDescriptionPropertyID_-ProductionDisplayWidth	SInt32, Read.

kICMImageDescriptionPropertyID_RowBytes	SInt32, ReadWrite.
kICMImageDescriptionPropertyID_StepDuration	Defines a duration for quantizing time. This is applicable for cases where a single media sample generates visual output that varies continuously through its duration. By interpreting this property, such a sample may be considered to have internal "step points" at multiples of the stepping duration. This can be used to throttle frame generation during playback, and when stepping using InterestingTime APIs. Setting a step duration with value zero removes any current step duration.
kICMImageDescriptionPropertyID_SummaryString	A localized, human readable string summarizing the image as a CFString, that is, "Apple DV, 720 x 480 (640 x 480), Millions". The elements are: the codec name, the encoded pixels dimensions, then parenthetically the clean aperture mode dimensions, but only if they are different from the encoded pixels dimensions; then the depth. The codec name shall be from the localized decompressor component name string if exactly one decompressor with the correct cType is available; otherwise the string in the image description shall be used. The caller of GetProperty is responsible for releasing this CFString, for example, by calling CFRelease.
kICMMultiPassStorage_DoNotDeleteWhenDone	
kICMMultiPassStorage_GetFirstTimeStamp	
kICMMultiPassStorage_GetLastTimeStamp	
kICMMultiPassStorage_GetNextTimeStamp	
kICMMultiPassStorage_GetPreviousTimeStamp	
kICMMultiPassStorageCallbacksVersionOne	
kICMScalingMode_Letterbox	The clean aperture of the source frames is scaled to fit inside the clean aperture of the destination, preserving the original display aspect ratio. If the display aspect ratios are different, the source frames are centered with black bars above and below, or to the left and right.
kICMScalingMode_StretchCleanAperture	The clean aperture of the source frames is scaled to the clean aperture of the destination.

kICMScalingMode_StretchProductionAperture	The full width and height of source frames is scaled to the full width and height of the destination. This is the default if no other scaling mode is specified.
kICMScalingMode_Trim	The clean aperture of the source frames is scaled to cover the clean aperture of the destination, preserving the original display aspect ratio. If the display aspect ratios are different, the source frames are centered and cropped.
kICMSourceTracking_CopiedPixelBuffer	
kICMSourceTracking_FrameTimeWasChanged	
kICMSourceTracking_FrameWasDropped	
kICMSourceTracking_FrameWasEncoded	
kICMSourceTracking_FrameWasMerged	
kICMSourceTracking_LastCall	
kICMSourceTracking_ReleasedPixelBuffer	
kICMUnlimitedCPUTimeBudget	No CPU time limit on compression.
kICMUnlimitedFrameDelayCount	No limit on the number of frames in the compression window.
kICMUnlimitedFrameDelayTime	No time limit on the frames in the compression window.
kICMValidTime_DecodeDurationIsValid	
kICMValidTime_DecodeTimeStampIsValid	
kICMValidTime_DisplayDurationIsValid	The time value passed in displayDuration is valid.
kICMValidTime_DisplayOffsetIsValid	
kICMValidTime_DisplayTimeStampIsValid	The time value passed in displayTimeStamp is valid.
kJPEG2000CodecType	
kPixletCodecType	
kQTAddComponentPropertyListenerSelect	

kQTApertureMode_Classic	An aperture mode which gives compatibility with behavior in QuickTime 7.0.x and earlier. A movie in classic aperture mode uses track dimensions as set in NewMovieTrack and SetTrackDimensions. A decompression session in classic aperture mode does not set the clean aperture or pixel aspect ratio attachments on emitted pixel buffers. Movies default to classic aperture mode. If you call SetTrackDimensions on a track, the movie is automatically switched into classic aperture mode.
kQTApertureMode_CleanAperture	An aperture mode for general display. Where possible, video will be displayed at the correct pixel aspect ratio, trimmed to the clean aperture. A movie in clean aperture mode sets each track's dimensions to match its kQTVisualPropertyID_-CleanApertureDimensions. A decompression session in clean aperture mode sets the clean aperture and pixel aspect ratio attachments on emitted pixel buffers based on the image description.
kQTApertureMode_EncodedPixels	An aperture mode for technical use. Displays all encoded pixels with no aspect ratio or clean aperture compensation. A movie in encoded pixels aperture mode sets each track's dimensions to match its kQTVisualPropertyID_-EncodedPixelsDimensions. A decompression session in encoded pixels aperture mode does not set the clean aperture or pixel aspect ratio attachments on emitted pixel buffers.
kQTApertureMode_ProductionAperture	An aperture mode for modal use in authoring applications. Where possible, video will be displayed at the correct pixel aspect ratio, but without trimming to the clean aperture so that the edge processing region can be viewed. A movie in production aperture mode sets each track's dimensions to match its kQTVisualPropertyID_-ProductionApertureDimensions. A decompression session in production aperture mode sets the pixel aspect ratio attachments on emitted pixel buffers based on the image description.
kQTComponentPropertyListenerCollectionContextVersion	
kQTDRMPropertyID_InteractWithUser	Boolean.

kQTDRMPropertyID_IsAuthorized	Boolean.
kQTDRMPropertyID_IsProtected	Boolean.
kQTGetComponentPropertyInfoSelect	
kQTGetComponentPropertySelect	
kQTJPEGHuffmanTables	
kQTJPEGQuantizationTables	
kQTPropertyClass_DRM	
kQTPropertyClass_ICMCompressionFrameOptions	
kQTPropertyClass_ICMCompressionSession	Class identifier for compression session properties. Also 'icse'.
kQTPropertyClass_ICMCompressionSessionOptions	Class identifier for compression session option object properties. Also 'icso'.
kQTPropertyClass_ICMDecompressionFrameOptions	
kQTPropertyClass_ICMDecompressionSession	
kQTPropertyClass_ICMDecompressionSessionOptions	
kQTPropertyClass_ImageDescription	Class identifier for image description properties.
kQTRemoveComponentPropertyListenerSelect	
kQTSetComponentPropertySelect	
MediaSampleFlags	
QTComponentPropertyListenerCollectionContext	
QTComponentPropertyListenerFilterProcPtr	
QTComponentPropertyListenerFilterUPP	
QTComponentPropertyListenerProcPtr	
QTComponentPropertyListenersRef	
QTComponentPropertyListenerUPP	
QTVisualContextImageAvailableCallback	
QTVisualContextRef	
uppCallComponentAddComponentPropertyListenerProcInfo	
uppCallComponentGetComponentPropertyInfoProcInfo	

uppCallComponentGetComponentPropertyProcInfo	
uppCallComponentRemoveComponent-PropertyListenerProcInfo	
uppCallComponentSetComponentPropertyProcInfo	

MediaHandlers.h

Functions

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

DisposePrePrerollCompleteUPP	Disposes of a PrePrerollCompleteUPP pointer.
InvokePrePrerollCompleteUPP	
MediaGGetLatency	
NewPrePrerollCompleteUPP	Allocates a Universal Procedure Pointer for the PrePrerollCompleteProc callback.

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.

kMediaGenerateApertureModeDimensionsSelect	
kMediaGetApertureModeClipRectForSampleDescriptionIndexSelect	
kMediaGetApertureModeMatrixForSampleDescriptionIndexSelect	
kMediaGGetLatencySelect	
kMediaSetTrackApertureModeDimensionsUsingSampleDescriptionSelect	

MediaHandlers.k.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.

uppMediaGenerateApertureModeDimensionsProcInfo	

uppMediaGetApertureModeClipRectForSampleDescriptionIndexProcInfo	
uppMediaGetApertureModeMatrixForSampleDescriptionIndexProcInfo	
uppMediaGGetLatencyProcInfo	
uppMediaSetTrackApertureModeDimensionsUsingSampleDescriptionProcInfo	

Movies.h

Functions

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

AttachMovieToCurrentThread	Attaches a movie to the current thread.
AttachTimeBaseToCurrentThread	Attaches a time base to the current thread.
ConvertDataRefToMovieDataRef	Converts a piece of data in a storage location to a movie file format and stores it in another storage location, supporting a user settings dialog box for import operations.
ConvertMovieToDataRef	Converts a specified movie (or a single track within a movie) into a specified file format and stores it in a specified storage location.
ConvertTimeToClockTime	Converts a time record in a time base to clock time.
DetachMovieFromCurrentThread	Detaches a movie from the current thread.
DetachTimeBaseFromCurrentThread	Detaches a time base from the current thread.
DisposeMCActionNotificationUPP	
DisposeQTMoviePropertyListenerUPP	
EnterMoviesOnThread	Indicates that the client will be using QuickTime on the current thread.
ExitMoviesOnThread	Indicates to QuickTime that the client will no longer be using QuickTime on the current thread.
GetMovieRateChangeConstraints	Returns the minimum and maximum delay you can get when a movie's rate is changed.
GetMovieThreadAttachState	Determines whether a given movie is attached to a thread.
GetTimeBaseMasterOffsetTimeBase	Allows an offset time base to retrieve the master time base it is attached to.

GetTimeBaseRateChangeStatus	Lets a time base client determine the time base's last rate change status.
GetTimeBaseThreadAttachState	Determines whether a given time base is attached to a thread.
InvokeMCActionNotificationUPP	
InvokeQTMovieMovieMPropertyListenerUPP	
NewMCActionNotificationUPP	
NewQTMoviePropertyListenerUPP	
QTAddMoviePropertyListener	Installs a callback to monitor a movie property.
QTGetConnectionSpeedFromPrefs	
QTGetDataHandlerDirectoryDataReference	Returns a new data reference to the parent directory of the storage location associated with a data handler instance.
QTGetDataHandlerFullPathCFString	Returns the full pathname of the storage location associated with a data handler.
QTGetDataHandlerTargetNameCFString	Returns the name of the storage location associated with a data handler.
QTGetDataReferenceDirectoryDataReference	Returns a new data reference for a parent directory.
QTGetDataReferenceFullPathCFString	Returns the full pathname of the target of the data reference as a CFString.
QTGetDataReferenceTargetNameCFString	Returns the name of the target of a data reference as a CFString.
QTGetMovieProperty	Returns the value of a specific movie property.
QTGetMoviePropertyInfo	Returns information about the properties of a movie.
QTNewDataReferenceFromCFURL	Creates a URL data reference from a CFURL.
QTNewDataReferenceFromFSRef	Creates an alias data reference from a file specification.
QTNewDataReferenceFromFSRefCFString	Creates an alias data reference from a file reference pointing to a directory and a file name.
QTNewDataReferenceFromFSSpec	Creates an alias data reference from a file specification of type FSSpec.
QTNewDataReferenceFromFullPathCFString	Creates an alias data reference from a CFString that represents the full pathname of a file.

QTNewDataReferenceFromURLCFString	Creates a URL data reference from a CFString that represents a URL string.
QTNewDataReferenceWithDirectoryCFString	Creates an alias data reference from another alias data reference pointing to the parent directory and a CFString that contains the file name.
QTRemoveMoviePropertyListener	Removes a movie property monitoring callback.
QTSetMovieProperty	Sets the value of a specific movie property.
SetTimeBaseOffsetTimeBase	Attaches an offset time base to another time base.

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	
ConnectionSpeedIsValidPrefsType	
ConstQTPropertyValuePtr	
fullScreenCaptureAllDisplays	Capturealldisplays is a Mac OS X specific parameter.
fullScreenCaptureDisplay	Capturedisplay is a Mac OS X specific parameter.
fullScreenDontSwitchMonitorResolution	
hintsAllowIdleSleep	Asks media handlers not to call UpdateSystemActivity etc.
hintsDeinterlaceFields	
hintsRenderingMode	
kActionListAddAttribute	(C string elementPath, long atIndex, C string newAttributeName).
kActionListRemoveAttributes	(C string elementPath, long startIndex, long endIndex).
kActionListSetAttributeValue	(C string elementPath, C string attributeName, C string valueString).

kCharacteristicSupportsApertureModes	Indicates that a media handler supports aperture modes, which enable video to be automatically scaled and cropped to compensate for non-square pixel aspect ratios and to trim possibly-dirty edge processing regions. The dimensions of such a track may change when the movie's aperture mode is changed.
kCharacteristicSupportsDisplayOffsets	
kEffectParentWindowCarbon	
kKeyboardAllowFocus	
kKeyboardHaveFocus	
kKeyboardValidate	
kOperandListCountAttributes	(C string elementPath).
kOperandListGetAttributeNameByIndex	(C string elementPath, long index).
kOperandListGetAttributeValue	(C string elementPath, C string attributeName).
kPropertyClass_MetaData	
kPropertyClass_MetaDataItem	
kQTAnimatedGIFLoopCountInfinite	
kQTAudioMeter_DeviceMix	Meter the movie's mix to the AudioChannelLayout of the device the movie is playing to. To determine the channel layout of this mix, you call the kAudioPropertyID_-DeviceChannelLayout movie property.
kQTAudioMeter_MonoMix	Meter the movie as if it had been mixed to monaural. This option is offered only for MovieAudioFrequencyMetering.
kQTAudioMeter_StereoMix	Meter a stereo (two-channel) mix of the enabled sound tracks in the movie. This option is offered only for MovieAudioFrequencyMetering.
kQTAudioPropertyID_Balance	
kQTAudioPropertyID_BitRateString	
kQTAudioPropertyID_ChannelLayout	
kQTAudioPropertyID_ChannelLayoutString	

kQTAudioPropertyID_DeviceASBD	This is a get-only property and returns the AudioStreamBasicDescription of the device the movie is playing to. The interesting fields are the sample rate, which reflects device's current state, and the number of channels, which matches what is reported by kQTAudioPropertyID_-DeviceChannelLayout.
kQTAudioPropertyID_DeviceChannelLayout	
kQTAudioPropertyID_FormatString	
kQTAudioPropertyID_Gain	
kQTAudioPropertyID_Mute	
kQTAudioPropertyID_Pitch	The movie pitch adjustment. This adjusts the pitch of all audio tracks that contribute to the AudioContext mix. Pitch control takes effect only if kQTAudioPropertyID_-RateChangesPreservePitch is in effect; otherwise, returns kQTMessageNotHandledErr. The Float32 value is specified in cents: 0.0 == no change, 1.0 == one cent up, 100.0 == one semi-tone up, -1.0 == one cent down. The most useful ranges for pitch are +/- 1200, that is, one octave.
kQTAudioPropertyID_RateChangesPreservePitch	This property was introduced in QuickTime 7 and must be set in order for pitch changes to take effect. When the playback rate is not unity, audio must be resampled in order to play at the new rate. The default resampling affects the pitch of the audio (for example, playing at 2x speed raises the pitch by an octave, 1/2x lowers an octave). If this property is set on the Movie, an alternative algorithm may be used, which alters the speed without changing the pitch. Because this is more computationally expensive, this property may be silently ignored on some slow CPUs. Media handlers may query this movie property and honor it when performing Scaled Edits. This property can be specified as a property to the NewMovieFromProperties API. Currently, it has no effect when set on an open movie.

kQTAudioPropertyID_RenderQuality	Movie audio render quality takes effect for movie playback. UInt32 values are as defined in AudioUnit/AudioUnitProperties.h and vary from 0x00 (kRenderQuality_Min) to 0x7F (kRenderQuality_Max). A special value kQTAudioRenderQuality_PlaybackDefault) is also defined which resets the quality settings of the playback processing chain to values that are chosen to be an optimal balance of performance and quality.
kQTAudioPropertyID_SampleRateString	
kQTAudioPropertyID_SampleSizeString	
kQTAudioPropertyID_SummaryASBD	Get-only. Returns the AudioStreamBasicDescription corresponding to the Summary Mix of a movie. This describes non-interleaved, Float32 linear PCM data, with a sample rate equal to the highest audio sample rate found among the sound tracks contributing to the AudioContext mix, and a number of channels that matches what is reported by kQTAudioPropertyID_SummaryChannelLayout.
kQTAudioPropertyID_SummaryChannelLayout	
kQTAudioPropertyID_SummaryString	
kQTAudioRenderQuality_PlaybackDefault	
kQTContextPropertyID_AudioContext	
kQTContextPropertyID_VisualContext	
kQTDataLocationPropertyID_CFStringHFSPath	
kQTDataLocationPropertyID_CFStringNativePath	
kQTDataLocationPropertyID_CFStringPosixPath	
kQTDataLocationPropertyID_CFStringWindowsPath	
kQTDataLocationPropertyID_CFURL	
kQTDataLocationPropertyID_DataFork	
kQTDataLocationPropertyID_DataReference	
kQTDataLocationPropertyID_LegacyMovieResourceHandle	
kQTDataLocationPropertyID_MovieUserProc	
kQTDataLocationPropertyID_QTDataHandler	

kQTDataLocationPropertyID_ResourceFork	
kQTDataLocationPropertyID_Scrap	
kQTEnterMoviesFlagDontSetComponentsThreadMode	See Movies.h.
kQTHFSPathStyle	The Macintosh HFS file system syntax where the delimiters are colons.
kQTMCActionNotifyAfter	
kQTMCActionNotifyBefore	
kQTMCActionNotifyCancelled	
kQTMCActionNotifyParamChanged	
kQTMCActionNotifySignature	
kQTMCActionNotifyUserFilterCancelled	
kQTMetaDataCommonKeyAlbum	Information to come.
kQTMetaDataCommonKeyArtist	
kQTMetaDataCommonKeyArtwork	
kQTMetaDataCommonKeyAuthor	
kQTMetaDataCommonKeyChapterName	
kQTMetaDataCommonKeyComment	
kQTMetaDataCommonKeyComposer	
kQTMetaDataCommonKeyCopyright	
kQTMetaDataCommonKeyDescription	
kQTMetaDataCommonKeyDirector	
kQTMetaDataCommonKeyDisplayName	
kQTMetaDataCommonKeyGenre	
kQTMetaDataCommonKeyInformation	
kQTMetaDataCommonKeyKeywords	
kQTMetaDataCommonKeyOriginalFormat	
kQTMetaDataCommonKeyOriginalSource	
kQTMetaDataCommonKeyPerformers	
kQTMetaDataCommonKeyProducer	

kQTMetaDataCommonKeySoftware	
kQTMetaDataCommonKeyWriter	
kQTMetaDataItemPropertyID_DataType	
kQTMetaDataItemPropertyID_Key	
kQTMetaDataItemPropertyID_KeyFormat	
kQTMetaDataItemPropertyID_Locale	
kQTMetaDataItemPropertyID_StorageFormat	
kQTMetaDataItemPropertyID_Value	
kQTMetaDataItemUninitialized	
kQTMetaDataKeyFormatCommon	
kQTMetaDataKeyFormatiTunesLongForm	Reverse DNS format.
kQTMetaDataKeyFormatiTunesShortForm	FourCharCode.
kQTMetaDataKeyFormatQuickTime	Reverse DNS format.
kQTMetaDataKeyFormatUserData	FourCharCode.
kQTMetaDataKeyFormatWildcard	Match any key regardless of key format.
kQTMetaDataPropertyID_Owner	
kQTMetaDataPropertyID_OwnerType	
kQTMetaDataPropertyID_StorageFormats	
kQTMetaDataStorageFormatiTunes	ITunes metadata storage format.
kQTMetaDataStorageFormatQuickTime	QuickTime metadata storage format.
kQTMetaDataStorageFormatUserData	UserData storage format.
kQTMetaDataStorageFormatWildcard	
kQTMetaDataTypeBinary	
kQTMetaDataTypeBMPImage	
kQTMetaDataTypeFloat32BE	
kQTMetaDataTypeFloat64BE	
kQTMetaDataTypeJPEGImage	
kQTMetaDataTypeMacEncodedText	

kQTMetaDataTypePNGImage	
kQTMetaDataTypeSignedIntegerBE	
kQTMetaDataTypeUnsignedIntegerBE	
kQTMetaDataTypeUTF16BE	
kQTMetaDataTypeUTF8	
kQTMovieAudioExtractionAudioPropertyID_-AudioChannelLayout	
kQTMovieAudioExtractionAudioPropertyID_-AudioStreamBasicDescription	
kQTMovieAudioExtractionAudioPropertyID_RenderQuality	Sets the render quality to be used for this audio extraction session. UInt32 values are as defined in AudioUnit/AudioUnitProperties.h and vary from 0x00(kRenderQuality_Min) to 0x7F (kRenderQuality_Max). A special value (kQTAudioRenderQuality_PlaybackDefault) is also defined which resets the quality settings to the same values that were chosen by default for playback.
kQTMovieAudioExtractionComplete	
kQTMovieAudioExtractionMoviePropertyID_-AllChannelsDiscrete	
kQTMovieAudioExtractionMoviePropertyID_CurrentTime	
kQTMovieInstantiationPropertyID_AsyncOK	
kQTMovieInstantiationPropertyID_-DontAskUnresolvedDataRefs	
kQTMovieInstantiationPropertyID_DontAutoAlternates	
kQTMovieInstantiationPropertyID_DontAutoUpdateClock	
kQTMovieInstantiationPropertyID_DontResolveDataRefs	
kQTMovieInstantiationPropertyID_-DontUpdateForeBackPointers	
kQTMovieInstantiationPropertyID_IdleImportOK	
kQTMovieInstantiationPropertyID_-ResultDataLocationChanged	
kQTMovieResourceLocatorPropertyID_Callback	
kQTMovieResourceLocatorPropertyID_FileOffset	

kQTMovieResourceLocatorPropertyID_LegacyResID	
kQTMovieResourceLocatorPropertyID_LegacyResName	
kQTNativeDefaultPathStyle	The default pathname syntax of the platform.
kQTNewMoviePropertyID_Active	
kQTNewMoviePropertyID_DefaultDataRef	
kQTNewMoviePropertyID_DontInteractWithUser	
kQTPOSIXPathStyle	Used on Unix-based systems where pathname components are delimited by slashes.
kQTPropertyClass_Audio	
kQTPropertyClass_Context	
kQTPropertyClass_DataLocation	
kQTPropertyClass_MovieAudioExtraction_Audio	
kQTPropertyClass_MovieAudioExtraction_Movie	
kQTPropertyClass_MovieInstantiation	
kQTPropertyClass_MovieResourceLocator	
kQTPropertyClass_NewMovieProperty	
kQTPropertyClass_SampleTable	
kQTPropertyClass_SoundDescription	Class for SoundDescription properties.
kQTPropertyClass_Visual	
kQTQuartzComposerMediaType	
kQTRestrictionClassPlay	
kQTRestrictionPlayDontPlay	
kQTSampleTableAnyAttributeChange	
kQTSampleTableAttribute_DataSizePerSampleChange	
kQTSampleTableAttribute_DecodeDurationChange	
kQTSampleTableAttribute_DiscontiguousData	
kQTSampleTableAttribute_DisplayOffsetChange	
kQTSampleTableAttribute_SampleDescriptionIDChange	
kQTSampleTableAttribute_SampleFlagsChange	

kQTSampleTablePropertyID_MaxDisplayOffset	
kQTSampleTablePropertyID_MaxRelativeDisplayTime	
kQTSampleTablePropertyID_MinDisplayOffset	
kQTSampleTablePropertyID_MinRelativeDisplayTime	
kQTSampleTablePropertyID_TotalDecodeDuration	
kQTSoundDescriptionKind_Movie_AnyVersion	
kQTSoundDescriptionKind_Movie_LowestPossibleVersion	
kQTSoundDescriptionKind_Movie_Version1	
kQTSoundDescriptionKind_Movie_Version2	
kQTSoundDescriptionPropertyID_AudioChannelLayout	
kQTSoundDescriptionPropertyID_- AudioStreamBasicDescription	
kQTSoundDescriptionPropertyID_BitRate	
kQTSoundDescriptionPropertyID_MagicCookie	
kQTSoundDescriptionPropertyID_UserReadableText	
kQTVisualPropertyID_ApertureMode	You can set the aperture mode property on a movie to indicate whether aspect ratio and clean aperture correction should be performed (kQTPropertyClass_Visual / kQTVisualPropertyID_ApertureMode). When a movie is in clean, production or encoded pixels aperture mode, each track's dimensions are overriden by special dimensions for that mode. The original track dimensions are preserved and can be restored by setting the movie into classic aperture mode. Aperture modes are not saved in movies. You can set the aperture mode property on a decompression session options object to indicate whether pixel buffers should be tagged to enable aspect ratio and clean aperture correction (kQTPropertyClass_-ICMDecompressionSessionOptions / kICMDecompressionSessionOptionsPropertyID_-ApertureMode).
kQTVisualPropertyID_Brightness	Float32, ReadWrite.

kQTVisualPropertyID_ClassicDimensions	The track dimensions used in QuickTime 7.0.x and earlier. Setting this property is equivalent to calling SetTrackDimensions, except that SetTrackDimensions also changes the aperture mode to kQTApertureMode_Classic, and setting this property does not.
kQTVisualPropertyID_CleanApertureDimensions	The track dimensions to use in clean aperture mode.
kQTVisualPropertyID_Contrast	Float32, ReadWrite.
kQTVisualPropertyID_EncodedPixelsDimensions	The track dimensions to use in encoded pixels aperture mode.
kQTVisualPropertyID_HasApertureModeDimensions	True if aperture mode dimensions have been set on this movie, even if they are all identical to the classic dimensions (as is the case for content with square pixels and no edge processing region). This property can also be tested on a movie, where it is true if any track has aperture mode dimensions.
kQTVisualPropertyID_Hue	Float32, ReadWrite.
kQTVisualPropertyID_ProductionApertureDimensions	The track dimensions to use in production aperture mode.
kQTVisualPropertyID_Saturation	Float32, ReadWrite.
kQTWindowsPathStyle	The Windows pathname syntax that uses backslashes as component delimiters.
kRefConNavigateClick	
kSystemIsClassicBlueBox	
kTx3gBlinkAtomType	
kTx3gFontTableAtomType	
kTx3gSampleType	
kUserDataAnimatedGIFBufferingSize	Data is big-endian UInt32.
kUserDataAnimatedGIFLoopCount	Data is big-endian UInt16.
mcActionAddActionNotification	Parameter is QTMCActionNotificationPtr.
mcActionChapterListChanged	No param.
mcActionCurrentChapterChanged	Parameter is a UInt32, new chapter index.
mcActionEditStateChanged	Parameter is a Boolean, editing enabled?.

mcActionGetConnectionStatus	Parameter is QTConnectionStatusPtr.
mcActionGetKeyboardFocus	Parameter is QTKeyboardFocusPtr.
mcActionKeyUp	Parameter is pointer to EventRecord.
mcActionMovieFinished	No param.
mcActionMovieLoadStateChanged	Parameter is SInt32, new load state.
MCActionNotificationProcPtr	
MCActionNotificationUPP	
mcActionRemoveActionNotification	Parameter is QTMCActionNotificationPtr.
mcActionSetKeyboardFocus	Parameter is QTKeyboardFocusPtr.
mediaSampleDependsOnOthers	Decoding this sample depends on decoding other samples.
mediaSampleDoesNotDependOnOthers	Decoding this sample does not depend on decoding other samples.
mediaSampleDroppable	Sample does not need to be decoded for later samples to be decoded properly.
mediaSampleEarlierDisplayTimesAllowed	Samples later in decode order may have earlier display times.
mediaSampleHasNoRedundantCoding	Sample is known not to contain redundant coding.
mediaSampleHasRedundantCoding	Sample is known to contain redundant coding.
mediaSampleIsDependedOnByOthers	One or more other samples depend on this sample being decoded.
mediaSampleIsNotDependedOnByOthers	Synonym for mediaSampleDroppable.
mediaSamplePartialSync	Sample is a partial sync sample (for example, 1 frame after open GOP).
MovieAudioExtractionRef	
nextTimePartialSyncSample	
pdActionConvertSettingsToText	Parameter is QTAtomContainer inbound, Handle outbound contains human readable text - createdDialog may be NIL.
pdActionConvertSettingsToXML	Parameter is QTAtomContainer inbound, Handle outbound contains the XML - createdDialog may be NIL.

pdActionConvertSettingsToXMLWithComments	Parameter is QTAtomContainer inbound, Handle outbound contains the XML with comments - createdDialog may be NIL.
pdActionConvertXMLToSettings	Parameter is Handle inbound, QTAtomContainer outbound contains parameters - createdDialog may be NIL.
pdActionRunInEventLoop	Parameter is QTEventLoopDescriptionPtr - OS X only.
pdActionSetPropertyComponent	Parameter is QTParamComponentPropertyPtr.
pdOptionsDisplayAsSheet	Dialog will be used as a sheet (on platforms that support it).
PtrToMovie	
QTAudioContextRef	
QTAudioFrequencyLevels	
QTAudioVolumeLevels	
QTCallBackOpaqueHeader	
QTConnectionStatusPtr	
QTConnectionStatusRecord	
QTEventLoopDescriptionPtr	
QTEventLoopDescriptionRecord	
QTKeyboardFocusPtr	
QTKeyboardFocusRecord	
QTMCActionNotificationPtr	
QTMCActionNotificationRecord	
QTMetaDataItem	
QTMetaDataKeyFormat	
QTMetaDataRef	
QTMetaDataStorageFormat	
QTMoviePropertyListenerProcPtr	
QTMoviePropertyListenerUPP	
QTMutableSampleTableRef	

QTNewMoviePropertyElement	
QTNewMovieUserProcRecord	
QTParamComponentPropertyPtr	
QTParamComponentPropertyRecord	
QTPathStyle	
QTPropertyClass	
QTPropertyID	
QTPropertyValuePtr	
QTPropertyValueType	
QTSampleDescriptionID	
QTSampleTableAttribute	
QTSampleTableRef	
QTSoundDescriptionKind	
QTTrackPropertyListenerProcPtr	
QTTrackPropertyListenerUPP	
SoundDescriptionV2	
SoundDescriptionV2Handle	
SoundDescriptionV2Ptr	
timeBaseRateChanging	The clock is waiting for a future time to start moving while its rate is nonzero. When set, the unpinned time will return a negative value telling how far you are from the real start time.
Tx3gDescription	
Tx3gDescriptionHandle	
Tx3gDescriptionPtr	
Tx3gFontRecord	
Tx3gFontRecordPtr	
Tx3gFontTableHandle	
Tx3gFontTablePtr	
Tx3gFontTableRecord	

Tx3gRGBAColor	
Tx3gStyleHandle	
Tx3gStylePtr	
Tx3gStyleRecord	
Tx3gStyleTableHandle	
Tx3gStyleTablePtr	
Tx3gStyleTableRecord	

MoviesFormat.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.

CompositionOffsetAtom	
CompositionOffsetEntry	
CompositionShiftLeastGreatestAtom	
kQTSampleDependency_DependsOnOthers	Not an I picture.
kQTSampleDependency_DoesNotDependOnOthers	An I picture.
kQTSampleDependency_EarlierDisplayTimesAllowed	Corresponds to flag mediaSampleEarlierDisplay-TimesAllowed except at different bit offset.
kQTSampleDependency_HasNoRedundantCoding	
kQTSampleDependency_HasRedundantCoding	
kQTSampleDependency_IsDependedOnByOthers	
kQTSampleDependency_IsNotDependedOnByOthers	MediaSampleDroppable.
PartialSyncSampleAtom	
QT_MEDIA_TYPE	
QT_MOVIE_TYPE	
QT_SOUND_TYPE	

QT_TRACK_TYPE	
QT_VIDEO_TYPE	
SampleDependencyAtom	
STCompositionOffsetAID	
STCompositionShiftLeastGreatestAID	
STPartialSyncSampleAID	
STSampleDependencyAID	
TrackApertureModeDimensionsAID	The container atom.
TrackCleanApertureDimensions	
TrackCleanApertureDimensionsAID	The container atom.
TrackCleanApertureDimensionsAtom	
TrackEncodedPixelsDimensions	
TrackEncodedPixelsDimensionsAID	The container atom.
TrackEncodedPixelsDimensionsAtom	
TrackProductionApertureDimensions	
TrackProductionApertureDimensionsAID	The container atom.
TrackProductionApertureDimensionsAtom	

QTML.h

Functions

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

QTMLCreateMutex	Creates a synchronization object to facilitate mutually exclusive access to a Windows data structure.
QTMLDestroyMutex	Deallocates a synchronization object created by QTMLCreateMutex.
QTMLGrabMutex	Confers ownership of a mutex created by QTMLCreateMutex.
QTMLReturnMutex	Releases ownership of a QTMLMutex object.
QTMLTryGrabMutex	Determines if you would be able to get immediate ownership of a mutex created by QTMLCreateMutex.

QTMLYieldCPU	Yields time to other threads while your code is in a tight loop.
QTMLYieldCPUTime	Yields time to other threads and specifies the sleep time while in a tight loop.

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.

kDirectoryPathOnly	The full pathname up to and including the enclosing directory but not the filename should be returned. This can be useful to get a path for the enclosing directory that might be used to find related files in the same directory.
kErrorIfFileNotFound	If this flag is set and the file designated by inName does not exist, a fnfErr error code (©43) is returned. Value = 1L<<31.
kFileNameOnly	Only the part of the pathname corresponding to the file should be returned. This might be useful to return a string for a Windows title.
kFullNativePath	The full pathname should be returned.
kFullPathSpecifiedMask	
kInitializeQTMLDisableDDClippers	Flag for requesting QTML not to use DirectDraw clipper objects; QTML 5.0 and later.
kInitializeQTMLDisableDirectSound	Disable QTML's use of DirectSound.
kInitializeQTMLEnableDoubleBufferedSurface	Flag for requesting QuickTime use a double-buffered destination surface; QT6.4 and later.
kInitializeQTMLNoSoundFlag	If this flag is set, the Sound Manager is not initialized and therefore no sound APIs will be supported during the session. Use this flag only if no sound support is needed.
kInitializeQTMLUseExclusiveFullScreenModeFlag	Operate exclusively in full screen mode, in versions of QuickTime later than 3.0.
kInitializeQTMLUseGDIFlag	If this flag is set, all drawing will be done using graphics device interface (GDI) calls; neither DirectDraw nor DCI services will be used for onscreen graphics support.

kQTMLHandlePortEvents	If this flag is set, QTML will call the Win32 functions PeekMessage, TranslateMessage, and DispatchMessage to process Win32 messages while in tight spin loops.
kQTMLIsDoubleBuffered	
kQTMLNoDoubleBufferPort	Ask for QTML not to double-buffer this port.
kQTMLNoIdleEvents	Asks QTML not to send idle events.
kTryVDIMask	
kUFSFullPathName	
QTMLMutex	Represents a type used by the Windows API API.
QTMLSyncVar	
QTMLSyncVarPtr	

QTStreamingComponents.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.

kRTPAMRMediaPacketizerType	
kRTPAMRReassemblerType	
kRTPMediaPacketizerInfoRezType	

QuickTimeComponents.h

Functions

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

ClockGetRateChangeConstraints	Obtains minimum and maximum delays that a clock could introduce during a rate change.
ClockGetTimesForRateChange	

DisposeMovieExportStageReachedCallbackUPP	Disposes of a MovieExportStageReachedCallbackUPP pointer.
InvokeMovieExportStageReachedCallbackUPP	
MovieImportDoUserDialogDataRef	Requests that a movie import component display its user dialog box.
MovieImportSetMediaDataRef	Specifies a storage location that is to receive imported movie data.
NewMovieExportStageReachedCallbackUPP	Allocates a new Universal Procedure Pointer for a MovieExportStageReachedCallbackProc callback.
QTVideoOutputCopyIndAudioOutputDeviceUID	Identifies the audio device being used by a video output component.
VDCopyPreferredAudioDevice	
VDIIDCGetCSRData	Reads a camera's CSR registers directly.
VDIIDCGetDefaultFeatures	Places atoms in a QuickTime atom container that specify the default capabilities and default state of a camera's IIDC features.
VDIIDCGetFeatures	Places atoms in a QuickTime atom container that specify the current capabilities of a camera and the state of its IIDC features.
VDIIDCGetFeaturesForSpecifier	Places atoms in a QuickTime atom container that specify the current state of a single camera IIDC feature or group of features.
VDIIDCSetCSRData	Writes to a camera's CSR registers directly.
VDIIDCSetFeatures	Changes the state of a camera's IIDC features.

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.

channelPlayPostConversion	
channelPlayPostMix	
channelPlayPreConversion	
channelPlayPreMix	
kClockGetRateChangeConstraintsSelect	
kClockGetTimesForRateChangeSelect	

kDataHGetAvailableFileSize64Select	
kDataHGetDataAvailability64Select	
kMovieImportDoUserDialogDataRefSelect	
kMovieImportSetMediaDataRefSelect	
kQTAudioDeviceAttribute_DefaultInputDeviceKey	
kQTAudioDeviceAttribute_DefaultOutputDeviceKey	
kQTAudioDeviceAttribute_DefaultSystemOutputDeviceKey	
kQTAudioDeviceAttribute_DeviceAliveKey	
kQTAudioDeviceAttribute_DeviceCanPreviewKey	
kQTAudioDeviceAttribute_DeviceCanRecordKey	
kQTAudioDeviceAttribute_DeviceHoggedKey	
kQTAudioDeviceAttribute_DeviceInputDescription	
kQTAudioDeviceAttribute_DeviceInputID	
kQTAudioDeviceAttribute_DeviceInUseKey	
kQTAudioDeviceAttribute_DeviceManufacturerKey	
kQTAudioDeviceAttribute_DeviceNameKey	
kQTAudioDeviceAttribute_DeviceOutputDescription	
kQTAudioDeviceAttribute_DeviceOutputID	
kQTAudioDeviceAttribute_- DeviceSupportsHardwarePlaythruKey	
kQTAudioDeviceAttribute_DeviceTransportTypeKey	
kQTAudioDeviceAttribute_DeviceUIDKey	
kQTAudioDeviceAttribute_InputListWithAttributesKey	
kQTAudioDeviceAttribute_InputSelectionKey	
kQTAudioDeviceAttribute_IsCoreAudioDeviceKey	
kQTAudioDeviceAttribute_OutputListWithAttributesKey	
kQTAudioDeviceAttribute_OutputSelectionKey	
kQTFileType3GP2	
kQTFileType3GPP	

kQTFileTypeAMC	
kQTFileTypeAMR	
kQTFileTypeJPEG2000	
kQTFileTypePDF	
kQTFileTypeSDV	
kQTMediaConfigCanDoFileAssociation	Can configure this file association.
kQTMediaConfigTakeFileAssociationByDefault	
kQTMovieExporterPropertyID_DeinterlaceVideo	
kQTMovieExporterPropertyID_- EnableHighResolutionAudioFeatures	
kQTMovieExporterPropertyID_StageReachedCallback	
kQTMovieExportStage_AllTracksAddedToMovie	
kQTMovieExportStage_EmptyMovieCreated	
kQTPropertyClass_MovieExporter	
kQTPropertyClass_SCAudio	
kQTPropertyClass_SGAudio	
kQTPropertyClass_SGAudioPreviewDevice	
kQTPropertyClass_SGAudioRecordDevice	
kQTSCAudioPropertyID_ApplicableChannelLayoutTagList	
kQTSCAudioPropertyID_- ApplicableChannelLayoutTagNamesList	
kQTSCAudioPropertyID_- ApplicableLPCMBitsPerChannelList	
kQTSCAudioPropertyID_ApplicableSampleRateList	
kQTSCAudioPropertyID_AvailableChannelLayoutTagList	
kQTSCAudioPropertyID_- AvailableChannelLayoutTagNamesList	
kQTSCAudioPropertyID_AvailableCompressionFormatList	
kQTSCAudioPropertyID_- AvailableCompressionFormatNamesList	
kQTSCAudioPropertyID_AvailableLPCMBitsPerChannelList	

kQTSCAudioPropertyID_AvailableSampleRateList	
kQTSCAudioPropertyID_BasicDescription	
kQTSCAudioPropertyID_BitRate	
kQTSCAudioPropertyID_ChannelLayout	
kQTSCAudioPropertyID_ChannelLayoutName	
kQTSCAudioPropertyID_- ClientRestrictedChannelLayoutTagList	
kQTSCAudioPropertyID_- ClientRestrictedCompressionFormatList	
kQTSCAudioPropertyID_- ClientRestrictedLPCMBitsPerChannelList	
kQTSCAudioPropertyID_ClientRestrictedLPCMFlags	
kQTSCAudioPropertyID_ClientRestrictedSampleRateList	
kQTSCAudioPropertyID_CodecSpecificSettingsArray	
kQTSCAudioPropertyID_ConstantBitRateFormatsOnly	
kQTSCAudioPropertyID_ExtendedProcs	
kQTSCAudioPropertyID_HasLegacyCodecOptionsDialog	
kQTSCAudioPropertyID_InputBasicDescription	
kQTSCAudioPropertyID_InputChannelLayout	
kQTSCAudioPropertyID_InputChannelLayoutName	
kQTSCAudioPropertyID_InputMagicCookie	
kQTSCAudioPropertyID_InputSoundDescription	
kQTSCAudioPropertyID_MagicCookie	
kQTSCAudioPropertyID_MaximumOutputPacketSize	
kQTSCAudioPropertyID_OutputFormatIsExternallyFramed	
kQTSCAudioPropertyID_PreferenceFlags	

kQTSCAudioPropertyID_RenderQuality	Specifies the quality with which QuickTime should render the audio stream during the compression/decompression/transcode operation. Accepted constants are those used by AudioUnits (defined in AudioUnitProperties.h): kRenderQuality_Max, kRenderQuality_High, kRenderQuality_Medium, kRenderQuality_Low, kRenderQuality_Min. Available in QuickTimeComponents.h.
kQTSCAudioPropertyID_SampleRateIsRecommended	
kQTSCAudioPropertyID_SettingsState	
kQTSCAudioPropertyID_SoundDescription	
kQTSCAudioPropertyID_WindowOptions	
kQTSettingsCleanAperture	The clean aperture for compression sessions. If this is all zeros, it means no clean aperture (that is, full width and height).
kQTSettingsDeinterlaceSource	If TRUE, deinterlacing is applied to source frames.
kQTSettingsImageHeight	
kQTSettingsImageWidth	The destination width. If this is zero, it means the source width.
kQTSettingsPixelAspectRatio	The pixel aspect ratio for compression sessions. If this is all zeros, it means square pixels (that is, 1:1).
kQTSettingsScalingMode	The scaling mode for compression sessions. If this is zero, it means scaling mode based on the source aperture mode.
kQTSettingsUseCodecEnforcedDimensions	If TRUE, compressor's enforced dimension overrides the image size settings.
kQTSettingsVideoSize	The video size-related container.
kQTSGAudioPropertyID_AveragePowerLevels	
kQTSGAudioPropertyID_ChannelLayout	
kQTSGAudioPropertyID_ChannelMap	

kQTSGAudioPropertyID_ChunkSize	
kQTSGAudioPropertyID_DeviceAlive	
kQTSGAudioPropertyID_DeviceAttributes	
kQTSGAudioPropertyID_DeviceHogged	
kQTSGAudioPropertyID_DeviceInUse	
kQTSGAudioPropertyID_DeviceListWithAttributes	
kQTSGAudioPropertyID_DeviceUID	
kQTSGAudioPropertyID_GainScalarToDecibels	
kQTSGAudioPropertyID_HardwarePlaythruEnabled	
kQTSGAudioPropertyID_InputListWithAttributes	
kQTSGAudioPropertyID_InputSelection	
kQTSGAudioPropertyID_LevelMetersEnabled	
kQTSGAudioPropertyID_MagicCookie	
kQTSGAudioPropertyID_MasterGain	
kQTSGAudioPropertyID_MixerCoefficients	
kQTSGAudioPropertyID_OutputListWithAttributes	
kQTSGAudioPropertyID_OutputSelection	
kQTSGAudioPropertyID_PeakHoldLevels	
kQTSGAudioPropertyID_PerChannelGain	
kQTSGAudioPropertyID_PostConversionCallback	
kQTSGAudioPropertyID_PostConversionCallbackFormat	
kQTSGAudioPropertyID_PostMixCallback	
kQTSGAudioPropertyID_PostMixCallbackFormat	
kQTSGAudioPropertyID_PreConversionCallback	
kQTSGAudioPropertyID_PreConversionCallbackFormat	
kQTSGAudioPropertyID_PreMixCallback	
kQTSGAudioPropertyID_PreMixCallbackFormat	
kQTSGAudioPropertyID_Settings	

kQTSGAudioPropertyID_SoftPreviewLatency	
kQTSGAudioPropertyID_SoundDescription	
kQTSGAudioPropertyID_StreamFormat	
kQTSGAudioPropertyID_StreamFormatList	
kQTSpecialScalingMode_FitWithinDimensions	Adjusts destination dimensions so that the source fits within the dimensions specified with kQTSettingsImageWidth and kQTSettingsImageHeight by fitting to the shortest side, and scales the source to the destination. Internally, the default scaling mode, which is based on the source aperture mode, is used for compression session, instead of this scaling mode..
kQTVideoOutputCopyIndAudioOutputDeviceUIDSelect	
kSCAudioFillBufferSelect	
kSCAudioInvokeLegacyCodecOptionsDialogSelect	
kSCAudioResetSelect	
kSCCopyCompressionSessionOptionsSelect	
kVDCopyPreferredAudioDeviceSelect	
kVDIIDCGetCSRDataSelect	
kVDIIDCGetDefaultFeaturesSelect	
kVDIIDCGetFeaturesForSpecifierSelect	
kVDIIDCGetFeaturesSelect	
kVDIIDCSetCSRDataSelect	
kVDIIDCSetFeaturesSelect	
movieExportChannelLayoutSize	
movieExportMagicCookieSize	
movieExportSourceApertureMode	A pointer to an OSType. The source movie's aperture mode.
MovieExportStageReachedCallbackProcPtr	
MovieExportStageReachedCallbackProcRecord	
MovieExportStageReachedCallbackUPP	

movieExportUseHighResolutionAudioProperties	
scAllowEncodingWithCompressionSession	
SCAudioFormatFlagsRestrictions	
SCAudioInputDataProc	
scDisableFrameReorderingItem	
scDisableMultiPassEncodingItem	
scVideoAllowFrameReorderingType	Pointer to Boolean.
SCVideoMultiPassEncodingSettings	
scVideoMultiPassEncodingSettingsType	
scWindowOptionsType	Pointer to SCWindowSettings struct.
scWindowRefKindCarbon	WindowRef.
SCWindowSettings	
seqGrabDontPreAllocateFileSize	
seqGrabRecordPreferQualityOverFrameRate	
SGAudioCallback	
SGAudioCallbackFlags	
SGAudioCallbackStruct	
SGAudioMediaType	
sgcAudioChannelLayout	
sgcAudioChannelMap	
sgcAudioChunkSize	
sgcAudioDeviceName	
sgcAudioDeviceUID	
sgcAudioHardwarePlaythruEnabled	
sgcAudioInputSelection	
sgcAudioLevelMetersEnabled	
sgcAudioMagicCookie	
sgcAudioMasterGain	

sgcAudioMixerCoefficients	
sgcAudioOutputSelection	
sgcAudioOutputSettingsAtom	
sgcAudioPerChannelGain	
sgcAudioPreviewDeviceSettingsAtom	
sgcAudioRecordDeviceSettingsAtom	
sgcAudioSettingsVersion	
sgcAudioSoftPreviewLatency	
sgcAudioStreamFormat	
StandardCompressionSubTypeAudio	
vdIIDCAtomIDFeatureAtomTypeAndID	
vdIIDCAtomIDFeatureSettings	
vdIIDCAtomIDFocusPointSettings	
vdIIDCAtomIDLightingHintSettings	
vdIIDCAtomIDTriggerSettings	
vdIIDCAtomTypeFeature	
vdIIDCAtomTypeFeatureAtomTypeAndID	
vdIIDCAtomTypeFeatureSettings	
vdIIDCAtomTypeFocusPointSettings	
vdIIDCAtomTypeLightingHintSettings	
vdIIDCAtomTypeTriggerSettings	
VDIIDCFeatureAtomTypeAndID	
vdIIDCFeatureBrightness	
VDIIDCFeatureCapabilities	
vdIIDCFeatureCaptureQuality	
vdIIDCFeatureCaptureSize	
vdIIDCFeatureEdgeEnhancement	
vdIIDCFeatureExposure	

vdIIDCFeatureFlagAbsoluteControl	Set if the feature's value can be specified in absolute units. Absolute values are expressed in engineering units, such as dB or degrees.
vdIIDCFeatureFlagAuto	Set if the feature can be put into automatic mode.
vdIIDCFeatureFlagManual	Set if the feature can be put into manual mode.
vdIIDCFeatureFlagOff	Set if the feature can be turned off.
vdIIDCFeatureFlagOn	Set if feature can be turned on.
vdIIDCFeatureFlagRawControl	Set if the feature's value can be specified in raw values. Raw values are unitless and their meaning can vary from camera to camera.
vdIIDCFeatureFlagTune	Set if the feature can be tuned. When tuned, a feature drops into automatic mode until it stabilizes and then it reverts to manual mode.
vdIIDCFeatureFocus	
vdIIDCFeatureFocusPoint	
vdIIDCFeatureGain	
vdIIDCFeatureGamma	
vdIIDCFeatureHue	
vdIIDCFeatureIris	
vdIIDCFeatureLightingHint	
vdIIDCFeatureOpticalFilter	
vdIIDCFeaturePan	
vdIIDCFeatureSaturation	
VDIIDCFeatureSettings	
vdIIDCFeatureSharpness	
vdIIDCFeatureShutter	
VDIIDCFeatureState	
vdIIDCFeatureTemperature	
vdIIDCFeatureTilt	

vdIIDCFeatureTrigger	
vdIIDCFeatureWhiteBalanceU	
vdIIDCFeatureWhiteBalanceV	
vdIIDCFeatureZoom	
VDIIDCFocusPointSettings	
vdIIDCGroupColor	
vdIIDCGroupImage	
vdIIDCGroupMechanics	
vdIIDCGroupTrigger	
vdIIDCLightingHintLow	Set if the camera is using or should be using the low light hint.
vdIIDCLightingHintNormal	Set if the camera is using or should be using the normal light hint.
VDIIDCLightingHintSettings	
VDIIDCTriggerCapabilities	
vdIIDCTriggerFlagAbsoluteControl	Set if the trigger's value is or should be specified in absolute (engineering) units of time.
vdIIDCTriggerFlagActiveHigh	Set if the trigger is or should be active high.
vdIIDCTriggerFlagActiveLow	Set if the trigger is or should be active low.
vdIIDCTriggerFlagMode0	Set if the trigger is or should operate in mode 0. In mode 0, the camera starts integrating light at the active edge of the external trigger. The integration time is controlled by the shutter feature.
vdIIDCTriggerFlagMode1	Set if the trigger is or should operate in mode 1. In mode 1, the camera starts integrating light at the active edge of the external trigger. The integration time is equal to the time the trigger is active.

vdIIDCTriggerFlagMode2	Set if the trigger is or should operate in mode 2. In mode 2, the camera starts integrating light at the active edge of the external trigger. Integration continues until the nth active edge, where n is greater than or equal to 2.
vdIIDCTriggerFlagMode3	Set if the trigger is or should operate in mode 3.
vdIIDCTriggerFlagOff	Set if the trigger is or should be turned off.
vdIIDCTriggerFlagOn	Set if the trigger is or should be turned on.
vdIIDCTriggerFlagRawControl	Set if the trigger's value is or should be specified in unitless terms.
VDIIDCTriggerSettings	
VDIIDCTriggerState	
vdSubtypeIIDC	
xmlParseFlagPreserveWhiteSpace	Preserve whitespace throughout the document.

QuickTimeComponents.k.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.

uppDataHGetAvailableFileSize64ProcInfo	
uppDataHGetDataAvailability64ProcInfo	
uppVDCopyPreferredAudioDeviceProcInfo	

QuickTimeErrors.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.

kICMCodecCantQueueOutOfOrderErr	
kQTCannotCoerceValueErr	
kQTDisplayTimeAlreadyInUseErr	
kQTDisplayTimeTooEarlyErr	
kQTIncompatibleDescriptionErr	
kQTMediaDoesNotSupportDisplayOffsetsErr	
kQTMediaHasDisplayOffsetsErr	
kQTMessageCommandNotSupportedErr	
kQTMessageNoSuchParameterErr	
kQTMessageNotHandledErr	
kQTMetaDataInvalidItemErr	
kQTMetaDataInvalidKeyFormatErr	
kQTMetaDataInvalidMetaDataErr	
kQTMetaDataInvalidStorageFormatErr	
kQTMetaDataNoMoreItemsErr	
kQTObsoleteLPCMSoundFormatErr	
kQTPropertyArrayElementUnprocessedErr	
kQTPropertyAskLaterErr	
kQTPropertyBadValueSizeErr	
kQTPropertyNotSupportedErr	
kQTPropertyReadOnlyErr	
kQTTimeValueTooBigErr	
kQTVisualContextNotAllowedErr	
kQTVisualContextRequiredErr	

10.2 Symbol Changes

This article lists the symbols added to `QuickTime.framework` in Mac OS X v10.2.

C Symbols

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

ImageCodec.h

Functions

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

`ImageCodecDITLEvent`	Lets an image codec component receive and process dialog events.
`ImageCodecDITLInstall`	Installs added items in an image codec settings dialog box before the dialog box is displayed to the user.
`ImageCodecDITLItem`	Receives and processes mouse clicks in the image codec settings dialog box.
`ImageCodecDITLRemove`	Removes a panel from the image codec settings dialog box.
`ImageCodecDITLValidateInput`	Validates the contents of the user dialog box for an image codec component.
`ImageCodecGetDITLForSize`	Returns the size of various dialog item lists.
`ImageCodecMergeFloatingImageOntoWindow`	Draws the current contents of a floating image.
`ImageCodecRemoveFloatingImage`	Hides an image codec's floating image without having to close the component.

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.

codecDrawsHigherQualityScaled	Indicates that the codec will draw a higher quality image if it performs scaling; for example, while drawing a wipe effect with a border.
codecImageBufferMemoryFlagsValid	Set by ImageCodecNewImageBufferMemory or NewImageGWorld to indicate that the codecImageBufferIsInAGPMemory and codecImageBufferIsInPCIPMemory flags have been set correctly.
codecSuggestedBufferSentinel	Codec public resource containing suggested data pattern to put past end of data buffer.
k3DMinorClass	"3D Transitions".
kAdjustmentMinorClass	"Adjustments".
kArtisticMinorClass	"Artistic.
kBlurMinorClass	"Blur".
kCompositorMajorClass	Multisource layer effects.
kCompositorMinorClass	"Compositors".
kCustomControl	Flags valid for popups.
kDistortMinorClass	"Distort".
kEffectMajorClassID	
kEffectMajorClassType	
kEffectMinorClassID	
kEffectMinorClassNameID	
kEffectMinorClassNameType	
kEffectMinorClassType	
kEffectPresetType	
kEffectsMinorClass	"Special Effects".
kFilterMajorClass	One source effects.
kFilterMinorClass	"Filters".
kGeneratorMajorClass	Zero source effects.
kGeneratorMinorClass	"Generators".
kICMImageBufferNoPreference	

kICMImageBufferPreferMainMemory	
kICMImageBufferPreferVideoMemory	
kImageCodecDITLEventSelect	
kImageCodecDITLInstallSelect	
kImageCodecDITLItemSelect	
kImageCodecDITLRemoveSelect	
kImageCodecDITLValidateInputSelect	
kImageCodecGetDITLForSizeSelect	
kImageCodecMergeFloatingImageOntoWindowSelect	
kImageCodecRemoveFloatingImageSelect	
kMiscMajorClass	
kMiscMinorClass	
kNoiseMinorClass	"Noise".
kPresetNameID	
kPresetNameType	
kPresetPreviewPictureID	
kPresetPreviewPictureType	
kPresetSettingsID	
kPresetSettingsType	
kRenderMinorClass	"Render".
kSharpenMinorClass	"Sharpen".
kTransitionMajorClass	Multisource morph effects.
kTransitionMinorClass	"Transitions".
kWipeMinorClass	"Wipes".

ImageCompression.h

Functions

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

`GraphicsImportSetImageIndexToThumbnail`	Looks for a graphics subimage that contains a thumbnail.
`QTGetPixelFormatDepthForImageDescription`	For a given pixel format, returns the depth value that should be used in image descriptions.

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.

`codecDSequenceBidirectionalPrediction`	
`codecDSequenceDisableOverlaySurface`	
`codecDSequenceEnableSubPixelPositioning`	
`codecDSequenceFlushInsteadOfDirtying`	
`k16GrayPixelFormat`	
`k32AlphaGrayPixelFormat`	
`k422YpCbCr8PixelFormat`	
`k4444YpCbCrA8PixelFormat`	
`k4444YpCbCrA8RPixelFormat`	
`k48RGBPixelFormat`	
`k64ARGBPixelFormat`	
`kChannelCompositeEffectType`	
`kCMYKPixelFormat`	
`kGraphicsImporterEnableSubPixelPositioning`	
`kGraphicsImporterTrustResolutionFromFile`	
`kGraphicsImportSetImageIndexToThumbnailSelect`	
`kICMPixelFormatHasAlphaChannel`	

kICMPixelFormatIsMonochrome	
kMPEG4VisualCodecType	
kQTExifUserDataApertureValue	1 RATIONAL.
kQTExifUserDataBrightnessValue	1 SIGNED RATIONAL.
kQTExifUserDataColorSpace	1 SHORT.
kQTExifUserDataComponentsConfiguration	4 bytes.
kQTExifUserDataCompressedBitsPerPixel	1 RATIONAL.
kQTExifUserDataDateTimeDigitized	20 ASCIIs.
kQTExifUserDataDateTimeOriginal	20 ASCIIs.
kQTExifUserDataExifVersion	4 bytes (import only).
kQTExifUserDataExposureBiasValue	1 SIGNED RATIONAL.
kQTExifUserDataExposureIndex	1 RATIONAL.
kQTExifUserDataExposureProgram	1 SHORT.
kQTExifUserDataExposureTime	1 RATIONAL.
kQTExifUserDataFileSource	1 UNDEFINED.
kQTExifUserDataFlash	
kQTExifUserDataFlashEnergy	1 RATIONAL.
kQTExifUserDataFlashPixVersion	4 bytes.
kQTExifUserDataFNumber	1 RATIONAL.
kQTExifUserDataFocalLength	1 RATIONAL.
kQTExifUserDataFocalPlaneResolutionUnit	1 SHORT.
kQTExifUserDataFocalPlaneXResolution	1 RATIONAL.
kQTExifUserDataFocalPlaneYResolution	1 RATIONAL.
kQTExifUserDataGPSAltitude	
kQTExifUserDataGPSAltitudeRef	1 BYTE.
kQTExifUserDataGPSDestBearing	
kQTExifUserDataGPSDestBearingRef	2 ASCIIs.
kQTExifUserDataGPSDestDistance	

kQTExifUserDataGPSDestDistanceRef	2 ASCIIs.
kQTExifUserDataGPSDestLatitude	
kQTExifUserDataGPSDestLatitudeRef	2 ASCIIs.
kQTExifUserDataGPSDestLongitude	
kQTExifUserDataGPSDestLongitudeRef	2 ASCIIs.
kQTExifUserDataGPSDOP	1 RATIONAL.
kQTExifUserDataGPSImgDirection	
kQTExifUserDataGPSImgDirectionRef	2 ASCIIs.
kQTExifUserDataGPSLatitude	
kQTExifUserDataGPSLatitudeRef	2 ASCIIs.
kQTExifUserDataGPSLongitude	
kQTExifUserDataGPSLongitudeRef	2 ASCIIs.
kQTExifUserDataGPSMapDatum	N ASCII.
kQTExifUserDataGPSMeasureMode	2 ASCIIs.
kQTExifUserDataGPSSatellites	N ASCIIs.
kQTExifUserDataGPSSpeed	
kQTExifUserDataGPSSpeedRef	2 ASCIIs.
kQTExifUserDataGPSStatus	2 ASCIIs.
kQTExifUserDataGPSTimeStamp	3 RATIONALs.
kQTExifUserDataGPSTrack	
kQTExifUserDataGPSTrackRef	2 ASCIIs.
kQTExifUserDataGPSVersionID	4 BYTEs.
kQTExifUserDataISOSpeedRatings	N SHORTs.
kQTExifUserDataLightSource	1 SHORT.
kQTExifUserDataMakerNote	N bytes.
kQTExifUserDataMaxApertureValue	1 RATIONAL.
kQTExifUserDataMeteringMode	1 SHORT.
kQTExifUserDataPixelXDimension	1 SHORT or LONG.

kQTExifUserDataPixelYDimension	1 SHORT or LONG.
kQTExifUserDataRelatedSoundFile	13 ASCIIs.
kQTExifUserDataSceneType	1 UNDEFINED.
kQTExifUserDataSensingMethod	1 SHORT.
kQTExifUserDataShutterSpeedValue	1 SIGNED RATIONAL.
kQTExifUserDataSpectralSensitivity	N ASCIIs.
kQTExifUserDataSubjectDistance	1 RATIONAL.
kQTExifUserDataSubjectLocation	1 SHORT.
kQTExifUserDataSubSecTime	N ASCIIs.
kQTExifUserDataSubSecTimeDigitized	N ASCIIs.
kQTExifUserDataSubSecTimeOriginal	N ASCIIs.
kQTExifUserDataUserComment	N bytes.
kQTTIFFUserDataGeoAsciiParams	N ASCIIs.
kQTTIFFUserDataGeoDoubleParams	N DOUBLEs.
kQTTIFFUserDataGeoKeyDirectory	N SHORTs.
kQTTIFFUserDataIntergraphMatrix	16 or 17 DOUBLEs.
kQTTIFFUserDataModelPixelScale	3 DOUBLEs.
kQTTIFFUserDataModelTiepoint	N DOUBLEs.
kQTTIFFUserDataModelTransformation	16 DOUBLEs.
kQTTIFFUserDataOrientation	1 SHORT.
kQTTIFFUserDataPrimaryChromaticities	6 RATIONALs.
kQTTIFFUserDataReferenceBlackWhite	N LONGs.
kQTTIFFUserDataTransferFunction	N SHORTs.
kQTTIFFUserDataTransferRange	6 SHORTs.
kQTTIFFUserDataWhitePoint	2 RATIONALs.
kQTTIFFUserDataYCbCrPositioning	1 SHORT.
kYUV420PixelFormat	
QTMediaContextID	

QTUUID	

MediaHandlers.h

Functions

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

MediaEmptyAllPurgeableChunks	Force QuickTime to empty all purgeable media chunks in this application.
MediaGetChunkManagementFlags	Returns the current settings of the media chunk management flags.
MediaGetPurgeableChunkMemoryAllowance	Returns the current purgeable chunk memory allowance.
MediaGGetIdleManager	Retrieves an Idle Manager object from a derived media handler.
MediaGSetIdleManager	Lets a derived media handler report its idling needs.
MediaNavigateTargetRefCon	Locates the object for keyboard focus.
MediaRefConGetProperty	Returns the current media handler state based on the property type.
MediaRefConSetProperty	Sets a new media handler state based on the property type.
MediaSetChunkManagementFlags	Sets application-global flags that control media chunk management.
MediaSetPurgeableChunkMemoryAllowance	Sets the maximum amount of memory that QuickTime will allow purgeable chunks to occupy.

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.

kEmptyPurgableChunksOverAllowance	Empty all purgeable chunks over the current chunk allowance. Set to TRUE by default.
kMediaEmptyAllPurgeableChunksSelect	
kMediaGetChunkManagementFlagsSelect	

kMediaGetPurgeableChunkMemoryAllowanceSelect	
kMediaGGetIdleManagerSelect	
kMediaGSetIdleManagerSelect	
kMediaNavigateTargetRefConSelect	
kMediaRefConGetPropertySelect	
kMediaRefConSetPropertySelect	
kMediaSetChunkManagementFlagsSelect	
kMediaSetPurgeableChunkMemoryAllowanceSelect	

MediaHandlers.k.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.

uppMediaEmptyAllPurgeableChunksProcInfo	
uppMediaGetChunkManagementFlagsProcInfo	
uppMediaGetPurgeableChunkMemoryAllowanceProcInfo	
uppMediaGGetIdleManagerProcInfo	
uppMediaGSetIdleManagerProcInfo	
uppMediaNavigateTargetRefConProcInfo	
uppMediaRefConGetPropertyProcInfo	
uppMediaRefConSetPropertyProcInfo	
uppMediaSetChunkManagementFlagsProcInfo	
uppMediaSetPurgeableChunkMemoryAllowanceProcInfo	

Movies.h

Functions

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

AddMovieToStorage	Adds a movie to a storage container that was created by CreateMovieStorage.
ChooseMovieClock	Searches media handlers to find the best clock for a movie.
CloseMovieStorage	Closes an open movie storage container.
CopyMediaUserData	Copies a source media's user data into a destination media's user data.
CopyMovieUserData	Copies a source movie's user data into a destination movie's user data.
CopyTrackUserData	Copies a source track's user data into a destination track's user data.
CopyUserData	Copies metadata items from the source user data container to the destination user data container.
CreateMovieControl	Creates a movie control object to pass to the Mac OS Control Manager.
CreateMovieStorage	Creates an empty storage location to hold a movie and opens a data handler to the stored movie with write permission.
DeleteMovieStorage	Deletes a movie storage container.
DisposeQTEffectListFilterUPP	Disposes of a QTEffectListFilterUPP pointer.
DisposeQTNextTaskNeededSoonerCallbackUPP	Disposes of a QTNextTaskNeededSoonerCallbackUPP pointer.
FlattenMovieDataToDataRef	Performs a flattening operation to a movie at a storage location.
InvokeQTEffectListFilterUPP	
InvokeQTNextTaskNeededSoonerCallbackUPP	
MCSetControllerCapabilities	Undocumented
MCSetIdleManager	Lets a movie controller component report its idling needs.
NewMovieForDataRefFromHandle	Creates a movie from a public movie handle, converting internal references to external references.
NewMovieFromStorageOffset	Creates a new movie based on the offset to data in a storage container.
NewQTEffectListFilterUPP	Allocates a Universal Procedure Pointer for the QTEffectListFilterProc callback.

NewQTNextTaskNeededSoonerCallbackUPP	Allocates a Universal Procedure Pointer for the QTNextTaskNeededSoonerCallbackProc callback.
OpenMovieStorage	Opens a data handler for movie storage.
PutMovieForDataRefIntoHandle	Puts a self-contained movie into a handle.
PutMovieIntoStorage	Writes a movie to a storage location managed by a data handler.
QTAddMovieError	Adds orthogonal errors to a movie's list of errors.
QTCreateUUID	Creates a 128-bit universal unique ID number.
QTDoTweenPtr	Runs a tween component and returns values in a pointer rather than a handle.
QTEqualUUIDs	Compares two 128-bit ID numbers.
QTGetEffectsListExtended	Provides for more advanced filtering of effects to be placed into the effect list.
QTGetMovieRestrictions	Returns the restrictions, if any, for a given movie.
QTGetSupportedRestrictions	Reports the movie restrictions enforced by the currently running version of QuickTime.
QTGetTimeUntilNextTask	Reports the duration until the next time QuickTime needs to run a task.
QTGetWallClockTimeBase	Returns the system's real-time time base.
QTIdleManagerClose	Closes the Mac OS Idle Manager.
QTIdleManagerGetNextIdleTime	Retrieves the next idle time known to the Idle Manager.
QTIdleManagerNeedsAnIdle	Tells the Idle Manager whether an idle will be required.
QTIdleManagerOpen	Opens the Mac OS Idle Manager.
QTIdleManagerSetNextIdleTime	Informs the idle manager of the next required idle time.
QTIdleManagerSetNextIdleTimeDelta	Informs the idle manager of the time from the currently set idle time to the next idle time required after it.
QTIdleManagerSetNextIdleTimeNever	Sets the next idle time indefinitely in the future.
QTIdleManagerSetNextIdleTimeNow	Requests an idle as soon as possible.
QTIdleManagerSetParent	Sets the parent of an Idle Manager instance.

QTInstallNextTaskNeededSoonerCallback	Installs a QTNextTaskNeededSoonerCallbackProc callback.
QTRestrictionsGetIndClass	Reports the class of a movie restriction.
QTRestrictionsGetInfo	Reports information about the restrictions in a specified restriction set.
QTRestrictionsGetItem	Retrieves specific movie restrictions.
QTUninstallNextTaskNeededSoonerCallback	Removes a QTNextTaskNeededSoonerCallbackProc callback.
SpriteMediaDisposeImage	Frees the memory allocated for a sprite image outside a movie and removes that image from the sprite track in which it appears.
SpriteMediaImageIDToIndex	Returns the index of an outside sprite image from the ID of that image.
SpriteMediaImageIndexToID	Returns the ID of an outside sprite image from the index of that image.
SpriteMediaNewImage	Creates a new movie sprite image outside a movie.
UpdateMovieInStorage	Updates a movie at a storage location.

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.

hintsEnableSubPixelPositioning	
hintsFlushVideoInsteadOfDirtying	
IdleManager	
kActionMovieGotoChapterByIndex	(short index).
kActionMovieGotoFirstChapter	No params.
kActionMovieGotoLastChapter	No params.
kActionMovieGotoNextChapter	No params.
kActionMovieGotoPreviousChapter	No params.
kActionMovieSetScale	(Fixed xScale, Fixed yScale).
kActionQTVRSetViewState	Long viewStateType, short state.
kActionSetRandomSeed	Long randomSeed.

kActionSpriteSetCanBeHitTested	(short canBeHitTested).
kActionSpriteTrackDisposeImage	(short imageIndex).
kActionSpriteTrackNewImage	(C string imageURL, QTAtomID desiredID).
kActionSpriteTrackSetAllSpritesHitTestingMode	
kCharacteristicHasSkinData	
kCharacteristicProvidesKeyFocus	
kControllerUnderstandsIdleManagers	
kFlashTrackPropertyAcceptAllClicks	
kGetSpriteWorldInvalidRegionAndLeaveIntact	
kGetSpriteWorldInvalidRegionAndThenSetEmpty	
kMCSetControllerCapabilitiesSelect	
kMCSetIdleManagerSelect	
kMovieControlDataManualIdling	
kMovieControlDataMovie	
kMovieControlDataMovieController	
kMovieControlOptionEnableEditing	Allows programmatic editing of the movie and enables drag and drop.
kMovieControlOptionHandleEditingHI	Installs event handler for Edit menu commands and menu updating; also asserts kMovieControlOptionEnableEditing.
kMovieControlOptionHideController	See Movies.h.
kMovieControlOptionLocateTopLeft	The movie is pinned to the top left of the localRect rather then being centered within it.
kMovieControlOptionManuallyIdled	Instead of being idled by the movie control event loop timer, this movie control is idled manually by the application.
kMovieControlOptionSetKeysEnabled	Allows the movie control to react to keystrokes and participate in the keyboard focus mechanism within the window.

kOperandMovieAnnotation	(c string requested, long flags).
kOperandMovieChapterCount	
kOperandMovieChapterIndex	
kOperandMovieChapterIndexByName	(c string name).
kOperandMovieChapterName	
kOperandMovieChapterNameByIndex	(short index).
kOperandMovieConnectionFlags	
kOperandMovieConnectionString	
kOperandQTVRViewState	
kOperandQTVRViewStateCount	
kOperandSpriteCanBeHitTested	Short.
kOperandSpriteTrackAllSpritesHitTestingMode	
kOperandSpriteTrackImageIDByIndex	Short imageIndex.
kOperandSpriteTrackImageIndexByID	QTAtomID.
kQTCopyUserDataMerge	Append all source items from the source user data container to the destination user data container. Do not delete items from the destination container.
kQTCopyUserDataReplace	Delete all user data items from the destination user data container and then add all source items to the destination.
kQTEventKeyUp	Qtevent.param1 = key, qtevent.param2 = modifiers, qtEvent.param3 = scanCode.
kQTRestrictionClassEdit	
kQTRestrictionClassSave	
kQTRestrictionEditDontClear	Disable movie deletion.
kQTRestrictionEditDontCopy	Disable movie copying.
kQTRestrictionEditDontCut	Disable the movie cut operation.
kQTRestrictionEditDontExtract	Don't allow any form of extraction of content while editing.
kQTRestrictionEditDontModify	Don't allow modification of content.

kQTRestrictionEditDontPaste	Disable movie pasting.
kQTRestrictionSaveDontAddMovieResource	
kQTRestrictionSaveDontExport	
kQTRestrictionSaveDontExtract	Don't allow any form of extraction of content while saving.
kQTRestrictionSaveDontFlatten	
kRefConNavigationNext	Tabbing direction is forward. If no item is currently selected, use the first item.
kRefConNavigationPrevious	Tabbing direction is backward. If no item is currently selected, use the last item.
kRefConPropertyCanHaveFocus	A Boolean value. TRUE means that the media handler supports keyboard focus control.
kRefConPropertyHasFocus	A Boolean value. TRUE means that the media handler has keyboard focus control.
kSpriteHitTestTreatAllSpritesAsHitTestableMode	Treat all sprites as hit testable. Value is 1.
kSpriteHitTestTreatAllSpritesAsNotHitTestableMode	Treat all sprites as not hit testable and pass mouse clicks on to other objects. Value is 2.
kSpriteHitTestUseSpritesOwnPropertiesMode	Base the hit testability of each sprite on its kSpritePropertyCanBeHitTested property. Value is 0.
kSpriteMediaDisposeImageSelect	
kSpriteMediaImageIDToIndexSelect	
kSpriteMediaImageIndexToIDSelect	
kSpriteMediaNewImageSelect	
kSpritePropertyCanBeHitTested	The propertyValue parameter is a Boolean that is TRUE (the default value) if the sprite receives mouse clicks and FALSE if the sprite passes mouse clicks on to other objects.
kSpriteTrackPreferredDepthCompatibilityMode	Let QuickTime override the user's bit depth setting for the track. Value is 0.
kSpriteTrackPreferredDepthModernMode	Treat the user's bit depth setting for the track as absolute. Value is 1.

kSpriteTrackPropertyAllSpritesHitTestingMode	The propertyValue parameter points to a new value for a constant (see below) that specifies the hit testability of the sprites in a track. To set this constant you must pass 'Trck' in the spriteID parameter.
kSpriteTrackPropertyPreferred-DepthInterpretationMode	The propertyValue parameter points to a new value for a constant (see below) that specifies whether the offscreen bit depth set by the user is preferred (requested) or actual (forced).
kSpriteWorldDirtyInsteadOfFlush	
kTrackDefaultFocusFlags	
kTrackFocusCanEditFlag	
kTrackFocusDefaultRefcon	
kTrackFocusOn	
kTrackHandlesTabs	This is reserved for a future release.
kTrackModifierObjectCanBeHitTested	
kTweenRecordIsAtInterruptTime	
kTweenRecordNoFlags	
newMovieDontInteractWithUser	
pdActionCompactSample	Parameter is QTAtomContainer with effect sample to compact, -createdDialog may be NIL.
pdActionCustomDisposeControl	Parameter is QTCustomControlNewPtr.
pdActionCustomDoEditCommand	Parameter is QTCustomControlDoEditCommand.
pdActionCustomGetEnableValue	Parameter is QTCustomControlGetEnableValuePtr.
pdActionCustomGetValue	Parameter is QTCustomControlGetValue.
pdActionCustomHandleEvent	Parameter is QTCustomControlHandleEventPtr.
pdActionCustomNewControl	Parameter is QTCustomControlNewPtr.
pdActionCustomPositionControl	Parameter is QTCustomControlPositionControlPtr.

pdActionCustomSetEditCallout	Parameter is QTCustomControlSetEditCalloutPtr.
pdActionCustomSetEditMenu	Parameter is QTCustomControlSetEditMenuPtr.
pdActionCustomSetFocus	Parameter is QTCustomControlSetFocusPtr.
pdActionCustomSetPreviewPicture	Parameter is QTCustomControlSetPreviewPicturePtr.
pdActionCustomSetSampleTime	Parameter is QTCustomControlSetSampleTimePtr.
pdActionCustomShowHideControl	Parameter is QTCustomControlShowHideControlPtr.
pdActionDoEditCommand	Parameter is long with menu command (that is, mcMenuCut etc).
pdActionFocusBackward	
pdActionFocusFirst	
pdActionFocusForward	
pdActionFocusLast	
pdActionFocusOff	
pdActionGetDialogSettings	Parameter is QTAtomContainer.
pdActionGetNextSample	Parameter is QTAtomContainer with effect sample to change - createdDialog may be NIL.
pdActionGetPreviousSample	Parameter is QTAtomContainer with effect sample to change - createdDialog may be NIL.
pdActionGetSubPanelMenuValue	Parameter is long and returns current sub-panel value selected by the effect.
pdActionSetDialogSettings	Parameter is QTAtomContainer.
pdActionSetEditCallout	Parameter is QTParamPreviewCalloutPtr, can be NIL.
pdActionSetSampleTime	Parameter is QTParamSampleTimePtr, can be NIL.
pdOptionsEditCurrentEffectOnly	List of effects will not be shown.
pdOptionsHidePreview	Preview item will not be shown.

pdSampleTimeDisplayOptionsNone	
QTCustomControlDoEditCommandPtr	
QTCustomControlDoEditCommandRecord	
QTCustomControlGetEnableValuePtr	
QTCustomControlGetEnableValueRecord	
QTCustomControlGetValuePtr	
QTCustomControlGetValueRecord	
QTCustomControlHandleEventPtr	
QTCustomControlHandleEventRecord	
QTCustomControlNewPtr	
QTCustomControlNewRecord	
QTCustomControlPositionControlPtr	
QTCustomControlPositionControlRecord	
QTCustomControlSetEditCalloutPtr	
QTCustomControlSetEditCalloutRecord	
QTCustomControlSetEditMenuPtr	
QTCustomControlSetEditMenuRecord	
QTCustomControlSetFocusPtr	
QTCustomControlSetFocusRecord	
QTCustomControlSetPreviewPicturePtr	
QTCustomControlSetPreviewPictureRecord	
QTCustomControlSetSampleTimePtr	
QTCustomControlSetSampleTimeRecord	
QTCustomControlShowHideControlPtr	
QTCustomControlShowHideControlRecord	
QTEffectListFilterProcPtr	
QTEffectListFilterUPP	Represents a type used by the Movie Toolkit API.

QTErrorReplacementPtr	Represents a type used by the Movie Toolkit API.
QTErrorReplacementRecord	
QTNextTaskNeededSoonerCallbackProcPtr	
QTNextTaskNeededSoonerCallbackUPP	Abst_-QTNextTaskNeededSoonerCallbackUPP
QTParamPreviewCalloutPtr	
QTParamPreviewCalloutRecord	
QTParamSampleTimePtr	
QTParamSampleTimeRecord	
QTRestrictionSet	Represents a type used by the Movie Toolkit API.
QTRestrictionSetRecord	
SkinMediaType	
spriteHitTestTreatAllSpritesAsHitTestable	Set this flag if you want the sprite world's hit testing to ignore the kSpritePropertyCanBeHitTested properties of its individual sprites and make them all hit testable. Individual sprite properties can be set by calling SetSpriteProperty.
TweenV2Record	

MoviesFormat.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.

FileTypeAID	
FileTypeAtom	
kQTFileTypeBrandISOFile	ISO Base Media files.
kQTFileTypeBrandMPEG4v1	MPEG-4 (ISOIEC 14496-1) version 1 files.
kQTFileTypeBrandMPEG4v2	MPEG-4 (ISOIEC 14496-1) version 2 files.

kQTFileTypeBrandQuickTimeMovie	QuickTime movie files.
SecureContentInfoAID	
SecureContentInfoAtom	
SecureContentSchemeInfoAID	
SecureContentSchemeInfoAtom	
SecureContentSchemeTypeAID	
SecureContentSchemeTypeAtom	

QTStreamingComponents.h

Functions

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

RTPRssmGetExtChunkAndIncrRefCount	Undocumented

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.

kQTSInfo_CompressionParams	QTAtomContainer.
kQTSInfo_PreviewWhileRecordingMode	Boolean.
kQTSInfo_SGChannelDeviceInput	Short.
kQTSInfo_SGChannelDeviceList	SGDeviceList.
kQTSInfo_SGChannelDeviceName	Handle.
kQTSInfo_SGChannelInputName	Handle.
kQTSInfo_SGChannelSettings	QTSSGChannelSettingsParams.
kQTSPushDataSourcerFlag_SampleTimeIsValid	
kRTPInfo_FormatString	
kRTPMP3MediaPacketizerType	
kRTPMP3ReassemblerType	

kRTPMPEG4AudioMediaPacketizerType	
kRTPMPEG4AudioReassemblerType	
kRTPMPEG4VideoMediaPacketizerType	
kRTPMPEG4VideoReassemblerType	
kRTPPayloadConformanceTag	
kRTPRssmGetExtChunkAndIncrRefCountSelect	
kRTPRssmInfo_MoreInitParams	
kRTPRssmMoreInitParamsVersion1	
kSHChunkFlagExtended	
kSHExtendedChunkFlag_HasFrameLengths	Frame length data is added.
kSHExtendedChunkFlag_HasSampleCount	Sample count data is added.
kSHNumExtendedDataLongs	
QTSSGChannelSettingsParams	
RTPDescParams	
RTPRssmMoreInitParams	
SHExtendedChunkRecord	

QuickTimeComponents.h

Functions

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

DataHDeleteFile	Deletes a data handler's data storage file.
DataHGetTemporaryDataRefCapabilities	Undocumented
DataHRenameFile	Undocumented
DataHSetIdleManager	Lets a data handler report its idling needs.
DataHSetMovieUsageFlags	Sets the way that a data handler appends data to its storage.
DataHUseTemporaryDataRef	Undocumented

DisposeCDataHandlerUPP	Disposes of a CDataHandlerUPP pointer.
InvokeCDataHandlerUPP	
MovieImportGetDestinationMediaType	Returns the current type of a movie importer's destination media.
MovieImportSetIdleManager	Lets a movie importer report its idling needs.
MovieImportSetNewMovieFlags	Implemented by a movie import component to determine the original flags for NewMovieFromDataRef.
NewCDataHandlerUPP	Allocates a Universal Procedure Pointer for the CDataHandlerProc callback.
QTVideoOutputBaseSetEchoPort	Called on the base video output component to inform it about a change in the echo port.
SGGetChannelDeviceAndInputNames	Returns the sequence grabber's current device and input names.
SGGetChannelRefCon	Returns a reference constant that was previously set by SGSetChannelRefCon.
SGPanelGetDITLForSize	Returns user interface elements that fit within a specified size panel.
SGSetChannelDeviceInput	Undocumented
SGSetChannelSettingsStateChanging	Tells a sequence grabber channel of the beginning and end of a group of setting calls.
SGSetSettingsSummary	Sets the summary of sequence grabber settings that is displayed in the lower left corner of the sequence grabber dialog.
VDCaptureStateChanging	Provides process information from a sequence grabber component to the VDIG.
VDGetDeviceNameAndFlags	Returns the current name and device visibility of a video digitizer.
VDGetUniqueIDs	Returns a unique identifier for a particular video digitizer device.
VDSelectUniqueIDs	Selects a video digitizer device by ID.
XMLParseSetCDataHandler	Undocumented

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.

CDataHandler	
CDataHandlerUPP	
ComponentPreflightFlags	
kDataHDeleteFileSelect	
kDataHGetTemporaryDataRefCapabilitiesSelect	
kDataHMovieUsageDoAppendMDAT	Set this flag to instruct the data handler to append 'wide' and 'mdat' atoms in an append call.
kDataHRenameFileSelect	
kDataHSetIdleManagerSelect	
kDataHSetMovieUsageFlagsSelect	
kDataHTempCreateFile	
kDataHTempOpenFile	
kDataHTempUseSameDirectory	
kDataHTempUseSameVolume	
kDataHUseTemporaryDataRefSelect	
kDataRefExtensionQuickTimeMediaType	
kMovieImportGetDestinationMediaTypeSelect	
kMovieImportSetIdleManagerSelect	
kMovieImportSetNewMovieFlagsSelect	
kQTBrowserInfoCanUseSystemFolderPlugin	
kQTBrowserInfoResourceType	
kQTBrowserInfoResourceVersion	
kQTFileTypeMP4	
kQTMediaConfigHasFileHasQTAtoms	The file has a "QuickTime like" file format.
kQTPreFlightOpenComponent	
kQTSettingsDVExportDVFormat	Exported DV Format, DV ('dv ') or DVCPRO ('dvp '). (OSType).
kQTVideoOutputBaseSetEchoPortSelect	
kSGGetChannelDeviceAndInputNamesSelect	

kSGGetChannelRefConSelect	
kSGLargestDITLSize	The h and v values are set to retrieve large size. Value is —2.
kSGPanelGetDITLForSizeSelect	
kSGSetChannelDeviceInputSelect	
kSGSetChannelSettingsStateChangingSelect	
kSGSetSettingsSummarySelect	
kSGSmallestDITLSize	The h and v values are set to retrieve small size. Value is —1.
kVDCaptureStateChangingSelect	
kVDGetDeviceNameAndFlagsSelect	
kVDGetUniqueIDsSelect	
kVDSelectUniqueIDsSelect	
kXMLParseSetCDataHandlerSelect	
movieImportMustGetDestinationMediaType	
scAvailableCompressionListType	Pointer to OSType Handle.
scSoundInputSampleRateType	Pointer to UnsignedFixed.
scSoundSampleRateChangeOK	Pointer to Boolean.
scSoundVBRCompressionOK	Pointer to Boolean.
seqGrabAlwaysUseTimeBase	Set this flag to 1 if you want your channel to tell VDIGs to always use timebase time instead of creating uniform frame durations. This produces more accurate live sync with audio.
seqGrabLowLatencyCapture	Set this flag to 1 if you want your channel to return the freshest frame possible during videoconferencing, live broadcast, and live image processing.
sgcSoundCodecSettingsType	
sgDeviceDisplayNameType	
SGDeviceInputList	
SGDeviceInputListPtr	
SGDeviceInputListRecord	

SGDeviceInputName	
sgDeviceInputNameFlagInputUnavailable	Set to 1 if the named sequence grabber input is not available.
sgDeviceListIncludeInputs	If you set this flag to 1, the sequence grabber gets the list of inputs as well as the list of devices. See the SGDeviceInputListRecord struct.
sgDeviceNameFlagShowInputsAsDevices	Undocumented.
sgDeviceUIDType	
sgInputUIDType	
sgSetSettingsBegin	Sequence grabber channel set calls about to start.
sgSetSettingsEnd	Finished sequence grabber channel set calls. Get ready to use the new settings.
vdDeviceFlagHideDevice	Hide the VDIG device. Set this flag for VDIGs that registers themselves, so then you can register a further VDIG for each device. If no hardware is available, returning this flag will omit the VDIG from the list.
vdDeviceFlagShowInputsAsDevices	Show VDIG inputs as separate devices. This flag is meant for components that register once and support multiple devices. The user interface is clearer if the inputs are presented as devices rather than inputs.
vdFlagCaptureAlwaysUseTimeBase	Use the timebase for every frame; don't worry about making durations uniform.
vdFlagCaptureIsForPreview	Capture is just to screen for preview purposes.
vdFlagCaptureIsForRecord	Capture is going to be recorded.
vdFlagCaptureLowLatency	Fresh frames are more important than delivering every frame; don't queue too much.
vdFlagCaptureSetSettingsBegin	
vdFlagCaptureSetSettingsEnd	
vdFlagCaptureStarting	Capture is about to start; allocate bandwidth.
vdFlagCaptureStopping	Capture is about to stop; stop queuing frames.

QuickTimeComponents.k.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.

uppDataHDeleteFileProcInfo	
uppDataHGetTemporaryDataRefCapabilitiesProcInfo	
uppDataHRenameFileProcInfo	
uppDataHSetIdleManagerProcInfo	
uppDataHSetMovieUsageFlagsProcInfo	
uppDataHUseTemporaryDataRefProcInfo	
uppVDCaptureStateChangingProcInfo	
uppVDGetDeviceNameAndFlagsProcInfo	
uppVDGetUniqueIDsProcInfo	
uppVDSelectUniqueIDsProcInfo	

QuickTimeStreaming.h

Functions

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

QTSPrefsGetInstantOnSettings	Undocumented
QTSPrefsSetInstantOnSettings	Undocumented

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.

kQTSDirectConnectSIPProtocol	
kQTSInfo_Authentication	
kQTSInfo_AverageFrameRate	

kQTSInfo_MediaPacketizer	
kQTSInfo_URL	
kQTSInstantOnFlag_Enable	Instant on is enabled for read or write operations.
kQTSInstantOnFlag_Permitted	Instant on is permitted, for read operations only.
kQTSInstantOnPrefsAtomType	
kQTSSIPDataType	
kQTSSIPProxyPrefsType	
QTSAuthenticationParams	
QTSInstantOnPref	

10.1 Symbol Changes

This article lists the symbols added to `QuickTime.framework` in Mac OS X v10.1.

C Symbols

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

ImageCompression.h

Functions

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

`GraphicsExportGetExifEnabled`	Returns the graphics exporter's current Exif export setting.
`GraphicsExportGetThumbnailEnabled`	Returns the current thumbnail creation setting for the graphics exporter when exporting Exif files.
`GraphicsExportSetExifEnabled`	Determines whether or not the graphics exporter component should create Exif files.
`GraphicsExportSetThumbnailEnabled`	Determines whether or not the graphics exporter component should create an embedded thumbnail inside an exported Exif file.
`GraphicsImportGetBaseDataOffsetAndSize64`	Undocumented

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.

`kGraphicsExportGetExifEnabledSelect`	
`kGraphicsExportGetThumbnailEnabledSelect`	
`kGraphicsExportSetExifEnabledSelect`	

kGraphicsExportSetThumbnailEnabledSelect	
kGraphicsImportGetBaseDataOffsetAndSize64Select	
kQTAlphaMode	UInt32; for example, graphicsModeStraightAlpha or graphicsModePreBlackAlpha.
kQTAlphaModePreMulColor	RGBColor; used if kQTAlphaMode is graphicsModePreMulColorAlpha.
kQTEnableExif	UInt8 (Boolean).
kQTIndexedImageIsLayer	The image at this index is a layer..
kQTIndexedImageIsMultiResolution	The image at this index is one of several identical images at different resolutions..
kQTIndexedImageIsPage	The image at this index is a page..
kQTIndexedImageIsThumbnail	
kQTIndexedImageType	
kQTMetaData	
kQTThumbnailSettings	
kQTTIFFExifGPSUserDataPrefix	Added to tag values in Exif GPS IFDs to generate user data codes. (0x6770 is 'gp'.).
kQTTIFFExifUserDataPrefix	Added to tag values in Exif IFDs to generate user data codes. (0x6578 is 'ex'.).
kQTTIFFUserDataPrefix	Added to some tag values in TIFF IFDs to generate user data codes. (0x7469 is 'ti'.).
kUserDataIPTC	
QTThumbnailSettings	

MediaHandlers.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.

kMediaQTIdleFrequencySelector	

MediaHandlers.k.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.

uppCallComponentExecuteWiredActionProcInfo	
uppMediaChangedNonPrimarySourceProcInfo	
uppMediaCompareProcInfo	
uppMediaCurrentMediaQueuedDataProcInfo	
uppMediaDisposeTargetRefConProcInfo	
uppMediaDoIdleActionsProcInfo	
uppMediaEmptySampleCacheProcInfo	
uppMediaEnterEmptyEditProcInfo	
uppMediaFlushNonPrimarySourceDataProcInfo	
uppMediaForceUpdateProcInfo	
uppMediaGetActionsForQTEventProcInfo	
uppMediaGetClockProcInfo	
uppMediaGetDrawingRgnProcInfo	
uppMediaGetEffectiveSoundBalanceProcInfo	
uppMediaGetEffectiveVolumeProcInfo	
uppMediaGetErrorStringProcInfo	
uppMediaGetGraphicsModeProcInfo	
uppMediaGetInvalidRegionProcInfo	
uppMediaGetMediaInfoProcInfo	
uppMediaGetMediaLoadStateProcInfo	
uppMediaGetNameProcInfo	
uppMediaGetNextBoundsChangeProcInfo	
uppMediaGetNextStepTimeProcInfo	
uppMediaGetOffscreenBufferSizeProcInfo	

uppMediaGetPublicInfoProcInfo	
uppMediaGetSampleDataPointerProcInfo	
uppMediaGetSoundBalanceProcInfo	
uppMediaGetSoundBassAndTrebleProcInfo	
uppMediaGetSoundEqualizerBandLevelsProcInfo	
uppMediaGetSoundEqualizerBandsProcInfo	
uppMediaGetSoundLevelMeterInfoProcInfo	
uppMediaGetSoundLevelMeteringEnabledProcInfo	
uppMediaGetSoundOutputComponentProcInfo	
uppMediaGetSrcRgnProcInfo	
uppMediaGetTrackOpaqueProcInfo	
uppMediaGetURLLinkProcInfo	
uppMediaGetUserPreferredCodecsProcInfo	
uppMediaGetVideoParamProcInfo	
uppMediaGGetStatusProcInfo	
uppMediaGSetActiveSegmentProcInfo	
uppMediaGSetVolumeProcInfo	
uppMediaHasCharacteristicProcInfo	
uppMediaHitTestForTargetRefConProcInfo	
uppMediaHitTestTargetRefConProcInfo	
uppMediaIdleProcInfo	
uppMediaInitializeProcInfo	
uppMediaInvalidateRegionProcInfo	
uppMediaMakeMediaTimeTableProcInfo	
uppMediaMCIsPlayerEventProcInfo	
uppMediaPrePrerollBeginProcInfo	
uppMediaPrePrerollCancelProcInfo	
uppMediaPrerollProcInfo	

uppMediaPutMediaInfoProcInfo	
uppMediaQueueNonPrimarySourceDataProcInfo	
uppMediaReleaseSampleDataPointerProcInfo	
uppMediaResolveTargetRefConProcInfo	
uppMediaSampleDescriptionB2NProcInfo	
uppMediaSampleDescriptionChangedProcInfo	
uppMediaSampleDescriptionN2BProcInfo	
uppMediaSetActionsCallbackProcInfo	
uppMediaSetActiveProcInfo	
uppMediaSetClipProcInfo	
uppMediaSetDimensionsProcInfo	
uppMediaSetDoMCActionCallbackProcInfo	
uppMediaSetGraphicsModeProcInfo	
uppMediaSetGWorldProcInfo	
uppMediaSetHandlerCapabilitiesProcInfo	
uppMediaSetHintsProcInfo	
uppMediaSetMatrixProcInfo	
uppMediaSetMediaTimeScaleProcInfo	
uppMediaSetMovieTimeScaleProcInfo	
uppMediaSetNonPrimarySourceDataProcInfo	
uppMediaSetPublicInfoProcInfo	
uppMediaSetRateProcInfo	
uppMediaSetScreenLockProcInfo	
uppMediaSetSoundBalanceProcInfo	
uppMediaSetSoundBassAndTrebleProcInfo	
uppMediaSetSoundEqualizerBandsProcInfo	
uppMediaSetSoundLevelMeteringEnabledProcInfo	
uppMediaSetSoundLocalizationDataProcInfo	

uppMediaSetSoundOutputComponentProcInfo	
uppMediaSetTrackInputMapReferenceProcInfo	
uppMediaSetUserPreferredCodecsProcInfo	
uppMediaSetVideoParamProcInfo	
uppMediaTargetRefConsEqualProcInfo	
uppMediaTimeBaseChangedProcInfo	
uppMediaTrackEditedProcInfo	
uppMediaTrackPropertyAtomChangedProcInfo	
uppMediaTrackReferencesChangedProcInfo	
uppMediaVideoOutputChangedProcInfo	

Movies.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.

kActionDontPassKeyEvent	No params.
kActionListServerQuery	(C string url, C string keyValuePairs, long flags, C string parentPath).
kActionLoadComponent	(ComponentDescription handlerDesc).
kActionQTVREnableHotSpot	Long ID, Boolean enable.
kActionQTVRShowHotSpots	Boolean show.
kActionQTVRTranslateObject	Float xMove, float yMove.
kActionSetFocus	[(TargetAtoms theObject)].
kActionTrackSetBassTreble	(short base, short treble).
kActionTrackSetIdleFrequency	(long frequency).
kGetMovieImporterAutoImportOnly	Reject aggressive movie importers which have dontAutoFileMovieImport set.
kOperandCanHaveFocus	[(TargetAtoms theObject)].
kOperandHasFocus	[(TargetAtoms theObject)].

kOperandMovieID	
kOperandMovieName	
kOperandQTVRHotSpotsVisible	
kOperandQTVRViewCenterH	
kOperandQTVRViewCenterV	
kOperandSpriteName	
kOperandTrackBass	
kOperandTrackID	
kOperandTrackIdleFrequency	
kOperandTrackName	
kOperandTrackTreble	
kQTAllowAggressiveImporters	Tells QuickTime to include movie importers for file types like PICT and TEXT that aren't traditionally thought of as movies. If this flag is clear, QuickTime excludes these movie importers.
kQTAppMessageDisplayChannels	Request for application to display the channel UI.
kQTAppMessageEnterFullScreenRequested	Request for application to turn on full screen mode.
kQTParseTextHREFChapter	String.
kSystemIsWindows9x	
kSystemIsWindowsNT	
mcActionGetMovieActive	Parameter is pointer to Boolean.
mcActionGetMovieID	Parameter is pointer to long.
mcActionGetMovieName	Parameter is a p String Handle.

QTStreamingComponents.h

Functions

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

QTSSourcerInitialize	Undocumented

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.

kQTSInfo_AudioAutoGainOnOff	Boolean - error if unavailable.
kQTSInfo_AudioGain	Fixed kFixed1 is unity gain.
kQTSInfo_CodecFlags	Pointer to CodecFlags.
kQTSInfo_CodecSettings	Pointer to Handle.
kQTSInfo_CompressionList	Pointer to OSType Handle.
kQTSInfo_CroppedInputRect	Rect - defined relative to kQTSInfo_FullInputRect below.
kQTSInfo_DataRateSettings	Pointer to SCDataRateSettings struct.
kQTSInfo_ForceKeyValue	Pointer to long.
kQTSInfo_FullInputRect	
kQTSInfo_InputDeviceName	
kQTSInfo_InputSourceName	
kQTSInfo_SoundChannelCount	Pointer to short.
kQTSInfo_SoundCompression	Pointer to OSType.
kQTSInfo_SoundSampleRate	Pointer to UnsignedFixed.
kQTSInfo_SoundSampleSize	Pointer to short.
kQTSInfo_SpatialSettings	Pointer to SCSpatialSettings struct.
kQTSInfo_TargetDataRate	UInt32 in bytes per second.
kQTSInfo_TemporalSettings	Pointer to SCTemporalSettings struct.
kQTSInfo_TimeScale	UInt32.
kQTSInfo_VideoBrightness	UInt16.
kQTSInfo_VideoContrast	UInt16.
kQTSInfo_VideoHue	UInt16.
kQTSInfo_VideoSaturation	UInt16.
kQTSInfo_VideoSharpness	UInt16.
kQTSSourcerInitializeSelect	

QuickTimeComponents.h

Functions

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

`DataHGetInfo`	Retrieves information from a data handler.

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.

`kDataHGetInfoSelect`	
`kQTMediaConfigAssociateByDefault`	Deprecated, use kQTMediaConfigTakeFile-AssociationByDefault instead.
`kQTMediaConfigAudioGroupID`	
`kQTMediaConfigBinaryFile`	File should be transfered in binary mode.
`kQTMediaConfigCanUseApp`	
`kQTMediaConfigCanUsePlugin`	This MIME type can be configured to use plug-in.
`kQTMediaConfigDefaultsMask`	
`kQTMediaConfigDefaultsShift`	((flags & kQTMediaConfigDefaultsMask) >> kQTMediaConfigDefaultsShift) to get default setting.
`kQTMediaConfigImageGroupID`	
`kQTMediaConfigInteractiveGroupID`	
`kQTMediaConfigMacintoshFile`	File's resource fork is significant.
`kQTMediaConfigMiscGroupID`	
`kQTMediaConfigMP3GroupID`	
`kQTMediaConfigMPEGGroupID`	
`kQTMediaConfigResourceType`	
`kQTMediaConfigResourceVersion`	
`kQTMediaConfigStreamGroupID`	

kQTMediaConfigTextFile	Not a bit, defined for clarity.
kQTMediaConfigUNUSED	Currently unused.
kQTMediaConfigUseAppByDefault	Use the application by default for this MIME type.
kQTMediaConfigUsePluginByDefault	Use the plug-in by default for this MIME type.
kQTMediaConfigVideoGroupID	
kQTMediaFileInfoHasChanged	
kQTMediaGroupResourceType	
kQTMediaGroupResourceVersion	
kQTMediaInfoMacGroup	
kQTMediaInfoMiscGroup	
kQTMediaInfoNetGroup	
kQTMediaInfoWinGroup	
kQTMediaMIMEInfoHasChanged	
seqGrabCanMoveWindowWhileRecording	

QuickTimeComponents.k.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.

uppDataHAddMovieProcInfo	
uppDataHAppend64ProcInfo	
uppDataHCanUseDataRefProcInfo	
uppDataHCloseForReadProcInfo	
uppDataHCloseForWriteProcInfo	
uppDataHCompareDataRefProcInfo	
uppDataHCreateFileProcInfo	
uppDataHCreateFileWithFlagsProcInfo	

uppDataHDoesBufferProcInfo	
uppDataHFinishDataProcInfo	
uppDataHFlushCacheProcInfo	
uppDataHFlushDataProcInfo	
uppDataHGetAvailableFileSizeProcInfo	
uppDataHGetCacheSizeLimitProcInfo	
uppDataHGetDataAvailabilityProcInfo	
uppDataHGetDataInBufferProcInfo	
uppDataHGetDataProcInfo	
uppDataHGetDataRateProcInfo	
uppDataHGetDataRefAsTypeProcInfo	
uppDataHGetDataRefExtensionProcInfo	
uppDataHGetDataRefProcInfo	
uppDataHGetDataRefWithAnchorProcInfo	
uppDataHGetDeviceIndexProcInfo	
uppDataHGetFileNameProcInfo	
uppDataHGetFileSize64ProcInfo	
uppDataHGetFileSizeAsyncProcInfo	
uppDataHGetFileSizeProcInfo	
uppDataHGetFileTypeOrderingProcInfo	
uppDataHGetFreeSpace64ProcInfo	
uppDataHGetFreeSpaceProcInfo	
uppDataHGetInfoFlagsProcInfo	
uppDataHGetInfoProcInfo	
uppDataHGetMacOSFileTypeProcInfo	
uppDataHGetMIMETypeAsyncProcInfo	
uppDataHGetMIMETypeProcInfo	
uppDataHGetMovieProcInfo	

uppDataHGetMovieWithFlagsProcInfo	
uppDataHGetPreferredBlockSizeProcInfo	
uppDataHGetScheduleAheadTimeProcInfo	
uppDataHGetVolumeListProcInfo	
uppDataHIsStreamingDataHandlerProcInfo	
uppDataHOpenForReadProcInfo	
uppDataHOpenForWriteProcInfo	
uppDataHPlaybackHints64ProcInfo	
uppDataHPlaybackHintsProcInfo	
uppDataHPollReadProcInfo	
uppDataHPreextend64ProcInfo	
uppDataHPreextendProcInfo	
uppDataHPutDataProcInfo	
uppDataHReadAsyncProcInfo	
uppDataHResolveDataRefProcInfo	
uppDataHScheduleData64ProcInfo	
uppDataHScheduleDataProcInfo	
uppDataHSetCacheSizeLimitProcInfo	
uppDataHSetDataRefExtensionProcInfo	
uppDataHSetDataRefProcInfo	
uppDataHSetDataRefWithAnchorProcInfo	
uppDataHSetFileSize64ProcInfo	
uppDataHSetFileSizeProcInfo	
uppDataHSetMacOSFileTypeProcInfo	
uppDataHSetTimeBaseProcInfo	
uppDataHSetTimeHintsProcInfo	
uppDataHTaskProcInfo	
uppDataHUpdateMovieProcInfo	

uppDataHWrite64ProcInfo	
uppDataHWriteProcInfo	
uppGraphicsImageImportGetSequenceEnabledProcInfo	
uppGraphicsImageImportSetSequenceEnabledProcInfo	
uppMIDIImportGetSettingsProcInfo	
uppMIDIImportSetSettingsProcInfo	
uppMovieExportDisposeGetDataAndPropertiesProcsProcInfo	
uppMovieExportNewGetDataAndPropertiesProcsProcInfo	
uppTextExportGetDisplayDataProcInfo	
uppTextExportGetSettingsProcInfo	
uppTextExportGetTimeFractionProcInfo	
uppTextExportSetSettingsProcInfo	
uppTextExportSetTimeFractionProcInfo	
uppVDAddKeyColorProcInfo	
uppVDClearClipRgnProcInfo	
uppVDCompressDoneProcInfo	
uppVDCompressOneFrameAsyncProcInfo	
uppVDDoneProcInfo	
uppVDGetActiveSrcRectProcInfo	
uppVDGetBlackLevelValueProcInfo	
uppVDGetBrightnessProcInfo	
uppVDGetClipStateProcInfo	
uppVDGetCLUTInUseProcInfo	
uppVDGetCompressionTimeProcInfo	
uppVDGetCompressionTypesProcInfo	
uppVDGetContrastProcInfo	
uppVDGetCurrentFlagsProcInfo	
uppVDGetDataRateProcInfo	

uppVDGetDigitizerInfoProcInfo	
uppVDGetDigitizerRectProcInfo	
uppVDGetDMADepthsProcInfo	
uppVDGetFieldPreferenceProcInfo	
uppVDGetHueProcInfo	
uppVDGetImageDescriptionProcInfo	
uppVDGetInputColorSpaceModeProcInfo	
uppVDGetInputFormatProcInfo	
uppVDGetInputGammaRecordProcInfo	
uppVDGetInputGammaValueProcInfo	
uppVDGetInputNameProcInfo	
uppVDGetInputProcInfo	
uppVDGetKeyColorProcInfo	
uppVDGetKeyColorRangeProcInfo	
uppVDGetMaskandValueProcInfo	
uppVDGetMaskPixMapProcInfo	
uppVDGetMaxAuxBufferProcInfo	
uppVDGetMaxSrcRectProcInfo	
uppVDGetNextKeyColorProcInfo	
uppVDGetNumberOfInputsProcInfo	
uppVDGetPlayThruDestinationProcInfo	
uppVDGetPLLFilterTypeProcInfo	
uppVDGetPreferredImageDimensionsProcInfo	
uppVDGetPreferredTimeScaleProcInfo	
uppVDGetSaturationProcInfo	
uppVDGetSharpnessProcInfo	
uppVDGetSoundInputDriverProcInfo	
uppVDGetSoundInputSourceProcInfo	

uppVDGetTimeCodeProcInfo	
uppVDGetVBlankRectProcInfo	
uppVDGetVideoDefaultsProcInfo	
uppVDGetWhiteLevelValueProcInfo	
uppVDGrabOneFrameAsyncProcInfo	
uppVDGrabOneFrameProcInfo	
uppVDPreflightDestinationProcInfo	
uppVDPreflightGlobalRectProcInfo	
uppVDReleaseAsyncBuffersProcInfo	
uppVDReleaseCompressBufferProcInfo	
uppVDResetCompressSequenceProcInfo	
uppVDSetBlackLevelValueProcInfo	
uppVDSetBrightnessProcInfo	
uppVDSetClipRgnProcInfo	
uppVDSetClipStateProcInfo	
uppVDSetCompressionOnOffProcInfo	
uppVDSetCompressionProcInfo	
uppVDSetContrastProcInfo	
uppVDSetDataRateProcInfo	
uppVDSetDestinationPortProcInfo	
uppVDSetDigitizerRectProcInfo	
uppVDSetDigitizerUserInterruptProcInfo	
uppVDSetFieldPreferenceProcInfo	
uppVDSetFrameRateProcInfo	
uppVDSetHueProcInfo	
uppVDSetInputColorSpaceModeProcInfo	
uppVDSetInputGammaRecordProcInfo	
uppVDSetInputGammaValueProcInfo	

uppVDSetInputProcInfo	
uppVDSetInputStandardProcInfo	
uppVDSetKeyColorProcInfo	
uppVDSetKeyColorRangeProcInfo	
uppVDSetMasterBlendLevelProcInfo	
uppVDSetPlayThruDestinationProcInfo	
uppVDSetPlayThruGlobalRectProcInfo	
uppVDSetPlayThruOnOffProcInfo	
uppVDSetPLLFilterTypeProcInfo	
uppVDSetPreferredImageDimensionsProcInfo	
uppVDSetPreferredPacketSizeProcInfo	
uppVDSetSaturationProcInfo	
uppVDSetSharpnessProcInfo	
uppVDSetTimeBaseProcInfo	
uppVDSetupBuffersProcInfo	
uppVDSetWhiteLevelValueProcInfo	
uppVDUseSafeBuffersProcInfo	
uppVDUseThisCLUTProcInfo	

QuickTimeMusic.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.

kNoteRequestSpecifyMIDIChannel	
NoteRequestMIDIChannel	

QuickTimeStreaming.h

Functions

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

`DisposeQTSPanelFilterUPP`	Disposes of a QTSPanelFilterUPP pointer.
`InvokeQTSPanelFilterUPP`	
`NewQTSPanelFilterUPP`	Allocates a Universal Procedure Pointer for the QTSPanelFilterProc callback.
`QTSInitializeMediaParams`	Undocumented
`QTSPrefsAddProxyUserInfo`	Undocumented
`QTSPrefsFindProxyUserInfoByType`	Undocumented
`QTSPresGetSettingsAsText`	Undocumented
`QTSPresSettingsDialogWithFilters`	Undocumented

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.

`kQTSInfo_SettingsText`	
`kQTSProxyUserInfoPrefsType`	
`kQTSSampleDataCallbackParamsVersion1`	
`kQTSSettingsTextDetails`	
`kQTSSettingsTextSummary`	
`kQTSSoundLevelMeterInfo`	
`kQTSSoundLevelMeteringEnabledInfo`	
`kQTSUserDataCallbackParamsVersion1`	
`QTSPanelFilterParams`	
`QTSPanelFilterProcPtr`	
`QTSPanelFilterUPP`	
`QTSSampleDataCallbackParams`	

QTSSettingsTextParams	
QTSUserDataCallbackParams	

Document Revision History

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

Date	Notes
2007-07-18	Updated with the symbols added to the QuickTime framework in Mac OS X v10.5.
2005-06-04	Updated for Mac OS X v10.4.
2005-04-29	New document that summarizes the symbols added to the QuickTime framework in Mac OS X v10.4.