QuickTime Movie Properties Reference

QuickTime > Movie Internals

2006-05-23

Contents

QuickTime Movie Properties Reference

Framework: Frameworks/QuickTime.framework

Declared in Movies.h

Overview

QuickTime movies and movie tracks have properties that an application can manage, including embedded metadata and sample tables that determine what, how, and when the movie will present its data.

Functions by Task

Working With QuickTime Metadata

`QTCopyMediaMetaData` (page 14)
: Retains a media's metadata object and returns it.

`QTCopyMovieMetaData` (page 14)
: Retains a movie's metadata object and returns it.

`QTCopyTrackMetaData` (page 15)
: Retains a track's metadata object and returns it.

`QTMetaDataAddItem` (page 19)
: Adds an inline metadata item to the metadata storage format.

`QTMetaDataGetItemProperty` (page 21)
: Returns a property of a metadata item.

`QTMetaDataGetItemPropertyInfo` (page 22)
: Returns information about a property of a metadata item.

`QTMetaDataGetItemValue` (page 23)
: Returns the value of a metadata item from an item identifier.

`QTMetaDataGetNextItem` (page 23)
: Returns the next metadata item corresponding to a specified key.

`QTMetaDataGetProperty` (page 25)
: Returns a property of a metadata object.

`QTMetaDataGetPropertyInfo` (page 25)
: Returns information about a property of a metadata object.

`QTMetaDataRelease` (page 26)
: Decrements the retain count of a metadata object.

`QTMetaDataRemoveItem` (page 27)
: Removes a metadata item from a storage format.

`QTMetaDataRemoveItemsWithKey` (page 27)
: Removes metadata items with a specific key from the storage format.

`QTMetaDataRetain` (page 28)
: Increments the retain count of a metadata object.

`QTMetaDataSetItem` (page 29)
: Sets the value of the metadata item from the item identifier.

`QTMetaDataSetItemProperty` (page 29)
: Sets a property of a metadata item.

`QTMetaDataSetProperty` (page 30)
: Sets a property of a metadata object.

Working With QuickTime Sample Tables

`QTSampleTableAddSampleDescription` (page 32)
: Adds a sample description to a sample table, returning a sample description ID that can be used to refer to it.

`QTSampleTableAddSampleReferences` (page 33)
: Adds sample references to a sample table.

`QTSampleTableCopySampleDescription` (page 34)
: Retrieves a sample description from a sample table.

`QTSampleTableCreateMutable` (page 35)
: Creates a new, empty sample table.

`QTSampleTableCreateMutableCopy` (page 36)
: Copies a sample table.

`QTSampleTableGetDataOffset` (page 36)
: Returns the data offset of a sample.

`QTSampleTableGetDataSizePerSample` (page 37)
: Returns the data size of a sample.

`QTSampleTableGetDecodeDuration` (page 37)
: Returns the decode duration of a sample.

`QTSampleTableGetDisplayOffset` (page 38)
: Returns the offset from decode time to display time of a sample.

`QTSampleTableGetNextAttributeChange` (page 38)
: Finds the next sample number at which one or more of a set of given sample attributes change.

`QTSampleTableGetNumberOfSamples` (page 40)
: Returns the number of samples in a sample table.

`QTSampleTableGetProperty` (page 40)
: Returns the value of a specific sample table property.

`QTSampleTableGetPropertyInfo` (page 41)
: Returns information about the properties of a sample table.

`QTSampleTableGetSampleDescriptionID` (page 43)
: Returns the sample description ID of a sample.

`QTSampleTableGetSampleFlags` (page 43)
: Returns the media sample flags of a sample.

`QTSampleTableGetTimeScale` (page 44)
: Returns the timescale of a sample table.

`QTSampleTableGetTypeID` (page 44)
: Returns the CFTypeID value for the current sample table.

`QTSampleTableRelease` (page 44)
: Decrements the retain count of a sample table.

`QTSampleTableReplaceRange` (page 45)
: Replaces a range of samples in a sample table with a range of samples from another sample table.

`QTSampleTableRetain` (page 46)
: Increments the retain count of a sample table.

`QTSampleTableSetProperty` (page 46)
: Sets the value of a specific sample table property.

`QTSampleTableSetTimeScale` (page 47)
: Changes the timescale of a sample table.

Supporting Functions

`DisposeQTTrackPropertyListenerUPP` (page 10)
: Disposes a track property listener UPP.

`InvokeQTTrackPropertyListenerUPP` (page 10)
: Invokes the specified property listener of a track.

`MusicMediaGetIndexedTunePlayer` (page 11)
: Undocumented

`NewQTTrackPropertyListenerUPP` (page 12)
: Creates a new callback to monitor a track property.

`QTAddMoviePropertyListener` (page 12)
: Installs a callback to monitor a movie property.

`QTAddTrackPropertyListener` (page 13)
: Installs a callback to monitor a track property.

`QTGetMovieProperty` (page 16)
: Returns the value of a specific movie property.

`QTGetMoviePropertyInfo` (page 16)
: Returns information about the properties of a movie.

`QTGetTrackProperty` (page 17)
: Returns the value of a specific track property.

`QTGetTrackPropertyInfo` (page 18)
: Returns information about the properties of a track.

`QTRemoveMoviePropertyListener` (page 31)
: Removes a movie property monitoring callback.

`QTRemoveTrackPropertyListener` (page 32)
: Removes a track property monitoring callback

`QTSetMovieProperty` (page 48)
: Sets the value of a specific movie property.

`QTSetTrackProperty` (page 49)
: Sets the value of a specific track property.

`DisposeQTBandwidthNotificationUPP` (page 57) Deprecated in Mac OS X v10.4
: Disposes of a QTBandwidthNotificationUPP pointer.

`NewQTBandwidthNotificationUPP` (page 57) Deprecated in Mac OS X v10.4
: Allocates a Universal Procedure Pointer for the QTBandwidthNotificationProc callback.

`QTBandwidthRelease` (page 58) Deprecated in Mac OS X v10.4
: Undocumented

`QTBandwidthRequest` (page 58) Deprecated in Mac OS X v10.4
: Undocumented

`QTBandwidthRequestForTimeBase` (page 59) Deprecated in Mac OS X v10.4
: Undocumented

`QTScheduledBandwidthRelease` (page 60) Deprecated in Mac OS X v10.4
: Undocumented

`QTScheduledBandwidthRequest` (page 61) Deprecated in Mac OS X v10.4
: Undocumented

Functions

DisposeQTTrackPropertyListenerUPP

Disposes a track property listener UPP.

```
void DisposeQTTrackPropertyListenerUPP (
   QTTrackPropertyListenerUPP userUPP
);
```

Parameters

userUPP
: A `QTTrackPropertyListenerUPP` pointer. See Universal Procedure Pointers in the QuickTime API Reference for more information.

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

Declared In
`Movies.h`

InvokeQTTrackPropertyListenerUPP

Invokes the specified property listener of a track.

```
void InvokeQTTrackPropertyListenerUPP (
   Track inTrack,
   QTPropertyClass inPropClass,
   QTPropertyID inPropID,
   void *inUserData,
   QTTrackPropertyListenerUPP userUPP
);
```

Parameters

`inTrack`

The track of this operation.

`inPropClass`

A property class.

`inPropID`

A property ID.

`inUserData`

A pointer to user data that will be passed to the callback.

`userUPP`

A `QTTrackPropertyListenerUPP` pointer.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`Movies.h`

MusicMediaGetIndexedTunePlayer

Undocumented

```
ComponentResult MusicMediaGetIndexedTunePlayer (
   ComponentInstance ti,
   long sampleDescIndex,
   ComponentInstance *tp
);
```

Parameters

`ti`

Undocumented

`sampleDescIndex`

Undocumented

`tp`

A pointer to a tune player component instance.

Return Value

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

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In
`Movies.h`

NewQTTrackPropertyListenerUPP

Creates a new callback to monitor a track property.

```
QTTrackPropertyListenerUPP NewQTTrackPropertyListenerUPP (
   QTTrackPropertyListenerProcPtr userRoutine
);
```

Parameters

`userRoutine`
: A pointer to a `QTTrackPropertyListenerProcPtr` callback.

Return Value
A new UPP; see Universal Procedure Pointers in the QuickTime API Reference.

Discussion
This routine creates a new callback to monitor a track property.

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

Declared In
`Movies.h`

QTAddMoviePropertyListener

Installs a callback to monitor a movie property.

```
OSErr QTAddMoviePropertyListener (
   Movie inMovie,
   QTPropertyClass inPropClass,
   QTPropertyID inPropID,
   QTMoviePropertyListenerUPP inListenerProc,
   void *inUserData
);
```

Parameters

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

`inPropClass`
: A property class.

`inPropID`
: A property ID.

`inListenerProc`
: A Universal Procedure Pointer to a `QTMoviePropertyListenerProc` callback.

`inUserData`
: A pointer to user data that will be passed to the callback.

Return Value
See `Error Codes` in the QuickTime API Reference. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 6.4.

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

Related Sample Code
QTAudioExtractionPanel

Declared In
`Movies.h`

QTAddTrackPropertyListener

Installs a callback to monitor a track property.

```
OSErr QTAddTrackPropertyListener (
   Track inTrack,
   QTPropertyClass inPropClass,
   QTPropertyID inPropID,
   QTTrackPropertyListenerUPP inListenerProc,
   void *inUserData
);
```

Parameters

`inTrack`
: The track for this operation.

`inPropClass`
: A property class.

`inPropID`
: A property ID.

`inListenerProc`
: A Universal Procedure Pointer to a `QTTrackPropertyListenerProc` callback.

`inUserData`
: A pointer to user data that will be passed to the callback.

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

Discussion
This routine installs a callback to monitor a track property.

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

Related Sample Code
QTAudioExtractionPanel

Declared In
`Movies.h`

QTCopyMediaMetaData

Retains a media's metadata object and returns it.

```
OSStatus QTCopyMediaMetaData (
   Media inMedia,
   QTMetaDataRef *outMetaData
);
```

Parameters

inMedia

The media for this operation. You obtain this media identifier from such functions as `NewTrackMedia` and `GetTrackMedia`.

outMetaData

A pointer to an opaque metadata object wrapper associated with the media passed in `inMedia`.

Return Value

Returns `invalidMedia` if the media passed in `inMedia` is invalid, or `noErr` if there is no error.

Discussion

This function returns the metadata object associated with a media. The object has retain/release semantics. It has already been retained before returning, but you should call `QTMetaDataRelease` (page 26) when you are done. Because the media can be disposed of at any time, the `QTMetaDataRef` may be valid when the media no longer exists. In this case, the function will fail with a `kQTMetaDataInvalidMetaDataErr` error.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

QTMetadataEditor

Declared In

`Movies.h`

QTCopyMovieMetaData

Retains a movie's metadata object and returns it.

```
OSStatus QTCopyMovieMetaData (
   Movie inMovie,
   QTMetaDataRef *outMetaData
);
```

Parameters

inMovie

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

outMetaData

A pointer to an opaque metadata object wrapper associated with the movie passed in `inMovie`.

Return Value

Returns `invalidMovie` if the movie passed in `inMovie` is invalid, or `noErr` if there is no error.

Discussion
This function returns the metadata object associated with a movie. The object has retain/release semantics. It has already been retained before returning, but you should call `QTMetaDataRelease` (page 26) when you are done. Because the movie can be disposed of at any time, the `QTMetaDataRef` may be valid when the movie no longer exists. In this case, the function will fail with a `kQTMetaDataInvalidMetaDataErr` error.

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

Related Sample Code
QTMetaData

QTMetadataEditor

Declared In
`Movies.h`

QTCopyTrackMetaData

Retains a track's metadata object and returns it.

```
OSStatus QTCopyTrackMetaData (
   Track inTrack,
   QTMetaDataRef *outMetaData
);
```

Parameters
`inTrack`

A track identifier, which your application obtains from such functions as `NewMovieTrack` and `GetMovieTrack`.

`outMetaData`

A pointer to an opaque metadata object wrapper associated with the track passed in `inTrack`.

Return Value
Returns `invalidMedia` if the track passed in `inTrack` is invalid, or `noErr` if there is no error.

Discussion
This function returns the metadata object associated with a track. The object has retain/release semantics. It has already been retained before returning, but you should call `QTMetaDataRelease` (page 26) when you are done. Because the track can be disposed of at any time, the `QTMetaDataRef` may be valid when the track no longer exists. In this case, the function will fail with a `kQTMetaDataInvalidMetaDataErr` error.

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

Related Sample Code
QTMetadataEditor

Declared In
`Movies.h`

QTGetMovieProperty

Returns the value of a specific movie property.

```
OSErr QTGetMovieProperty (
   Movie inMovie,
   QTPropertyClass inPropClass,
   QTPropertyID inPropID,
   ByteCount inPropValueSize,
   QTPropertyValuePtr outPropValueAddress,
   ByteCount *outPropValueSizeUsed
);
```

Parameters

inMovie

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

inPropClass

A property class.

inPropID

A property ID.

inPropValueSize

The size of the buffer allocated to hold the property value.

outPropValueAddress

A pointer to the buffer allocated to hold the property value.

outPropValueSizeUsed

On return, the actual size of the value written to the buffer.

Return Value

See `Error Codes` in the QuickTime API Reference. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 6.4.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

QTAudioExtractionPanel

QTExtractAndConvertToAIFF

QTExtractAndConvertToMovieFile

Declared In

`Movies.h`

QTGetMoviePropertyInfo

Returns information about the properties of a movie.

```
OSErr QTGetMoviePropertyInfo (
   Movie inMovie,
   QTPropertyClass inPropClass,
   QTPropertyID inPropID,
   QTPropertyValueType *outPropType,
   ByteCount *outPropValueSize,
   UInt32 *outPropertyFlags
);
```

Parameters

inMovie

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

inPropClass

A property class.

inPropID

A property ID.

outPropType

A pointer to memory allocated to hold the `property` type on return.

outPropValueSize

A pointer to memory allocated to hold the size of the property value on return.

outPropertyFlags

A pointer to memory allocated to hold property flags on return.

Return Value

See `Error Codes` in the QuickTime API Reference. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 6.4.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

QTAudioExtractionPanel

QTExtractAndConvertToAIFF

QTExtractAndConvertToMovieFile

Declared In

`Movies.h`

QTGetTrackProperty

Returns the value of a specific track property.

```
OSErr QTGetTrackProperty (
   Track inTrack,
   QTPropertyClass inPropClass,
   QTPropertyID inPropID,
   ByteCount inPropValueSize,
   QTPropertyValuePtr outPropValueAddress,
   ByteCount *outPropValueSizeUsed
);
```

Parameters

`inTrack`
 The track for this operation.

`inPropClass`
 A property class.

`inPropID`
 A property ID.

`inPropValueSize`
 The size of the buffer allocated to hold the property value.

`outPropValueAddress`
 A pointer to the buffer allocated to hold the property value.

`outPropValueSizeUsed`
 On return, the actual size of the value written to the buffer.

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

Discussion
This routine returns the value of a specific track property.

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

Related Sample Code
QTAudioExtractionPanel

Declared In
`Movies.h`

QTGetTrackPropertyInfo

Returns information about the properties of a track.

```
OSErr QTGetTrackPropertyInfo (
   Track inTrack,
   QTPropertyClass inPropClass,
   QTPropertyID inPropID,
   QTPropertyValueType *outPropType,
   ByteCount *outPropValueSize,
   UInt32 *outPropertyFlags
);
```

Parameters

inTrack

The track for this operation.

inPropClass

A property class.

inPropID

A property ID.

outPropType

A pointer to memory allocated to hold the `property` type on return.

outPropValueSize

A pointer to memory allocated to hold the size of the property value on return.

outPropertyFlags

A pointer to memory allocated to hold property flags on return.

Return Value

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

Discussion

This routine returns information about the properties of a track.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

QTAudioExtractionPanel

Declared In

`Movies.h`

QTMetaDataAddItem

Adds an inline metadata item to the metadata storage format.

```
OSStatus QTMetaDataAddItem (
   QTMetaDataRef inMetaData,
   QTMetaDataStorageFormat inMetaDataFormat,
   QTMetaDataKeyFormat inKeyFormat,
   const UInt8 *inKeyPtr,
   ByteCount inKeySize,
   const UInt8 *inValuePtr,
   ByteCount inValueSize,
   UInt32 inDataType,
   QTMetaDataItem *outItem
);
```

Parameters

inMetaData

The metadata object for this operation.

inMetaDataFormat

The metadata storage format used by the object passed in `inMetaData`. The format may be `UserData` storage, iTunes metadata storage, or QuickTime metadata storage. Not all objects will include all forms of storage, and other storage formats may appear in the future. You cannot pass `kQTMetaDataStorageFormatWildcard` to target all storage formats.

inKeyFormat

The format of the key.

inKeyPtr

A pointer to the key of the item to be fetched next. You may pass NULL in this parameter if you are not interested in any specific key.

inKeySize

The size of the key in bytes.

inValuePtr

A pointer to the value to be added. This can be NULL if `inValueSize` is 0.

inValueSize

The size of `inValuePtr` in bytes. Pass 0 if you want to add an item with no value.

inDataType

A data type from the following list: `kQTMetaDataTypeBinary` = 0, kQTMetaDataTypeUTF8 = 1, kQTMetaDataTypeUTF16BE = 2, `kQTMetaDataTypeMacEncodedText` = 3, `kQTMetaDataTypeSignedIntegerBE` = 21, `kQTMetaDataTypeUnsignedIntegerBE` = 22, kQTMetaDataTypeFloat32BE = 23, kQTMetaDataTypeFloat64BE = 24With `kQTMetaDataTypeSignedIntegerBE` and `kQTMetaDataTypeUnsignedIntegerBE`, the size of the integer is determined by the value size.

outItem

On return, a pointer to an opaque, unique UInt64 identifier of the newly added item. Your application can use this to identify the metadata item within a metadata object for other metadata functions. You may pass NULL if you are not interested in the identifier of the newly added item. This identifier does not need to be disposed of.

Return Value

Returns `kQTMetaDataInvalidMetaDataErr` if the metadata object or its reference is invalid, `kQTMetaDataInvalidStorageFormatErr` if the metatada storage format is invalid, `kQTMetaDataInvalidKeyErr` if the key or its format is invalid, or `noErr` if there is no error. See `Metadata Error Codes`.

Discussion
The data type of the metadata item is assumed to be binary.

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

Related Sample Code
QTMetadataEditor

Declared In
`Movies.h`

QTMetaDataGetItemProperty

Returns a property of a metadata item.

```
OSStatus QTMetaDataGetItemProperty (
   QTMetaDataRef inMetaData,
   QTMetaDataItem inItem,
   QTPropertyClass inPropClass,
   QTPropertyID inPropID,
   ByteCount inPropValueSize,
   QTPropertyValuePtr outPropValueAddress,
   ByteCount *outPropValueSizeUsed
);
```

Parameters

inMetaData
The metadata object for this operation.

inItem
The opaque, unique UInt64 identifier of the metadata item for this operation. Your application obtains this item identifier from such functions as `QTMetaDataAddItem` (page 19) and `QTMetaDataGetNextItem` (page 23).

inPropClass
The class of the property being asked about.

inPropID
The ID of the property being asked about.

inPropValueSize
`Size` of the buffer allocated to receive the property value.

outPropValueAddress
A pointer to the buffer allocated to receive the item's property value.

outPropValueSizeUsed
On return, the actual size of buffer space used.

Return Value
Returns `kQTMetaDataInvalidMetaDataErr` if the metadata object or its reference is invalid, `kQTMetaDataInvalidItemErr` if the metatada item ID is invalid, `errPropNotSupported` if the metatada object does not support the property being asked about, `buffersTooSmall` if the allocated buffer is too small to hold the property, or `noErr` if there is no error. See `Metadata Error Codes`.

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

Related Sample Code
QTMetaData

QTMetadataEditor

Declared In
`Movies.h`

QTMetaDataGetItemPropertyInfo

Returns information about a property of a metadata item.

```
OSStatus QTMetaDataGetItemPropertyInfo (
   QTMetaDataRef inMetaData,
   QTMetaDataItem inItem,
   QTPropertyClass inPropClass,
   QTPropertyID inPropID,
   QTPropertyValueType *outPropType,
   ByteCount *outPropValueSize,
   UInt32 *outPropFlags
);
```

Parameters

`inMetaData`
The metadata object for this operation.

`inItem`
The opaque, unique UInt64 identifier of the metadata item for this operation. Your application obtains this item identifier from such functions as `QTMetaDataAddItem` (page 19) and `QTMetaDataGetNextItem` (page 23).

`inPropClass`
The class of the property being asked about.

`inPropID`
The ID of the property being asked about.

`outPropType`
A pointer to the type of the returned property's value.

`outPropValueSize`
A pointer to the size of the returned property's value.

`outPropFlags`
On return, a pointer to flags representing the requested information about the item's property.

Return Value
Returns `kQTMetaDataInvalidMetaDataErr` if the metadata object or its reference is invalid, `kQTMetaDataInvalidItemErr` if the metatada item ID is invalid, `errPropNotSupported` if the metatada object does not support the item property being asked about, or `noErr` if there is no error. See `Metadata Error Codes`.

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

Related Sample Code
QTMetaData

QTMetadataEditor

Declared In
`Movies.h`

QTMetaDataGetItemValue

Returns the value of a metadata item from an item identifier.

```
OSStatus QTMetaDataGetItemValue (
   QTMetaDataRef inMetaData,
   QTMetaDataItem inItem,
   UInt8 *outValuePtr,
   ByteCount inValueSize,
   ByteCount *outActualSize
);
```

Parameters

`inMetaData`
The metadata object for this operation.

`inItem`
The opaque, unique UInt64 identifier of the metadata item for this operation. Your application can obtain this item identifier from such functions as `QTMetaDataAddItem` (page 19).

`outValuePtr`
A pointer to the first value of the item. You may pass NULL in this parameter if you just want to find out the size of the buffer needed.

`inValueSize`
The number of bytes in the `outValuePtr` buffer. You may pass 0 if you just want to find out the size of the buffer needed.

`outActualSize`
The actual size of the `value` if this parameter is not NULL.

Return Value
Returns `kQTMetaDataInvalidMetaDataErr` if the metadata object or its reference is invalid, `kQTMetaDataInvalidItemErr` if the metatada item ID is invalid, or `noErr` if there is no error. See `Metadata Error Codes`.

Discussion
You can use this function to get the value of a metadata item that has a known item identifier.

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

Related Sample Code
QTMetaData

Declared In
`Movies.h`

QTMetaDataGetNextItem

Returns the next metadata item corresponding to a specified key.

```
OSStatus QTMetaDataGetNextItem (
   QTMetaDataRef inMetaData,
   QTMetaDataStorageFormat inMetaDataFormat,
   QTMetaDataItem inCurrentItem,
   QTMetaDataKeyFormat inKeyFormat,
   const UInt8 *inKeyPtr,
   ByteCount inKeySize,
   QTMetaDataItem *outNextItem
);
```

Parameters

inMetaData

The metadata object for this operation.

inMetaDataFormat

The metadata storage format used by the object passed in `inMetaData`. The format may be `UserData` storage, iTunes metadata storage, or QuickTime metadata storage. Not all objects will include all forms of storage, and other storage formats may appear in the future. Pass `kQTMetaDataStorageFormatWildcard` to target all storage formats.

inCurrentItem

The opaque, unique UInt64 identifier of the current metadata item to start the search. Your application obtains this item identifier from such functions as `QTMetaDataAddItem` (page 19).

inKeyFormat

The format of the key.

inKeyPtr

A pointer to the key of the item to be fetched next. You may pass NULL in this parameter if you are not interested in any specific key.

inKeySize

The size of the key in bytes.

outNextItem

The ID of the next metadata item after the item specified by `inCurrentItem` that has the specified key.

Return Value

Returns `kQTMetaDataInvalidMetaDataErr` if the metadata object or its reference is invalid, `kQTMetaDataInvalidItemErr` if the metatada item ID is invalid, `kQTMetaDataInvalidStorageFormatErr` if the metatada storage format is invalid, `kQTMetaDataInvalidKeyErr` if the key or its format is invalid, `kQTMetaDataNoMoreItemErr` if the last item has been fetched, or `noErr` if there is no error. See `Metadata Error Codes`.

Discussion

If the item designated by `inCurrentItem` is `kQTMetaDataItemUninitialized`, the function returns the first item with the specified key in the storage format. If it refers to a valid item in the storage format, the function will return the next item with the key after the item designated by `inCurrentItem`.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

QTMetaData

QTMetadataEditor

Declared In
`Movies.h`

QTMetaDataGetProperty

Returns a property of a metadata object.

```
OSStatus QTMetaDataGetProperty (
   QTMetaDataRef inMetaData,
   QTPropertyClass inPropClass,
   QTPropertyID inPropID,
   ByteCount inPropValueSize,
   QTPropertyValuePtr outPropValueAddress,
   ByteCount *outPropValueSizeUsed
);
```

Parameters

`inMetaData`
The metadata object for this operation.

`inPropClass`
The class of the property being asked about.

`inPropID`
The ID of the property being asked about.

`inPropValueSize`
`Size` of the buffer allocated to receive the property value.

`outPropValueAddress`
A pointer to the buffer allocated to receive the property value.

`outPropValueSizeUsed`
On return, the actual size of buffer space used.

Return Value
Returns `kQTMetaDataInvalidMetaDataErr` if the metadata object or its reference is invalid, `errPropNotSupported` if the metatada object does not support the property being asked about, `buffersTooSmall` if the allocated buffer is too small to hold the property, or `noErr` if there is no error. See `Metadata Error Codes`.

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

Declared In
`Movies.h`

QTMetaDataGetPropertyInfo

Returns information about a property of a metadata object.

```
OSStatus QTMetaDataGetPropertyInfo (
   QTMetaDataRef inMetaData,
   QTPropertyClass inPropClass,
   QTPropertyID inPropID,
   QTPropertyValueType *outPropType,
   ByteCount *outPropValueSize,
   UInt32 *outPropFlags
);
```

Parameters

inMetaData

The metadata object for this operation.

inPropClass

The class of the property being asked about.

inPropID

The ID of the property being asked about.

outPropType

A pointer to the type of the returned property's value.

outPropValueSize

A pointer to the size of the returned property's value.

outPropFlags

On return, a pointer to flags representing the requested information about the property.

Return Value

Returns `kQTMetaDataInvalidMetaDataErr` if the metadata object or its reference is invalid, `errPropNotSupported` if the metatada object does not support the property being asked about, or `noErr` if there is no error. See `Metadata Error Codes`.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`Movies.h`

QTMetaDataRelease

Decrements the retain count of a metadata object.

```
void QTMetaDataRelease (
   QTMetaDataRef inMetaData
);
```

Discussion

This function releases a metadata object by decrementing its reference count. When the count becomes 0 the memory allocated to the object is freed and the object is destroyed. If you retain a metadata object you are responsible for releasing it when you no longer need it.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

QTMetaData

QTMetadataEditor

Declared In
`Movies.h`

QTMetaDataRemoveItem

Removes a metadata item from a storage format.

```
OSStatus QTMetaDataRemoveItem (
   QTMetaDataRef inMetaData,
   QTMetaDataItem inItem
);
```

Parameters

`inMetaData`
: The metadata object for this operation.

`inItem`
: The opaque, unique UInt64 identifier of the metadata item for this operation. Your application obtains this item identifier from such functions as `QTMetaDataAddItem` (page 19) and `QTMetaDataGetNextItem` (page 23).

Return Value
Returns `kQTMetaDataInvalidMetaDataErr` if the metadata object or its reference is invalid, `kQTMetaDataInvalidItemErr` if the metatada item ID is invalid, or `noErr` if there is no error. See `Metadata Error Codes`.

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

Related Sample Code
QTMetadataEditor

Declared In
`Movies.h`

QTMetaDataRemoveItemsWithKey

Removes metadata items with a specific key from the storage format.

```
OSStatus QTMetaDataRemoveItemsWithKey (
   QTMetaDataRef inMetaData,
   QTMetaDataStorageFormat inMetaDataFormat,
   QTMetaDataKeyFormat inKeyFormat,
   const UInt8 *inKeyPtr,
   ByteCount inKeySize
);
```

Parameters

`inMetaData`
: The metadata object for this operation.

inMetaDataFormat
: The metadata storage format used by the object passed in `inMetaData`. The format may be `UserData` storage, iTunes metadata storage, or QuickTime metadata storage. Not all objects will include all forms of storage, and other storage formats may appear in the future. You can pass `kQTMetaDataStorageFormatWildcard` to target all storage formats.

inKeyFormat
: The format of the key.

inKeyPtr
: A pointer to the key of the item to be removed. You may pass NULL in this parameter if you want to remove all items.

inKeySize
: The size of the key in bytes.

Return Value
Returns `kQTMetaDataInvalidMetaDataErr` if the metadata object or its reference is invalid, `kQTMetaDataInvalidStorageFormatErr` if the metatada storage format is invalid, `kQTMetaDataInvalidKeyErr` if the key or its format is invalid, or `noErr` if there is no error. See `Metadata Error Codes`.

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

Declared In
`Movies.h`

QTMetaDataRetain

Increments the retain count of a metadata object.

```
QTMetaDataRef QTMetaDataRetain (
   QTMetaDataRef inMetaData
);
```

Parameters

inMetaData
: A metadata object that you want to retain.

Return Value
If successful, returns a metadata object that is the same as that passed in `inMetaData`.

Discussion
This function retains a metadata object by incrementing its reference count. You should retain every metadata object when you receive it from elsewhere and you want it to persist. If you retain a metadata object you are responsible for releasing it by calling `QTMetaDataRelease` (page 26).

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

Declared In
`Movies.h`

QTMetaDataSetItem

Sets the value of the metadata item from the item identifier.

```
OSStatus QTMetaDataSetItem (
   QTMetaDataRef inMetaData,
   QTMetaDataItem inItem,
   UInt8 *inValuePtr,
   ByteCount inValueSize,
   UInt32 inDataType
);
```

Parameters

inMetaData
: The metadata object for this operation.

inItem
: The opaque, unique UInt64 identifier of the metadata item for this operation. Your application obtains this item identifier from such functions as `QTMetaDataAddItem` (page 19) and `QTMetaDataGetNextItem` (page 23).

inValuePtr
: A pointer to the value to be set. This can be NULL if `inValueSize` is 0.

inValueSize
: The size of `inValuePtr` in bytes. Pass 0 if you want to set an item with no value.

inDataType
: A data type from the following list: `kQTMetaDataTypeBinary` = 0, kQTMetaDataTypeUTF8 = 1, kQTMetaDataTypeUTF16BE = 2, `kQTMetaDataTypeMacEncodedText` = 3, `kQTMetaDataTypeSignedIntegerBE` = 21, `kQTMetaDataTypeUnsignedIntegerBE` = 22, kQTMetaDataTypeFloat32BE = 23, kQTMetaDataTypeFloat64BE = 24With `kQTMetaDataTypeSignedIntegerBE` and `kQTMetaDataTypeUnsignedIntegerBE`, the size of the integer is determined by the value size.

Return Value

Returns `kQTMetaDataInvalidMetaDataErr` if the metadata object or its reference is invalid, `kQTMetaDataInvalidItemErr` if the metatada item ID is invalid, or `noErr` if there is no error. See `Metadata Error Codes`.

Discussion

You can use this function to set the value of the metadata item with a given item identifier. You can set an item with an empty value by passing 0 in `inValueSize`.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`Movies.h`

QTMetaDataSetItemProperty

Sets a property of a metadata item.

```
OSStatus QTMetaDataSetItemProperty (
   QTMetaDataRef inMetaData,
   QTMetaDataItem inItem,
   QTPropertyClass inPropClass,
   QTPropertyID inPropID,
   ByteCount inPropValueSize,
   ConstQTPropertyValuePtr inPropValueAddress
);
```

Parameters

inMetaData

The metadata object for this operation.

inItem

The opaque, unique UInt64 identifier of the metadata item for this operation. Your application obtains this item identifier from such functions as `QTMetaDataAddItem` (page 19) and `QTMetaDataGetNextItem` (page 23).

inPropClass

The class of the property being set.

inPropID

The ID of the property being set.

inPropValueSize

`Size` of the buffer containing the property value being set.

inPropValueAddress

A pointer to the buffer containing the item property value being set.

Return Value

Returns `kQTMetaDataInvalidMetaDataErr` if the metadata object or its reference is invalid, `kQTMetaDataInvalidItemErr` if the metatada item ID is invalid, `errPropNotSupported` if the metatada object does not support the property being set, `qtReadOnlyErr` if the property being set is read-only, or `noErr` if there is no error. See `Metadata Error Codes`.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`Movies.h`

QTMetaDataSetProperty

Sets a property of a metadata object.

```
OSStatus QTMetaDataSetProperty (
   QTMetaDataRef inMetaData,
   QTPropertyClass inPropClass,
   QTPropertyID inPropID,
   ByteCount inPropValueSize,
   ConstQTPropertyValuePtr inPropValueAddress
);
```

Parameters

inMetaData

The metadata object for this operation.

inPropClass

The class of the property being set.

inPropID

The ID of the property being set.

inPropValueSize

`Size` of the buffer containing the property value being set.

inPropValueAddress

A pointer to the buffer containing the property value being set.

Return Value

Returns `kQTMetaDataInvalidMetaDataErr` if the metadata object or its reference is invalid, `errPropNotSupported` if the metatada object does not support the property being set, `qtReadOnlyErr` if the property being set is read-only, or `noErr` if there is no error. See `Metadata Error Codes`.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`Movies.h`

QTRemoveMoviePropertyListener

Removes a movie property monitoring callback.

```
OSErr QTRemoveMoviePropertyListener (
   Movie inMovie,
   QTPropertyClass inPropClass,
   QTPropertyID inPropID,
   QTMoviePropertyListenerUPP inListenerProc,
   void *inUserData
);
```

Parameters

inMovie

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

inPropClass

A property class.

inPropID

A property ID.

inListenerProc

A Universal Procedure Pointer to a `QTMoviePropertyListenerProc` callback.

inUserData

User data to be passed to the callback.

Return Value

See `Error Codes` in the QuickTime API Reference. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 6.4.

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

Related Sample Code
QTAudioExtractionPanel

Declared In
`Movies.h`

QTRemoveTrackPropertyListener

Removes a track property monitoring callback

```
OSErr QTRemoveTrackPropertyListener (
   Track inTrack,
   QTPropertyClass inPropClass,
   QTPropertyID inPropID,
   QTTrackPropertyListenerUPP inListenerProc,
   void *inUserData
);
```

Parameters

`inTrack`
: The track for this operation.

`inPropClass`
: A property class.

`inPropID`
: A property ID.

`inListenerProc`
: A Universal Procedure Pointer to a `QTTrackPropertyListenerProc` callback.

`inUserData`
: User data to be passed to the callback.

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

Discussion
This routine removes a track property monitoring callback.

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

Related Sample Code
QTAudioExtractionPanel

Declared In
`Movies.h`

QTSampleTableAddSampleDescription

Adds a sample description to a sample table, returning a sample description ID that can be used to refer to it.

```
OSStatus QTSampleTableAddSampleDescription (
   QTMutableSampleTableRef sampleTable,
   SampleDescriptionHandle sampleDescriptionH,
   long mediaSampleDescriptionIndex,
   QTSampleDescriptionID *sampleDescriptionIDOut
);
```

Parameters

sampleTable

A reference to an opaque sample table object.

sampleDescriptionH

A handle to a `SampleDescription` structure. QuickTime will make its own copy of this handle.

mediaSampleDescriptionIndex

The sample description index of this sample description in a media. Pass 0 for sample descriptions you add to sample tables, to indicate that this was not retrieved from a media.

sampleDescriptionIDOut

A pointer to a variable to receive a sample description ID.

Return Value

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

Discussion

You can use the returned sample description ID when adding samples to the sample table.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`Movies.h`

QTSampleTableAddSampleReferences

Adds sample references to a sample table.

```
OSStatus QTSampleTableAddSampleReferences (
   QTMutableSampleTableRef sampleTable,
   SInt64 dataOffset,
   ByteCount dataSizePerSample,
   TimeValue64 decodeDurationPerSample,
   TimeValue64 displayOffset,
   SInt64 numberOfSamples,
   MediaSampleFlags sampleFlags,
   QTSampleDescriptionID sampleDescriptionID,
   SInt64 *newSampleNumOut
);
```

Parameters

sampleTable

A reference to an opaque sample table object.

dataOffset

A 64-bit signed integer that specifies the offset at which the first sample begins.

dataSizePerSample
: The number of bytes of data per sample. You must pass the data size per sample, not the total size of all the samples as with some other APIs.

decodeDurationPerSample
: A 64-bit time value that specifies the decode duration of each sample.

displayOffset
: A 64-bit time value that specifies the offset from decode time to display time of each sample. If the decode times and display times are the same, pass 0.

numberOfSamples
: A 64-bit signed integer, which must be greater than 0, that specifies the number of samples.

sampleFlags
: Flags that indicate the sync status of all samples: `mediaSampleNotSync` If set to 1, indicates that the sample to be added is not a sync sample. Set this flag to 0 if the sample is a sync sample. `mediaSampleShadowSync` If set to 1, the sample is a shadow sync sample. See these constants:

    `mediaSampleNotSync`

    `mediaSampleShadowSync`

sampleDescriptionID
: The ID of a sample description that has been added to the sample table with `QTSampleTableAddSampleDescription` (page 32).

newSampleNumOut
: A 64-bit signed integer that points to a variable to receive the sample number of the first sample that was added. Pass NULL if you don't want this information.

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

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

Declared In
`Movies.h`

QTSampleTableCopySampleDescription

Retrieves a sample description from a sample table.

```
OSStatus QTSampleTableCopySampleDescription (
   QTSampleTableRef sampleTable,
   QTSampleDescriptionID sampleDescriptionID,
   long *mediaSampleDescriptionIndexOut,
   SampleDescriptionHandle *sampleDescriptionHOut
);
```

Parameters

sampleTable
: A reference to an opaque sample table object.

sampleDescriptionID
: The sample description ID.

mediaSampleDescriptionIndexOut

A pointer to a variable to receive a media sample description index. If the sample description came from a media, this is the index that could be passed to `GetMediaSampleDescription` to retrieve the same sample description handle. The index will be 0 if the sample description did not come directly from a media. Pass NULL if you do not want to receive this information.

sampleDescriptionHOut

A pointer to a variable to receive a newly allocated sample description handle. Pass NULL if you do not want one. The caller is responsible for disposing the returned sample description handle using `DisposeHandle`.

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

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

Related Sample Code
MovieVideoChart

Declared In
`Movies.h`

QTSampleTableCreateMutable

Creates a new, empty sample table.

```
OSStatus QTSampleTableCreateMutable (
   CFAllocatorRef allocator,
   TimeScale timescale,
   void *hints,
   QTMutableSampleTableRef *newSampleTable
);
```

Parameters

allocator

The allocator to use for the new sample table.

timescale

A long integer that represents the timescale to use for durations and display offsets.

hints

Reserved; pass NULL.

newSampleTable

A pointer to a variable that receives a new reference to an opaque sample table object.

Return Value
An error code. Returns `memFullErr` if it could not allocate memory, `paramErr` if the time scale is not positive or `newSampleTable` is NULL, or `noErr` if there is no error.

Discussion
The newly created sample table contains no sample references. When sample references are added, their durations and display offsets are interpreted according to the sample table's current timescale.

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

Declared In
`Movies.h`

QTSampleTableCreateMutableCopy

Copies a sample table.

```
OSStatus QTSampleTableCreateMutableCopy (
   CFAllocatorRef allocator,
   QTSampleTableRef sampleTable,
   void *hints,
   QTMutableSampleTableRef *newSampleTable
);
```

Parameters

allocator
    The allocator to use for the new sample table.

sampleTable
    A reference to an opaque sample table object to copy.

hints
    Reserved; set to NULL.

newSampleTable
    A pointer to a variable that receives a reference to an opaque sample table object.

Return Value
An error code. Returns `memFullErr` if it could not allocate memory, `paramErr` if the time scale is not positive or `newSampleTable` is NULL, or `noErr` if there is no error.

Discussion
All the sample references and sample descriptions in the sample table are copied.

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

Declared In
`Movies.h`

QTSampleTableGetDataOffset

Returns the data offset of a sample.

```
SInt64 QTSampleTableGetDataOffset (
   QTSampleTableRef sampleTable,
   SInt64 sampleNum
);
```

Parameters

sampleTable
    A reference to an opaque sample table object.

sampleNum
    A 64-bit signed integer that represents a sample number. The first sample's number is 1.

Return Value
A 64-bit signed integer that represents the offset to the sample. Returns 0 if `sampleTable` is NULL or if the sample number is out of range.

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

Declared In
`Movies.h`

QTSampleTableGetDataSizePerSample

Returns the data size of a sample.

```
ByteCount QTSampleTableGetDataSizePerSample (
   QTSampleTableRef sampleTable,
   SInt64 sampleNum
);
```

Parameters

`sampleTable`
A reference to an opaque sample table object.

`sampleNum`
A 64-bit signed integer that represents the sample number. The first sample's number is 1.

Return Value
The size of the sample in bytes. Returns 0 if `sampleTable` is NULL or if the sample number is out of range.

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

Related Sample Code
MovieVideoChart

Declared In
`Movies.h`

QTSampleTableGetDecodeDuration

Returns the decode duration of a sample.

```
TimeValue64 QTSampleTableGetDecodeDuration (
   QTSampleTableRef sampleTable,
   SInt64 sampleNum
);
```

Parameters

`sampleTable`
A reference to an opaque sample table object.

`sampleNum`
A 64-bit signed integer that represents the sample number. The first sample's number is 1.

Return Value
A 64-bit time value that represents the decode duration of the sample. Returns 0 if `sampleTable` is NULL or if the sample number is out of range.

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

Related Sample Code
MovieVideoChart

Declared In
`Movies.h`

QTSampleTableGetDisplayOffset

Returns the offset from decode time to display time of a sample.

```
TimeValue64 QTSampleTableGetDisplayOffset (
   QTSampleTableRef sampleTable,
   SInt64 sampleNum
);
```

Parameters

`sampleTable`
A reference to an opaque sample table object.

`sampleNum`
A 64-bit signed integer that represents the sample number. The first sample's number is 1.

Return Value
A 64-bit time value that represents the offset from decode time to display time of the sample. Returns 0 if `sampleTable` is NULL or if the sample number is out of range.

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

Related Sample Code
MovieVideoChart

Declared In
`Movies.h`

QTSampleTableGetNextAttributeChange

Finds the next sample number at which one or more of a set of given sample attributes change.

```
OSStatus QTSampleTableGetNextAttributeChange (
   QTSampleTableRef sampleTable,
   SInt64 startSampleNum,
   QTSampleTableAttribute attributeMask,
   SInt64 *sampleNumOut
);
```

Parameters

sampleTable

A reference to an opaque sample table object.

startSampleNum

A 64-bit signed integer that contains the sample number to start searching from.

attributeMask

An unsigned 32-bit integer that contains flags indicating which kinds of attribute changes to search for: `kQTSampleTableAttribute_DiscontiguousData` = 1L << 0 Set this flag to find the first sample number `num` such that samples num-1 and `num` are not adjacent; that is, `dataOffset` of num-1 + `dataSize` of num-1 != `dataOffset` of `num`. `kQTSampleTableAttribute_DataSizePerSampleChange` = 1L << 1 Set this flag to find the first sample with data size per sample different from that of the starting sample. `kQTSampleTableAttribute_DecodeDurationChange` = 1L << 2 Set this flag to find the first sample with decode duration different from that of the starting sample. `kQTSampleTableAttribute_DisplayOffsetChange` = 1L << 3 Set this flag to find the first sample with display offset different from that of the starting sample. `kQTSampleTableAttribute_SampleDescriptionIDChange` = 1L << 4 Set this flag to find the first sample with sample description ID different from that of the starting sample. `kQTSampleTableAttribute_SampleFlagsChange` = 1L << 5 Set this flag to find the first sample with any media sample flags different from those of the starting sample. `kQTSampleTableAnyAttributeChange` = 0 If no flags are set, find the first sample with any attribute different from the starting sample. See these constants:

- `kQTSampleTableAttribute_DiscontiguousData`
- `kQTSampleTableAttribute_DataSizePerSampleChange`
- `kQTSampleTableAttribute_DecodeDurationChange`
- `kQTSampleTableAttribute_DisplayOffsetChange`
- `kQTSampleTableAttribute_SampleDescriptionIDChange`
- `kQTSampleTableAttribute_SampleFlagsChange`
- `kQTSampleTableAnyAttributeChange`

sampleNumOut

A 64-bit signed integer that points to a variable to receive the next sample number after `startSampleNum` at which any of the requested attributes change. If no attribute changes are found, this variable is set to 0.

Return Value

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

Availability

Available in Mac OS X v10.3 and later.

Declared In

`Movies.h`

QTSampleTableGetNumberOfSamples

Returns the number of samples in a sample table.

```
SInt64 QTSampleTableGetNumberOfSamples (
   QTSampleTableRef sampleTable
);
```

Parameters

sampleTable

A reference to an opaque sample table object.

Return Value

A 64-bit signed integer that contains the number of samples, or 0 if `sampleTable` is NULL.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

MovieVideoChart

Declared In

`Movies.h`

QTSampleTableGetProperty

Returns the value of a specific sample table property.

```
OSStatus QTSampleTableGetProperty (
   QTSampleTableRef sampleTable,
   QTPropertyClass inPropClass,
   QTPropertyID inPropID,
   ByteCount inPropValueSize,
   QTPropertyValuePtr outPropValueAddress,
   ByteCount *outPropValueSizeUsed
);
```

Parameters

sampleTable

A reference to an opaque sample table object.

inPropClass

Pass the following constant to define the property class: `kQTPropertyClass_SampleTable` = `'qtst'` Property of a sample table. See these constants:

`kQTPropertyClass_SampleTable`

inPropID

Pass one of these constants to define the property ID: `kQTSampleTablePropertyID_TotalDecodeDuration` = `'tded'` The total decode duration of all samples in the sample table. Read-only. `kQTSampleTablePropertyID_MinDisplayOffset` = '<ddd' The least display offset in the table. Negative offsets are less than positive offsets. Read-only. `kQTSampleTablePropertyID_MaxDisplayOffset` = '>ddd' The greatest display offset in the table. Positive offsets are greater than negative offsets. Read-only. `kQTSampleTablePropertyID_MinRelativeDisplayTime` = '<dis' The least display time of all samples in the table, relative to the decode time of the first sample in the table. Read-only. `kQTSampleTablePropertyID_MaxRelativeDisplayTime` = '>dis' The greatest display time of all samples in the table, relative to the decode time of the first sample in the table. Read-only. See these constants:

`kQTSampleTablePropertyID_TotalDecodeDuration`

`kQTSampleTablePropertyID_MinDisplayOffset`

`kQTSampleTablePropertyID_MaxDisplayOffset`

`kQTSampleTablePropertyID_MinRelativeDisplayTime`

`kQTSampleTablePropertyID_MaxRelativeDisplayTime`

inPropValueSize

The size of the buffer allocated to receive the property value.

outPropValueAddress

A pointer to the buffer allocated to receive the property value.

outPropValueSizeUsed

On return, the actual size of the property value written to the buffer.

Return Value

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

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

MovieVideoChart

Declared In

`Movies.h`

QTSampleTableGetPropertyInfo

Returns information about the properties of a sample table.

```
OSStatus QTSampleTableGetPropertyInfo (
   QTSampleTableRef sampleTable,
   QTPropertyClass inPropClass,
   QTPropertyID inPropID,
   QTPropertyValueType *outPropType,
   ByteCount *outPropValueSize,
   UInt32 *outPropertyFlags
);
```

Parameters

sampleTable

A reference to an opaque sample table object.

inPropClass

Pass the following constant to define the property class: `kQTPropertyClass_SampleTable` = `'qtst'` Property of a sample table. See these constants:

`kQTPropertyClass_SampleTable`

inPropID

Pass one of these constants to define the property ID: `kQTSampleTablePropertyID_TotalDecodeDuration` = `'tded'` The total decode duration of all samples in the sample table. Read-only. `kQTSampleTablePropertyID_MinDisplayOffset` = '<ddd' The least display offset in the table. Negative offsets are less than positive offsets. Read-only. `kQTSampleTablePropertyID_MaxDisplayOffset` = '>ddd' The greatest display offset in the table. Positive offsets are greater than negative offsets. Read-only. `kQTSampleTablePropertyID_MinRelativeDisplayTime` = '<dis' The least display time of all samples in the table, relative to the decode time of the first sample in the table. Read-only. `kQTSampleTablePropertyID_MaxRelativeDisplayTime` = '>dis' The greatest display time of all samples in the table, relative to the decode time of the first sample in the table. Read-only. See these constants:

`kQTSampleTablePropertyID_TotalDecodeDuration`

`kQTSampleTablePropertyID_MinDisplayOffset`

`kQTSampleTablePropertyID_MaxDisplayOffset`

`kQTSampleTablePropertyID_MinRelativeDisplayTime`

`kQTSampleTablePropertyID_MaxRelativeDisplayTime`

outPropType

A pointer to memory allocated to hold the `property` type on return: Pass NULL if you do not want this information.

outPropValueSize

A pointer to memory allocated to hold the size of the property value on return. Pass NULL if you do not want this information.

outPropertyFlags

A pointer to memory allocated to hold property flags on return. Pass NULL if you do not want this information.

Return Value

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

Availability

Available in Mac OS X v10.3 and later.

Declared In

`Movies.h`

QTSampleTableGetSampleDescriptionID

Returns the sample description ID of a sample.

```
QTSampleDescriptionID QTSampleTableGetSampleDescriptionID (
   QTSampleTableRef sampleTable,
   SInt64 sampleNum
);
```

Parameters

`sampleTable`

A reference to an opaque sample table object.

`sampleNum`

A 64-bit signed integer that represents the sample number. The first sample's number is 1.

Return Value
The sample's sample description ID. Returns 0 if `sampleTable` is NULL or if the sample number is out of range.

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

Related Sample Code
MovieVideoChart

Declared In
`Movies.h`

QTSampleTableGetSampleFlags

Returns the media sample flags of a sample.

```
MediaSampleFlags QTSampleTableGetSampleFlags (
   QTSampleTableRef sampleTable,
   SInt64 sampleNum
);
```

Parameters

`sampleTable`

A reference to an opaque sample table object.

`sampleNum`

A 64-bit signed integer that represents the sample number. The first sample's number is 1.

Return Value
A constant that describes characteristics of the sample (see below). Returns 0 if `sampleTable` is NULL or if the sample number is out of range.

Discussion
This function can return one or more of the following constants:

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

Related Sample Code
MovieVideoChart

Declared In
`Movies.h`

QTSampleTableGetTimeScale

Returns the timescale of a sample table.

```
TimeScale QTSampleTableGetTimeScale (
   QTSampleTableRef sampleTable
);
```

Parameters

`sampleTable`
    A reference to an opaque sample table object.

Return Value
A long integer that represents the sample's time scale, or 0 if `sampleTable` is NULL.

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

Declared In
`Movies.h`

QTSampleTableGetTypeID

Returns the CFTypeID value for the current sample table.

```
CFTypeID QTSampleTableGetTypeID (
   void
);
```

Return Value
A `CFTypeID` value.

Discussion
You could use this to test whether a `CFTypeRef` that was extracted from a CF container such as a `CFArray` is a `QTSampleTableRef`.

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

Declared In
`Movies.h`

QTSampleTableRelease

Decrements the retain count of a sample table.

```
void QTSampleTableRelease (
   QTSampleTableRef sampleTable
);
```

Parameters

sampleTable

A reference to an opaque sample table object. If you pass NULL in this parameter, nothing happens.

Discussion

If the retain count decreases to zero, the sample table is disposed.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

MovieVideoChart

Declared In

`Movies.h`

QTSampleTableReplaceRange

Replaces a range of samples in a sample table with a range of samples from another sample table.

```
OSStatus QTSampleTableReplaceRange (
   QTMutableSampleTableRef destSampleTable,
   SInt64 destStartingSampleNum,
   SInt64 destSampleCount,
   QTSampleTableRef sourceSampleTable,
   SInt64 sourceStartingSampleNum,
   SInt64 sourceSampleCount
);
```

Parameters

destSampleTable

A reference to an opaque sample table object to be modified.

destStartingSampleNum

A 64-bit signed integer that represents the first sample number in `destSampleTable` to be replaced or deleted, or the sample number at which samples should be inserted.

destSampleCount

A 64-bit signed integer that represents the number of samples to be removed from `destSampleTable`. Pass 0 to insert samples without removing samples.

sourceSampleTable

A reference to an opaque sample table object from which samples should be copied, or NULL to delete samples.

sourceStartingSampleNum

A 64-bit signed integer that represents the first sample number to be copied. This parameter is ignored when deleting samples.

sourceSampleCount

A 64-bit signed integer that represents the number of samples which should be copied. Pass 0 to delete samples.

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

Discussion
This function removes `destSampleCount` samples from `destSampleTable` starting with `destStartingSampleNum`, and then inserts `sourceSampleCount` samples from `sourceSampleTable` starting with `sourceStartingSampleNum` where the removed samples were. Sample descriptions will be copied if necessary and new sample description IDs defined. This function can also be used to delete a range of samples, or to insert samples without removing any.

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

Declared In
`Movies.h`

QTSampleTableRetain

Increments the retain count of a sample table.

```
QTSampleTableRef QTSampleTableRetain (
   QTSampleTableRef sampleTable
);
```

Parameters

sampleTable

A reference to an opaque sample table object. If you pass NULL in this parameter, nothing happens.

Return Value
A pointer to the `OpaqueQTSampleTable` structure that is returned for your convenience, or NULL if the function fails.

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

Declared In
`Movies.h`

QTSampleTableSetProperty

Sets the value of a specific sample table property.

```
OSStatus QTSampleTableSetProperty (
   QTSampleTableRef sampleTable,
   QTPropertyClass inPropClass,
   QTPropertyID inPropID,
   ByteCount inPropValueSize,
   ConstQTPropertyValuePtr inPropValueAddress
);
```

Parameters

sampleTable

A reference to an opaque sample table object.

inPropClass

Pass the following constant to define the property class: `kQTPropertyClass_SampleTable` = `'qtst'` Property of a sample table. See these constants:

`kQTPropertyClass_SampleTable`

inPropID

Pass one of these constants to define the property ID: `kQTSampleTablePropertyID_TotalDecodeDuration` = `'tded'` The total decode duration of all samples in the sample table. Read-only. `kQTSampleTablePropertyID_MinDisplayOffset` = '<ddd' The least display offset in the table. Negative offsets are less than positive offsets. Read-only. `kQTSampleTablePropertyID_MaxDisplayOffset` = '>ddd' The greatest display offset in the table. Positive offsets are greater than negative offsets. Read-only. `kQTSampleTablePropertyID_MinRelativeDisplayTime` = '<dis' The least display time of all samples in the table, relative to the decode time of the first sample in the table. Read-only. `kQTSampleTablePropertyID_MaxRelativeDisplayTime` = '>dis' The greatest display time of all samples in the table, relative to the decode time of the first sample in the table. Read-only. See these constants:

`kQTSampleTablePropertyID_TotalDecodeDuration`

`kQTSampleTablePropertyID_MinDisplayOffset`

`kQTSampleTablePropertyID_MaxDisplayOffset`

`kQTSampleTablePropertyID_MinRelativeDisplayTime`

`kQTSampleTablePropertyID_MaxRelativeDisplayTime`

inPropValueSize

Pass the size of the property value.

inPropValueAddress

Pass a `const` void pointer to the property value.

Return Value

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

Availability

Available in Mac OS X v10.3 and later.

Declared In

`Movies.h`

QTSampleTableSetTimeScale

Changes the timescale of a sample table.

```
OSStatus QTSampleTableSetTimeScale (
   QTMutableSampleTableRef sampleTable,
   TimeScale newTimeScale
);
```

Parameters

sampleTable

A reference to an opaque sample table object.

newTimeScale

A long integer whose value is the time scale to be set.

Return Value
An error code. Returns `paramErr` if the time scale is not positive or `sampleTable` is NULL, or `noErr` if there is no error.

Discussion
The durations and display offsets of all the sample references in the sample table are scaled from the old timescale to the new timescale. No durations are scaled to a value less than 1. Display offsets are adjusted to avoid display time collisions.

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

Declared In
`Movies.h`

QTSetMovieProperty

Sets the value of a specific movie property.

```
OSErr QTSetMovieProperty (
   Movie inMovie,
   QTPropertyClass inPropClass,
   QTPropertyID inPropID,
   ByteCount inPropValueSize,
   ConstQTPropertyValuePtr inPropValueAddress
);
```

Parameters

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

`inPropClass`
: A property class.

`inPropID`
: A property ID.

`inPropValueSize`
: The size of the property value.

`inPropValueAddress`
: A pointer to the the property value.

Return Value
See `Error Codes` in the QuickTime API Reference. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 6.4.

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

Related Sample Code
QTAudioExtractionPanel

Declared In
`Movies.h`

QTSetTrackProperty

Sets the value of a specific track property.

```
OSErr QTSetTrackProperty (
   Track inTrack,
   QTPropertyClass inPropClass,
   QTPropertyID inPropID,
   ByteCount inPropValueSize,
   ConstQTPropertyValuePtr inPropValueAddress
);
```

Parameters

inTrack
The track for this operation.

inPropClass
A property class.

inPropID
A property ID.

inPropValueSize
The size of the property value.

inPropValueAddress
A pointer to the the property value.

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

Discussion
This routine sets the value of a specific track property.

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

Related Sample Code
QTAudioExtractionPanel

Declared In
`Movies.h`

Callbacks

QTBandwidthNotificationProc

Undocumented

```
typedef OSErr (*QTBandwidthNotificationProcPtr) (long flags, void *reserved, void
 *refcon);
```

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

```
OSErr MyQTBandwidthNotificationProc (
    long    flags,
    void    *reserved,
    void    *refcon );
```

Parameters

`flags`

Undocumented

`reserved`

Reserved.

`refcon`

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

Return Value
See `Error Codes`. Your callback should return `noErr` if there is no error.

Declared In
`Movies.h`

Data Types

QTBandwidthNotificationUPP

Represents a type used by the Movie Properties API.

```
typedef STACK_UPP_TYPE(QTBandwidthNotificationProcPtr) QTBandwidthNotificationUPP;
```

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

Declared In
`Movies.h`

QTBandwidthReference

Represents a type used by the Movie Properties API.

```
typedef struct OpaqueQTBandwidthReference * QTBandwidthReference;
```

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

Declared In
`Movies.h`

QTScheduledBandwidthPtr

Represents a type used by the Movie Properties API.

```
typedef QTScheduledBandwidthRecord * QTScheduledBandwidthPtr;
```

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

Declared In
`Movies.h`

QTScheduledBandwidthRecord

Provides information to the QTScheduledBandwidthRequest function.

```
struct QTScheduledBandwidthRecord {
     long              recordSize;
     long              priority;
     long              dataRate;
     CompTimeValue     startTime;
     CompTimeValue     duration;
     TimeScale         scale;
     TimeBase          base;
 };
```

Fields
`recordSize`

Discussion
The number of bytes in this structure.

`priority`

Discussion
Undocumented

`dataRate`

Discussion
The data rate.

`startTime`

Discussion
The bandwidth usage start time.

`duration`

Discussion
`Duration` of bandwidth usage, or 0 if unknown.

`scale`

Discussion
The timescale of the `duration` field.

`base`

Discussion
The time base.

Declared In
`Movies.h`

QTScheduledBandwidthReference

Represents a type used by the Movie Properties API.

```
typedef struct OpaqueQTScheduledBandwidthReference * QTScheduledBandwidthReference;
```

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

Declared In
`Movies.h`

Constants

kQTPropertyClass_SampleTable

Constants grouped with kQTPropertyClass_SampleTable.

```
enum {
  /*
   * Property class for sample tables.
   */
  kQTPropertyClass_SampleTable  = 'qtst',
  /*
   * The total decode duration of all samples in the sample table.
   * Read-only.
   */
  kQTSampleTablePropertyID_TotalDecodeDuration = 'tded', /* TimeValue64, Read */
  /*
   * The least display offset in the table. (-50 is a lesser offset
   * than 20.)  Read-only.
   */
  kQTSampleTablePropertyID_MinDisplayOffset = '<ddd', /* TimeValue64, Read */
  /*
   * The greatest display offset in the table. (20 is a greater offset
   * than -50.)  Read-only.
   */
  kQTSampleTablePropertyID_MaxDisplayOffset = '>ddd', /* TimeValue64, Read */
  /*
   * The least display time of all samples in the table, relative to
   * the decode time of the first sample in the table.  Read-only.
   */
  kQTSampleTablePropertyID_MinRelativeDisplayTime = '<dis', /* TimeValue64, Read
*/
  /*
   * The greatest display time of all samples in the table, relative to
   * the decode time of the first sample in the table.  Read-only.
   */
  kQTSampleTablePropertyID_MaxRelativeDisplayTime = '>dis' /* TimeValue64, Read */
};
```

Declared In
`Movies.h`

QTSampleTableGetNextAttributeChange Values

Constants passed to QTSampleTableGetNextAttributeChange.

```
enum {
  /*
   * Set this flag to find first num such that samples num-1 and num
   * are not adjacent, ie, dataOffset of num-1 + dataSize of num-1 !=
   * dataOffset of num
   */
  kQTSampleTableAttribute_DiscontiguousData = 1L << 0,
  /*
   * Set this flag to find the first sample with data size per sample
   * different from that of the starting sample.
   */
  kQTSampleTableAttribute_DataSizePerSampleChange = 1L << 1,
  /*
   * Set this flag to find the first sample with decode duration
   * different from that of the starting sample.
   */
  kQTSampleTableAttribute_DecodeDurationChange = 1L << 2,
  /*
   * Set this flag to find the first sample with display offset
   * different from that of the starting sample.
   */
  kQTSampleTableAttribute_DisplayOffsetChange = 1L << 3,
  /*
   * Set this flag to find the first sample with sample description ID
   * different from that of the starting sample.
   */
  kQTSampleTableAttribute_SampleDescriptionIDChange = 1L << 4,
  /*
   * Set this flag to find the first sample with any media sample flags
   * different from those of the starting sample.
   */
  kQTSampleTableAttribute_SampleFlagsChange = 1L << 5,
  /*
   * If no flags are set, find the first sample with any attribute
   * different from the starting sample.
   */
  kQTSampleTableAnyAttributeChange = 0
};
```

Declared In
`Movies.h`

QTSampleTableGetSampleFlags Values

Constants passed to QTSampleTableGetSampleFlags.

```
enum {
  mediaSampleNotSync            = 1 << 0, /* sample is not a sync sample (eg. is
frame differenced */
  mediaSampleShadowSync         = 1 << 1, /* sample is a shadow sync */
  mediaSampleDroppable         = 1 << 27, /* sample is not required to be decoded
 for later samples to be decoded properly */
  mediaSamplePartialSync        = 1 << 16, /* sample is a partial sync (e.g., I
frame after open GOP) */
  mediaSampleHasRedundantCoding = 1 << 24, /* sample is known to contain redundant
 coding */
  mediaSampleHasNoRedundantCoding = 1 << 25, /* sample is known not to contain
redundant coding */
  mediaSampleIsDependedOnByOthers = 1 << 26, /* one or more other samples depend
upon the decode of this sample */
 mediaSampleIsNotDependedOnByOthers = 1 << 27, /* synonym for mediaSampleDroppable
 */
  mediaSampleDependsOnOthers    = 1 << 28, /* sample's decode depends upon decode
 of other samples */
  mediaSampleDoesNotDependOnOthers = 1 << 29, /* sample's decode does not depend
upon decode of other samples */
  mediaSampleEarlierDisplayTimesAllowed = 1 << 30 /* samples later in decode order
 may have earlier display times */
};
```

Constants

`mediaSampleNotSync`

Returned for frame-differenced video sample data.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

Declared In

`Movies.h`

Deprecated QuickTime Movie Properties Functions

A function identified as deprecated has been superseded and may become unsupported in the future.

Deprecated in Mac OS X v10.4

DisposeQTBandwidthNotificationUPP

Disposes of a QTBandwidthNotificationUPP pointer. (Deprecated in Mac OS X v10.4.)

```
void DisposeQTBandwidthNotificationUPP (
   QTBandwidthNotificationUPP userUPP
);
```

Parameters

userUPP

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

Return Value

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

Version Notes

Introduced in QuickTime 4.1.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.4.

Declared In

`Movies.h`

NewQTBandwidthNotificationUPP

Allocates a Universal Procedure Pointer for the QTBandwidthNotificationProc callback. (Deprecated in Mac OS X v10.4.)

```
QTBandwidthNotificationUPP NewQTBandwidthNotificationUPP (
   QTBandwidthNotificationProcPtr userRoutine
);
```

Parameters

userRoutine

A pointer to your application-defined function.

Return Value

A new UPP; see `Universal Procedure Pointers`.

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

Version Notes
Introduced in QuickTime 4.1. Replaces `NewQTBandwidthNotificationProc`.

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

Deprecated in Mac OS X v10.4.

Declared In
`Movies.h`

QTBandwidthRelease

Undocumented (Deprecated in Mac OS X v10.4.)

```
OSErr QTBandwidthRelease (
   QTBandwidthReference bwRef,
   long flags
);
```

Parameters

`bwRef`
 Undocumented

`flags`
 Undocumented

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

Version Notes
Introduced in QuickTime 4.

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

Deprecated in Mac OS X v10.4.

Declared In
`Movies.h`

QTBandwidthRequest

Undocumented (Deprecated in Mac OS X v10.4.)

```
OSErr QTBandwidthRequest (
   long priority,
   QTBandwidthNotificationUPP callback,
   const void *refcon,
   QTBandwidthReference *bwRef,
   long flags
);
```

Parameters

priority
: *Undocumented*

callback
: A `QTBandwidthNotificationProc` callback.

refcon
: A reference constant to be passed to your callback. Use this parameter to point to a data structure containing any information your function needs.

bwRef
: *Undocumented*

flags
: *Undocumented*

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

Version Notes
Introduced in QuickTime 4.

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

Deprecated in Mac OS X v10.4.

Declared In
`Movies.h`

QTBandwidthRequestForTimeBase

Undocumented (Deprecated in Mac OS X v10.4.)

```
OSErr QTBandwidthRequestForTimeBase (
   TimeBase tb,
   long priority,
   QTBandwidthNotificationUPP callback,
   const void *refcon,
   QTBandwidthReference *bwRef,
   long flags
);
```

Parameters

tb
: A time base. Your application obtains this time base identifier from `NewTimeBase`.

priority
: *Undocumented*

callback
: A `QTBandwidthNotificationProc` callback.

refcon
: A reference constant to be passed to your callback. Use this parameter to point to a data structure containing any information your function needs.

bwRef
: *Undocumented*

flags
: *Undocumented*

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

Version Notes
Introduced in QuickTime 4.1.

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

Deprecated in Mac OS X v10.4.

Declared In
`Movies.h`

QTScheduledBandwidthRelease

Undocumented (Deprecated in Mac OS X v10.4.)

```
OSErr QTScheduledBandwidthRelease (
   QTScheduledBandwidthReference sbwRef,
   long flags
);
```

Parameters

sbwRef
: A pointer to an opaque data structure.

flags
: *Undocumented*

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

Version Notes
Introduced in QuickTime 4.1.

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

Deprecated in Mac OS X v10.4.

Declared In
`Movies.h`

QTScheduledBandwidthRequest

Undocumented (Deprecated in Mac OS X v10.4.)

```
OSErr QTScheduledBandwidthRequest (
   QTScheduledBandwidthPtr scheduleRec,
   QTBandwidthNotificationUPP notificationCallback,
   void *refcon,
   QTScheduledBandwidthReference *sbwRef,
   long flags
);
```

Parameters

`scheduleRec`
A pointer to a `QTScheduledBandwidthRecord` structure.

`notificationCallback`
A Universal Procedure Pointer that accesses a `QTBandwidthNotificationProc` callback.

`refcon`
A reference constant to be passed to your callback. Use this parameter to point to a data structure containing any information your function needs.

`sbwRef`
A pointer to an opaque data structure.

`flags`
Undocumented

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

Version Notes
Introduced in QuickTime 4.1.

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

Deprecated in Mac OS X v10.4.

Declared In
`Movies.h`

Document Revision History

This table describes the changes to *QuickTime Movie Properties Reference*.

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

Index