QTSampleBuffer Class Reference

QuickTime > Cocoa

2009-04-08

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

Simultaneously published in the United States and Canada.

Contents

QTSampleBuffer Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/QTKit.framework
Availability	Available in QuickTime 7.2.1 and later.
Declared in	QTSampleBuffer.h
Related sample code	QTRecorder

Overview

This class provides format information, timing information, and metadata on media sample buffers. `QTSampleBuffer` objects contain data from media samples as well as metadata about those samples, including format information, timing information, and other attributes. Some extended information can be accessed via a QTSampleBuffer's `attributeForKey:` and `sampleBufferAttributes` methods, using the keys described in the Constants section. In addition to these explicit methods, applications can use key-value coding to get extended attributes. For an object that supports a given attribute, `valueForKey:` will be functionally identical to `attributeForKey:`. Applications wishing to observe changes for a given attribute can add a key-value observer where the key path is the attribute key.

Tasks

Getting Sample Buffer Information

- `attributeForKey:` (page 6)
  Returns a sample buffer attribute for the given key.
- `audioBufferListWithOptions:` (page 7)
  Returns a pointer to a Core Audio `AudioBufferList` containing audio data owned by the receiver.
- `bytesForAllSamples` (page 7)
  Returns a pointer to the bytes of media data contained in the sample buffer.
- `decodeTime` (page 8)
  Returns the decode time of the buffer.

Instance Methods

attributeForKey:

Returns a sample buffer attribute for the given key.

```
- (id)attributeForKey:(NSString *)key
```

Parameters

key

The key of the returned attribute. Attribute keys are described in the "Sample Buffer Attributes" (page 12) section.

Return Value

An object for the given attribute key, or `NIL` if the sample buffer does not have the given attribute.

Discussion

Use this method to get attributes of a sample buffer. The keys that can be used with this method are described in the Constants section. Applications using key-value coding can also get an attribute for a given key by passing that key to the NSObject `valueForKey:` method.

Availability

Mac OS X v10.5 and later.

Declared In
`QTSampleBuffer.h`

audioBufferListWithOptions:

Returns a pointer to a Core Audio `AudioBufferList` containing audio data owned by the receiver.

```
- (AudioBufferList
    *)audioBufferListWithOptions:(QTSampleBufferAudioBufferListOptions)options;
```

Parameters

`options`

A bitfield containing options that determine what kind of audio buffer list will be returned. The options constants, which can be combined using the bitwise or operator, are described as part of the `QTSampleBufferAudioBufferListOptions` type.

Return Value
A pointer to an `AudioBufferList` structure. This pointer and its associated audio buffers will remain valid as long as the receiver is valid and the value returned by `sampleUseCount` is greater then 0.

Discussion
This method returns a pointer to a Core Audio `AudioBufferList` containing all of the audio data in the sample buffer. The `AudioBufferList` can then be passed to Core Audio APIs for rendering and processing audio. The returned `AudioBufferList` will be valid for as long as the receiver is valid and the value returned by `sampleUseCount` has not been decremented to 0. Clients passing the `AudioBufferList` to an audio unit must include the `QTSampleBufferAudioBufferListOptionAssure16ByteAlignment` flag in the options parameter. This method will throw an NSInternalInconsistencyException if called after `decrementSampleUseCount` has been used to invalidate the media data contained in the sample buffer.

Availability
Mac OS X v10.5 and later.
Not available to 64-bit applications.

Declared In
`QTSampleBuffer.h`

bytesForAllSamples

Returns a pointer to the bytes of media data contained in the sample buffer.

```
- (void *)bytesForAllSamples
```

Return Value
A pointer to a buffer of media data.

Discussion
This method returns a pointer to the data for the media samples contained within the sample buffer. Clients reading bytes from this pointer should check the total length of the buffer using `lengthForAllSamples`. Applications can interpret the media data returned by this method using the infomation from the sample buffer's `formatDescription`. This method will throw an NSInternalInconsistencyException if called after `decrementSampleUseCount` has been used to invalidate the media data contained in the sample buffer.

Availability
Mac OS X v10.5 and later.
Not available to 64-bit applications.

Declared In
`QTSampleBuffer.h`

decodeTime

Returns the decode time of the buffer.

```
- (QTTime)decodeTime
```

Return Value
A `QTTime` representing the decode time of the buffer. For B-frame video media, the decode time may be different from the `presentationTime`.

Availability
Mac OS X v10.5 and later.

Declared In
`QTSampleBuffer.h`

decrementSampleUseCount

Decrements the use count of the sample data owned by the receiver, allowing the sample data to be invalidated after a matching call to `incrementSampleUseCount`.

```
- (void)decrementSampleUseCount
```

Discussion
This method allows clients to control when the potentially large memory buffers owned by the receiver are deallocated. A newly allocated `QTSampleBuffer` has a sample use count of 1. When the sample use count drops to 0, the memory allocated for the samples will be freed and the `bytesForAllSamples`, `lengthForAllSamples`, and`audioBufferListWithOptions`: methods will each throw an NSInternalInconsistencyException when called.

This method is analogous to the NSObject release method in that it allows clients to relinquish ownership over data contained within the sample buffer. In particular, clients that have called `incrementSampleUseCount` because they were interested in the sample data of `QTSampleBuffer` objects returned by other APIs in QTKit should call this method when they no longer need that data. It is particularly important that clients using garbage collection ensure that the sample use count is 0 when they no longer require the sample data owned by a `QTSampleBuffer`, so that memory can be deallocated promptly rather than when the object is finalized.

Availability
Mac OS X v10.5 and later.

Declared In
`QTSampleBuffer.h`

duration

Returns the duration of the buffer.

```
- (QTTime)duration
```

Return Value
A `QTTime` representing the duration of the buffer.

Availability
Mac OS X v10.5 and later.

Declared In
`QTSampleBuffer.h`

formatDescription

Returns the format description of the buffer.

```
- (QTFormatDescription *)formatDescription
```

Return Value
A `QTFormatDescription` object describing the media format of the buffer.

Availability
Mac OS X v10.5 and later.

Declared In
`QTSampleBuffer.h`

getAudioStreamPacketDescriptions:inRange:

Gets an array of Core Audio AudioStreamPacketDescriptions describing the lengths of samples in variable bit- rate audio buffers.

```
- (BOOL)getAudioStreamPacketDescriptions:(void *)audioStreamPacketDescriptions
    inRange:(NSRange)range
```

Parameters

audioStreamPacketDescriptions

An array of Core Audio AudioStreamPacketDescription structures allocated to be large enough to fit the number of packet descriptions indicated by range.

range

The range of packet descriptions to use when filling the array. If the range falls outside the number of samples returned by `numberOfSamples`, this method raises an NSRangeException.

Return Value
If the buffer contains variable bit-rate audio, this method fills the `audioStreamPacketDescriptions` with `AudioStreamPacketDescription` structures and returns `YES`. If the buffer contains single bit-rate audio, this method returns `NO` and leaves `audioStreamPacketDescriptions` untouched.

Discussion
Applications that need to process individual packets of variable bit-rate audio from the buffer should call this method to determine the length of each sample in the buffer. This method raises an NSInternalInconsistencyException if this method is invoked on a `QTSampleBuffer` object that does not describe an audio sample buffer.

Availability
Mac OS X v10.5 and later.

Declared In
`QTSampleBuffer.h`

incrementSampleUseCount

Increments the use count of the sample data owned by the receiver, preventing the sample data from being invalidated until a matching call to `decrementSampleUseCount`.

```
- (void)incrementSampleUseCount
```

Discussion
This method allows clients to control when the potentially large memory buffers owned by the receiver are deallocated. A newly allocated `QTSampleBuffer` has a sample use count of 1. When the sample use count drops to 0, the memory allocated for the samples will be freed and the `bytesForAllSamples`, `lengthForAllSamples`, and `audioBufferListWithOptions:` methods will each throw an NSInternalInconsistencyException when called.

This method is analogous to the `NSObject` retain method in that it allows clients to declare ownership over data contained within the sample buffer. In particular, clients interested in the sample data of `QTSampleBuffer` objects returned by other APIs in QTKit should call this method to ensure that they have acceess to the sample data, and later call `decrementSampleUseCount` when they no longer need that data. It is particularly important that clients using garbage collection ensure that the sample use count is 0 when they no longer require the sample data owned by a `QTSampleBuffer`, so that memory can be deallocated promptly rather than when the object is finalized.

Availability
Mac OS X v10.5 and later.

Declared In
`QTSampleBuffer.h`

lengthForAllSamples

Returns the length of the buffer returned by `bytesForAllSamples`.

```
- (NSUInteger)lengthForAllSamples
```

Return Value
The length, in bytes of the buffer returned by `bytesForAllSamples`.

Discussion
Clients reading bytes from the pointer returned by `bytesForAllSamples` should use this method to check the total length of the buffer. This method will throw an NSInternalInconsistencyException if called after `decrementSampleUseCount` has been used to invalidate the media data contained in the sample buffer.

Availability
Mac OS X v10.5 and later.
Not available to 64-bit applications.

Declared In
`QTSampleBuffer.h`

numberOfSamples

Returns the number of media samples contained in the buffer.

```
- (NSInteger)numberOfSamples
```

Return Value
The number of samples in the buffer.

Discussion
In general, video buffers will always contain one sample (a single frame), while audio buffers may contain multiple samples. Applications that need to interpret variable bit-rate audio can get the individual sample lengths with the `getAudioStreamPacketDescriptions:inRange:` method.

Availability
Mac OS X v10.5 and later.

Declared In
`QTSampleBuffer.h`

presentationTime

Returns the presentation time of the buffer.

```
- (QTTime)presentationTime
```

Return Value
A `QTTime` representing the presentation time of the buffer. For B-frame video media, the presentation time may be different from the `decodeTime`.

Availability
Mac OS X v10.5 and later.

Declared In
`QTSampleBuffer.h`

sampleBufferAttributes

Returns a dictionary of the sample buffer's current attirbutes.

```
- (NSDictionary *)sampleBufferAttributes
```

Return Value
A dictionary of attributes attached to the sample buffer. Attribute keys are described in the Constants section that discusses the attributes.

Discussion
Applications can use this method to determine what attributes a specific sample buffer supports.

Availability
Mac OS X v10.5 and later.

Declared In
`QTSampleBuffer.h`

sampleUseCount

Returns the use count of the sample data owned by the receiver.

```
- (NSUInteger)sampleUseCount
```

Return Value
The use count of the sample data owned by the receiver.

Discussion
This method returns the use count of the data owned by the reciever, as determined buy the number of invocations of `incrementSampleUseCount` and `decrementSampleUseCount`. If the value retunred by this method is 0, then the data owned by the reciever has been invalidated and the `bytesForAllSamples`, `lengthForAllSamples`, and `audioBufferListWithOptions:` methods wil throw an NSInternalInconsistencyException. Clients should rarely need to call this method. It is generally only useful for debugging purposes.

Availability
Mac OS X v10.5 and later.

Declared In
`QTSampleBuffer.h`

Constants

Sample Buffer Attributes

The following are constants for different sample buffer attributes.

```
NSString * const QTSampleBufferHostTimeAttribute;
NSString * const QTSampleBufferSMPTETimeAttribute
NSString * const QTSampleBufferSceneChangeTypeAttribute;
NSString * const QTSampleBufferDateRecordedAttribute;
NSString * const QTSampleBufferExplicitSceneChange;
NSString * const QTSampleBufferTimeStampDiscontinuitySceneChange;
```

Constants

`QTSampleBufferHostTimeAttribute`

Returns the buffer's host time, if the buffer is from a real time source.

The value returned by this attribute can be compared with the return value of `CVGetCurrentHostTime()` or `AudioGetCurrentHostTime()` to determine whether or not it is too late for the buffer to be processed in real time. Value is an `NSNumber` interpreted as a UInt64. This string value can be used in key paths for key-value coding, key-value observing, and bindings.

Available in Mac OS X v10.5 and later.

Declared in `QTSampleBuffer.h`.

`QTSampleBufferSMPTETimeAttribute`

Returns the SMPTE timecode of the sample buffer, if it has one.

The value is an `NSValue` interpreted as a `SMPTETime` (defined in `CoreAudio/CoreAudioTypes.h`). This string value can be used in key paths for key-value coding, key-value observing, and bindings.

Available in Mac OS X v10.5 and later.

Declared in `QTSampleBuffer.h`.

`QTSampleBufferSceneChangeTypeAttribute`

If the buffer marks a scene change in the input content, returns a constant.

The returned constant specifies the type of scene change. This string value can be used in key paths for key-value coding, key-value observing, and bindings.

Available in Mac OS X v10.5 and later.

Declared in `QTSampleBuffer.h`.

`QTSampleBufferDateRecordedAttribute`

Returns the date on which the media in the buffer was originally recorded.

The value is an `NSDate`. This string value can be used in key paths for key-value coding, key-value observing, and bindings.

Available in Mac OS X v10.5 and later.

Declared in `QTSampleBuffer.h`.

`QTSampleBufferExplicitSceneChange`

Indicates that a scene change was explicitly marked in the sample buffer's metadata.

This constant is returned by `QTSampleBufferSceneChangeTypeAttribute` specifying what kind of scene change, if any, is marked by a sample buffer.

Available in Mac OS X v10.5 and later.

Declared in `QTSampleBuffer.h`.

`QTSampleBufferTimeStampDiscontinuitySceneChange`

Indicates that the scene changed due to a discontinuity in time stamps between the current sample buffer and the previous sample buffer.

This constant is returned by `QTSampleBufferSceneChangeTypeAttribute` specifying what kind of scene chnage, if any, is marked by a sample buffer.

Available in Mac OS X v10.5 and later.

Declared in `QTSampleBuffer.h`.

Document Revision History

This table describes the changes to *QTSampleBuffer Class Reference*.

Date	Notes
2009-04-08	Fixed constant listings. Updated links. Minor fixes.
2007-07-23	New document that describes the Objective-C API for supporting and working with QuickTime Capture.

Document Revision History

Index