
QTCaptureDecompressedVideoOutput Class Reference

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





Apple Inc.
© 2009 Apple Inc.
All rights reserved.

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

The Apple logo is a trademark of Apple Inc.

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

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

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

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

.Mac is a registered service mark of Apple Inc.

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

Simultaneously published in the United States and Canada.

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

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

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

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

Contents

QTCaptureDecompressedVideoOutput Class Reference 5

Overview 5

Tasks 5

 Decompressing Video Output 5

Instance Methods 6

 delegate 6

 outputVideoFrame:withSampleBuffer:fromConnection: 6

 pixelBufferAttributes 7

 setDelegate: 8

 setMinimumVideoFrameInterval 8

 setPixelBufferAttributes: 8

Delegate Methods 9

 captureOutput:didOutputVideoFrame:withSampleBuffer:fromConnection: 9

Document Revision History 11

Index 13

QTCaptureDecompressedVideoOutput Class Reference

Inherits from	QTCaptureOutput : NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/QTKit.framework
Availability	Available in QuickTime 7.2.1 and later.
Declared in	QTCaptureDecompressedVideoOutput.h

Overview

This class represents an output destination for a `QTCaptureSession` object that can be used to process decompressed frames from the video being captured. Instances of `QTCaptureDecompressedVideoOutput` produce decompressed video frames suitable for high-quality processing. Because instances maintain maximum frame quality and avoid dropping frames, using this output may result in reduced performance while capturing. Applications that need to process decompressed frames but can tolerate dropped frames or drops in decompression quality should use `QTCaptureVideoPreviewOutput` instead. Applications can access the decompressed frames via the `captureOutput:didOutputVideoFrame:withSampleBuffer:fromConnection:` delegate method. Clients can also create subclasses of `QTCaptureDecompressedVideoOutput` to add custom capturing behavior.

Tasks

Decompressing Video Output

- [delegate](#) (page 6)
Returns the receiver's delegate.
- [setDelegate:](#) (page 8)
Sets the receiver's delegate.
- [setMinimumVideoFrameInterval](#) (page 8)
Sets the minimum time interval between which the receiver should output consecutive video frames.
- [outputVideoFrame:withSampleBuffer:fromConnection:](#) (page 6)
Called whenever the receiver outputs a new video frame.

- [pixelBufferAttributes](#) (page 7)
Returns the Core Video pixel buffer attributes previously set by `setPixelBufferAttributes:` that determine what kind of pixel buffers are output by the receiver.
- [setPixelBufferAttributes:](#) (page 8)
Sets the Core Video pixel buffer attributes that determine what kind of pixel buffers are output by the receiver.
- [captureOutput:didOutputVideoFrame:withSampleBuffer:fromConnection:](#) (page 9) *delegate method*
Called whenever the video preview output outputs a new video frame.

Instance Methods

delegate

Returns the receiver's delegate.

- (id)delegate

Availability

Mac OS X v10.5 and later.

Declared In

QTCaptureDecompressedVideoOutput.h

outputVideoFrame:withSampleBuffer:fromConnection:

Called whenever the receiver outputs a new video frame.

- (void)outputVideoFrame:(CVImageBufferRef)videoFrame
withSampleBuffer:(QTSampleBuffer *)sampleBuffer
fromConnection:(QTCaptureConnection *)connection

Parameters

videoFrame

A Core Video buffer containing the decompressed frame.

sampleBuffer

A sample buffer containing additional information about the frame, such as its presentation time.

connection

The connection from which the video was received.

Discussion

This method should not be invoked directly. Subclasses can override this method to provide custom processing behavior for each frame. The default implementation calls the delegate's `captureOutput:didOutputVideoFrame:withSampleBuffer:fromConnection:` method.



Warning: Subclasses should not assume that this method will be called on the main thread. In addition, this method is called periodically, so it must be efficient to prevent capture performance problems.

Special Considerations

In order to promptly reclaim memory resources, after this method returns, the sample data contained within the `QTSampleBuffer` object will be released using its `decrementSampleUseCount` method. Clients that reference the sample buffer and are interested in the sample data that it contains after this method returns should call `incrementSampleUseCount` on the sample buffer within this method to ensure that the data remains valid until they no longer need it (at which time they should call `decrementSampleUseCount`). Clients that reference the sample buffer after this method returns, but only need access to its metadata, such as duration, presentation time, and other attributes, need not call `incrementSampleUseCount`. Note that to maintain optimal performance, some sample buffers directly reference pools of memory that may need to be reused by the device system and other capture inputs. This is frequently the case for uncompressed device native capture where memory blocks are copied as little as possible. If multiple sample buffers reference such pools of memory for too long, inputs will no longer be able to copy new samples into memory and those samples will be dropped. If your application is causing samples to be dropped by holding on to sample data for too long using `incrementSampleUseCount`, but it needs access to the sample data for a long period of time, consider copying the data into a new buffer and then calling `decrementSampleUseCount` on the sample buffer so that the memory it references can be reused.

Availability

Mac OS X v10.5 and later.

Not available to 64-bit applications.

Declared In

`QTCaptureDecompressedVideoOutput.h`

pixelBufferAttributes

Returns the Core Video pixel buffer attributes previously set by `setPixelBufferAttributes:` that determine what kind of pixel buffers are output by the receiver.

- (NSDictionary *)pixelBufferAttributes

Return Value

A dictionary containing pixel buffer attributes for buffers output by the receiver. The keys in the dictionary are described in `CoreVideo/CVPixelBuffer.h`. If the return value is `NIL`, then the receiver outputs buffers using the fastest possible pixel buffer attributes.

Discussion

This method returns the pixel buffer attributes set by `setPixelBufferAttributes:` that clients can use to customize the size and pixel format of the video frames output by the receiver. When the dictionary is non-nil, the receiver will attempt to output pixel buffers using the attributes specified in the dictionary. A non-nil dictionary also guarantees that the output `CVImageBuffer` is a `CVPixelBuffer`. When the value for `kCVPixelBufferPixelFormatTypeKey` is set to an `NSNumber`, all image buffers output by the receiver will be in that format. When the value is an `NSArray`, image buffers output by the receiver will be in the most optimal format specified in that array. If the captured images are not in the one of the specified pixel formats, then a format conversion will be performed. If the dictionary is `NIL` or there is no value for the `kCVPixelBufferPixelFormatTypeKey`, then the receiver will output images in the most efficient possible format given the input. For example, if the source is an `iSight` producing component Y'CbCr 8-bit 4:2:2 video then Y'CbCr 8-bit 4:2:2 will be used as the output format in order to avoid any conversions. The default value for the returned dictionary is `NIL`.

Availability

Available in Mac OS X v10.5 and later.

Declared In

QTCaptureDecompressedVideoOutput.h

setDelegate:

Sets the receiver's delegate.

- (void)setDelegate:(id)delegate

Availability

Mac OS X v10.5 and later.

Declared In

QTCaptureDecompressedVideoOutput.h

setMinimumVideoFrameInterval

Sets the minimum time interval between which the receiver should output consecutive video frames.

- (void)setMinimumVideoFrameInterval:(NSTimeInterval)minimumVideoFrameInterval

Parameters*minimumVideoFrameInterval*An `NSTimeInterval` specifying the minimum interval between video frames. A value of 0 indicates that there should be no frame rate limit.**Discussion**

This method sets the minimum amount of time that should separate consecutive frames output by the receiver. This is equivalent to the inverse of the maximum frame rate. A value of 0 indicates an unlimited maximum frame rate. The default value is 0.

Availability

Mac OS X v10.5 and later. QuickTime 7.6.1.

setPixelBufferAttributes:

Sets the CoreVideo pixel buffer attributes that determine what kind of pixel buffers are output by the receiver.

- (void)setPixelBufferAttributes:(NSDictionary *)pixelBufferAttributes

Parameters*pixelBufferAttributes*

A dictionary containing pixel buffer attributes for buffers that will be output by the receiver. The keys in the dictionary are described in `CoreVideo/CVPixelBuffer.h`. If the dictionary is `NIL`, then the receiver outputs buffers using the fastest possible pixel buffer attributes.

Discussion

This method sets the pixel buffer attributes that clients can use to customize the size and pixel format of the video frames output by the receiver. When the dictionary is non-nil, the receiver will attempt to output pixel buffers using the attributes specified in the dictionary. A non-nil dictionary also guarantees that the output `CVImageBuffer` is a `CVPixelBuffer`. When the value for `kCVPixelBufferPixelFormatTypeKey` is set to an `NSNumber`, all image buffers output by the receiver will be in that format. When the value is an `NSArray`,

image buffers output by the receiver will be in the most optimal format specified in that array. If the captured images are not in the one of the specified pixel formats, then a format conversion will be performed. If the dictionary is `NIL` or there is no value for the `kCVPixelBufferPixelFormatTypeKey`, then the receiver will output images in the most efficient possible format given the input. For example, if the source is an iSight producing component Y'CbCr 8-bit 4:2:2 video then Y'CbCr 8-bit 4:2:2 will be used as the output format in order to avoid any conversions.

Availability

Available in Mac OS X v10.5 and later.

Declared In

QTCaptureDecompressedVideoOutput.h

Delegate Methods

captureOutput:didOutputVideoFrame:withSampleBuffer:fromConnection:

Called whenever the video preview output outputs a new video frame.

```
- (void)captureOutput:(QTCaptureOutput *)captureOutput
  didOutputVideoFrame:(CVImageBufferRef)videoFrame
  withSampleBuffer:(QTSampleBuffer *)sampleBuffer
  fromConnection:(QTCaptureConnection *)connection
```

Parameters

captureOutput

The `QTCaptureDecompressedVideoOutput` instance that output the frame.

videoFrame

A Core Video image buffer containing the decompressed frame.

sampleBuffer


A sample buffer containing additional information about the frame, such as its presentation time.

connection

The connection from which the video was received.

Discussion

Delegates receive this message whenever the output decompresses and outputs a new video frame. Delegates can use the provided video frame for a custom preview or for further image processing.

 **Warning:** Delegates should not assume that this method will be called on the main thread. In addition, this method is called periodically, so it must be efficient to prevent capture performance problems.

Special Considerations

In order to promptly reclaim memory resources, after this method returns, the sample data contained within the `QTSampleBuffer` object will be released using its `decrementSampleUseCount` method. Clients that reference the sample buffer and are interested in the sample data that it contains after this method returns should call `incrementSampleUseCount` on the sample buffer within this method to ensure that the data remains valid until they no longer need it (at which time they should call `decrementSampleUseCount`). Clients that reference the sample buffer after this method returns, but only need access to its metadata, such as duration, presentation time, and other attributes, need not call `incrementSampleUseCount`. Note that

to maintain optimal performance, some sample buffers directly reference pools of memory that may need to be reused by the device system and other capture inputs. This is frequently the case for uncompressed device native capture where memory blocks are copied as little as possible. If multiple sample buffers reference such pools of memory for too long, inputs will no longer be able to copy new samples into memory and those samples will be dropped. If your application is causing samples to be dropped by holding on to sample data for too long using `incrementSampleUseCount`, but it needs access to the sample data for a long period of time, consider copying the data into a new buffer and then calling `decrementSampleUseCount` on the sample buffer so that the memory it references can be reused.

Availability

Mac OS X v10.5 and later.

Declared In

`QTCaptureDecompressedVideoOutput.h`

Document Revision History

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

Date	Notes
2009-03-04	Added four new method descriptions; updated availability information.
2007-07-19	New document that describes the Objective-C API for supporting and working with QuickTime Capture.

REVISION HISTORY

Document Revision History

Index

C

captureOutput:didOutputVideoFrame:
withSampleBuffer:fromConnection:
<NSObject> delegate method [9](#)

D

delegate [instance method 6](#)

O

outputVideoFrame:withSampleBuffer:fromConnection:
[instance method 6](#)

P

pixelBufferAttributes [instance method 7](#)

S

setDelegate: [instance method 8](#)
setMinimumVideoFrameInterval [instance method 8](#)
setPixelBufferAttributes: [instance method 8](#)