
QTCaptureFileOutput Reference

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



2007-07-23



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

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

The Apple logo is a trademark of Apple Inc.

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

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

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

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

.Mac is a registered service mark of Apple Inc.

Apple, the Apple logo, Cocoa, eMac, 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

QTCaptureFileOutput Reference 5

Overview	5
Tasks	5
Recording File Outputs	5
Methods Implemented by the Delegate	6
Instance Methods	7
captureOutput:didFinishRecordingToOutputFileAtURL:forConnections:dueToError:	7
captureOutput:didOutputSampleBuffer:fromConnection:	7
captureOutput:didStartRecordingToOutputFileURL:forConnections:	8
captureOutput:mustChangeOutputFileAtURL:forConnections:dueToError:	8
captureOutput:shouldChangeOutputFileAtURL:forConnections:	9
captureOutput:willFinishRecordingToOutputFileAtURL:forConnections:dueToError:	10
captureOutput:willStartRecordingToOutputFileURL:forConnections:	11
compressionOptionsForConnection:	11
delegate	12
maximumRecordedDuration	12
maximumRecordedFileSize	12
outputFileURL	13
recordedDuration	13
recordedFileSize	13
recordToOutputFileURL:	14
recordToOutputFileURL:bufferDestination:	14
setCompressionOptions:forConnection:	15
setDelegate:	15
setMaximumRecordedDuration:	16
setMaximumRecordedFileSize:	16
Constants	16
QTCaptureFileOutputBufferDestination	16

Document Revision History 19

Index 21

QTCaptureFileOutput Reference

Inherits from	QTCaptureOutput : NSObject
Conforms to	NSObject (NSObject)
Framework	Library/Frameworks/QTKit.framework
Availability	Available in QuickTime 7.2.1 and later.
Declared in	QTCaptureFileOutput.h
Related sample code	QT Capture Widget QTRecorder

Overview

This is an abstract superclass output destination for `QTCaptureSession` that writes captured media to files. This superclass defines the interface for outputs that record media samples to files. File outputs are designated a recording output file using the `recordToFileURL:` and `recordToFileURL:bufferDestination:` methods. On successive invocations of these methods, the output file can be changed dynamically without losing media samples. A file output can also be set to not record incoming frames (the default behavior when an output is first initialized) by passing `NIL` as the output file URL. Because files are recorded in the background, applications will generally need to set a delegate for a file output so that they can be notified when recording files are started and finished. The file output delegate can also be used to control recording for exact media samples by implementing the `captureOutput:didOutputSampleBuffer:fromConnection:` method. Currently, the only concrete subclass of this class is `QTCaptureMovieFileOutput`.

Tasks

Recording File Outputs

- [outputFileURL](#) (page 13)
Returns the file written to by the receiver.
- [recordToOutputFileURL:](#) (page 14)
Sets the file written to by the receiver.
- [recordToOutputFileURL:bufferDestination:](#) (page 14)
Sets the file written to by the receiver, specifying where the sample buffer currently in flight should be recorded.

- [recordedDuration](#) (page 13)
Returns the duration of the media recorded by the receiver.
- [recordedFileSize](#) (page 13)
Returns the size, in bytes, of the data recorded by the receiver to output files.
- [maximumRecordedDuration](#) (page 12)
Returns the maximum duration of the media that should be recorded by the receiver.
- [setMaximumRecordedDuration:](#) (page 16)
Sets the maximum duration of the media that should be recorded by the receiver.
- [maximumRecordedFileSize](#) (page 12)
Returns the maximum file size, in bytes, of the file that should be recorded by the receiver.
- [setMaximumRecordedFileSize:](#) (page 16)
Sets the maximum file size, in bytes, of the file that should be recorded by the receiver.
- [compressionOptionsForConnection:](#) (page 11)
Returns the options the receiver uses to compress media on the given connection as it is being captured.
- [setCompressionOptions:forConnection:](#) (page 15)
Sets the options the receiver uses to compress media on the given connection as it is being captured.
- [delegate](#) (page 12)
Returns the receiver's delegate.
- [setDelegate:](#) (page 15)
Sets the receiver's delegate.

Methods Implemented by the Delegate

- [captureOutput:didOutputSampleBuffer:fromConnection:](#) (page 7)
Gives the delegate the opportunity to inspect samples as they are received by the output and start and stop capturing at exact times.
- [captureOutput:willStartRecordingToOutputFileURL:forConnections:](#) (page 11)
Informs the delegate when the output is about to start writing to a file.
- [captureOutput:didStartRecordingToOutputFileURL:forConnections:](#) (page 8)
Informs the delegate when the output has started writing to a file.
- [captureOutput:shouldChangeOutputFileAtURL:forConnections:](#) (page 9)
Gives the delegate the opportunity to determine what should happen when an output file has reached a soft limit.
- [captureOutput:mustChangeOutputFileAtURL:forConnections:dueToError:](#) (page 8)
Informs the delegate when an output file can no longer be written using the incoming media.
- [captureOutput:willFinishRecordingToOutputFileAtURL:forConnections:dueToError:](#) (page 10)
Informs the delegate when the output will stop writing new samples to a file.
- [captureOutput:didFinishRecordingToOutputFileAtURL:forConnections:dueToError:](#) (page 7)
Informs the delegate when an output file is ready to be opened by applications.

Instance Methods

captureOutput:didFinishRecordingToOutputFileAtURL:forConnections:dueToError:

Informs the delegate when an output file is ready to be opened by applications.

```
- (void)captureOutput:(QTCaptureFileOutput *)captureOutput
  didFinishRecordingToOutputFileAtURL:(NSURL *)outputFileURL
  forConnections:(NSArray *)connections
  dueToError:(NSError *)error
```

Parameters

captureOutput

The capture file output that has finished writing the file.

outputURL

The file URL of the file that has been written.

connections

An array of QTCaptureConnection objects owned by the receiver that provided the data that was written to the file.

error

An error describing what caused the file to stop recording, or NIL if there was no error.

Discussion

Whenever the receiver's `recordToOutputFileURL:` or `recordToOutputFileURL:bufferDestination:` method is called during recording, they return immediately, finishing any pending file writing in the background. Delegates must implement this method to be informed when those files are finished and ready to be opened by applications.



Warning: Applications should not assume that this method will be called on the main thread.

Availability

Mac OS X v10.5 and later.

captureOutput:didOutputSampleBuffer:fromConnection:

Gives the delegate the opportunity to inspect samples as they are received by the output and start and stop capturing at exact times.

```
- (void)captureOutput:(QTCaptureFileOutput *)captureOutput
  didOutputSampleBuffer:(QTSampleBuffer *)sampleBuffer
  fromConnection:(QTCaptureConnection *)connection
```

Parameters

captureOutput

The capture file output that is receiving the media data.

sampleBuffer

A sample buffer object containing the sample data and additional information about the sample, such as its time code and record date.

connection

The capture connection object owned by the receiver that is receiving the sample data.

Discussion

This method is called whenever the file output receives a single media sample (a single video frame, for example) through the given connection. This gives delegates an opportunity to start and stop capturing or change output files at an exact sample. Calls to the file output's `recordToOutputFileURL:` and `recordToOutputFileURL:bufferDestination:` methods are guaranteed to include the received sample if called from within this method. Delegates can gather information particular to the sample, such as its record time, and whether it marks a scene change, by inspecting the `sampleInfo` object. Sample buffers always contain a single frame of video if called from this method but may also contain multiple packets of audio. For B-frame video formats, this method is always called in presentation order.



Warning: Applications 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.

Availability

Mac OS X v10.5 and later.

captureOutput:didStartRecordingToOutputFileURL:forConnections:

Informs the delegate when the output has started writing to a file.

```
(void)captureOutput:(QTCaptureFileOutput *)captureOutput
    didStartRecordingToOutputFileURL:(NSURL *)fileURL
    forConnections:(NSArray *)connections
```

Parameters

captureOutput

The capture file output that started writing the file.

outputURL

The file URL of the file being written.

connections

An array of `QTCaptureConnection` objects owned by the receiver that provided the data that is being written to the file.

Discussion

Applications should not assume that this method will be called on the main thread.

Availability

Mac OS X v10.5 and later.

captureOutput:mustChangeOutputFileAtURL:forConnections:dueToError:

Informs the delegate when an output file can no longer be written using the incoming media.

```
(void)captureOutput:(QTCaptureFileOutput *)captureOutput
    mustChangeOutputFileAtURL:(NSURL *)outputFileURL
    forConnections:(NSArray *)connections
    dueToError:(NSError *)error
```


Parameters*captureOutput*

The capture file output that must finish writing the file.

outputURL

The file URL of the file that is being written.

*connections*An array of `QTCaptureConnection` objects owned by the receiver that provided the data that is being written to the file.*error*

The error that caused the output to require that a new file be written.

Discussion

This method is called if the existing output file for that connection can no longer be written (this occurs, for example, if the stream format of the samples has changed, the output is receiving invalid samples, or there is insufficient disk space remaining on the output file's disk). Delegates implementing this method can start recording on a new file using `recordToOutputFileURL:` or `recordToOutputFileURL:bufferDestination:` to ensure that incoming data will continue to be recorded. If the delegate does not implement this method or does not set new output files for the given connections, recording stops automatically.



Warning: Applications should not assume that this method will be called on the main thread.

Availability

Mac OS X v10.5 and later.

captureOutput:shouldChangeOutputFileAtURL:forConnections:

Gives the delegate the opportunity to determine what should happen when an output file has reached a soft limit.

```
- (BOOL)captureOutput:(QTCaptureFileOutput *)captureOutput
  shouldChangeOutputFileAtURL:(NSURL *)outputFileURL
  forConnections:(NSArray *)connections
  dueToError:(NSError *)error
```

Parameters*captureOutput*

The capture file output that should finish writing the file.

outputURL

The file URL of the file that is being written.

*connections*An array of `QTCaptureConnection` objects owned by the receiver that provided the data that is being written to the file.*error*

The error that caused the output to suggest that a new file be written.

Return Value

Delegates should return YES if the current file should no longer be written, or NO if the current file should continue to be written.

Discussion

This method is called when the file output encounters a problem, such as dropped media samples (indicated by a `QTErrormediaDiscontinuity` error), that doesn't require that recording stop but may be a reason for some applications to change files or stop recording. For example, applications concerned with recording every frame of video or every sample of audio may want to treat such problems as error conditions rather than ignoring them. This method is also called when the file output reaches a soft limit, namely one of the limits set using the `setMaximumRecordedDuration:` and `setMaximumRecordedFileSize:` methods. Delegates should check the value of the error parameter to see what kind of error caused this delegate method to be called. If the delegate returns `NO`, the output will continue writing the same file. If the delegate returns `YES` and doesn't set a new output file, `captureOutput:mustChangeOutputFileAtURL:forConnections:dueToError:` will be called. If the delegate returns `YES` and sets a new output file, recording will continue on the new file. If the delegate does not respond to this method, the file output will automatically continue recording when it encounters one of these errors, unless it is a `QTErrormaximumDurationReached` or `QTErrormaximumFileSizeReached` error, in which case the file output will automatically stop recording.



Warning: Applications should not assume that this method will be called on the main thread.

Availability

Mac OS X v10.5 and later.

captureOutput:willFinishRecordingToOutputFileAtURL:forConnections:dueToError:

Informs the delegate when the output will stop writing new samples to a file.

```
- (void)captureOutput:(QTCaptureFileOutput *)captureOutput
  willFinishRecordingToOutputFileAtURL:(NSURL *)outputFileURL
  forConnections:(NSArray *)connections
  dueToError:(NSError *)error
```

Parameters

captureOutput

The capture file output that will finish writing the file.

outputURL

The file URL of the file that is being written.

connections

An array of `QTCaptureConnection` objects owned by the receiver that provided the data that is being written to the file.

error

An error describing what caused the file to stop recording, or `nil` if there was no error.

Discussion

This method is called when the file output will stop recording new samples to the file at `outputFileURL`, either because `recordToFile:` or `recordToFile:bufferDestination:` was called, or because an error, described by the error parameter, occurred (if no error occurred, the error parameter will be `NIL`). Delegates should also implement

`captureOutput:didFinishRecordingToOutputFileAtURL:forConnections:dueToError:` to be notified when the file is ready to be opened by applications.



Warning: Applications should not assume that this method will be called on the main thread.

Availability

Mac OS X v10.5 and later.

captureOutput:willStartRecordingToOutputFileURL:forConnections:

Informs the delegate when the output is about to start writing to a file.

```
- (void)captureOutput:(QTCaptureFileOutput *)captureOutput
  willStartRecordingToOutputFileURL:(NSURL *)fileURL
  forConnections:(NSArray *)connections
```

Parameters

captureOutput

The capture file output that will start writing the file.

outputURL

The file URL of the file that will be written.

connections

An array of `QTCaptureConnection` objects owned by the receiver that provided the data that will be written to the file.

Discussion

Applications should not assume that this method will be called on the main thread.

Availability

Mac OS X v10.5 and later.

compressionOptionsForConnection:

Returns the options the receiver uses to compress media on the given connection as it is being captured.

```
- (QTCompressionOptions *)compressionOptionsForConnection:(QTCaptureConnection
 *)connection
```

Parameters

connection

The connection containing the media to be compressed.

Return Value

A `QTCompressionOptions` object detailing the options being used to compress captured media on the given connection, or `NIL` if the media will not be recompressed.

Discussion

This method returns the options for compressing media set with the `setCompressionOptions:forConnection:` method. If the receiver should not recompress the output media, this method returns `NIL`. The default value is `NIL`.

Availability

Mac OS X v10.5 and later.

Declared In

QTCaptureFileOutput.h

delegate

Returns the receiver's delegate.

- (id)delegate

Availability

Mac OS X v10.5 and later.

Declared In

QTCaptureFileOutput.h

maximumRecordedDuration

Returns the maximum duration of the media that should be recorded by the receiver.

- (QTime)maximumRecordedDuration

Return ValueThe maximum time to be recorded, or `QTZeroTime` if there is no limit set.**Discussion**

This method returns a soft limit on the duration of recorded files set by `setMaximumRecordedDuration:`. Delegates can determine what to do when the limit is reached by implementing the `captureOutput:shouldChangeOutputFileAtURL:forConnections:dueToError:` method. By default, the current output file is set to `NIL` when the limit is reached.

Availability

Mac OS X v10.5 and later.

Declared In

QTCaptureFileOutput.h

maximumRecordedFileSize

Returns the maximum file size, in bytes, of the file that should be recorded by the receiver.

- (UInt64)maximumRecordedFileSize

Return Value

The maximum file size, in bytes, to be recorded, or 0 if there is no limit set.

Discussion

This method returns a soft limit on the duration of recorded files set by `setMaximumRecordedFileSize:`. Delegates can determine what to do when the limit is reached by implementing the `captureOutput:shouldChangeOutputFileAtURL:forConnections:dueToError:` method. By default, the current output file is set to `NIL` when the limit is reached.

Availability

Mac OS X v10.5 and later.

Declared In

QTCaptureFileOutput.h

outputFileURL

Returns the file written to by the receiver.

- (NSURL *)outputFileURL

Return Value

An NSURL object containing the file URL of the file currently being written by the receiver. Returns NIL if the receiver is not recording to any file.

Availability

Mac OS X v10.5 and later.

Declared In

QTCaptureFileOutput.h

recordedDuration

Returns the duration of the media recorded by the receiver.

- (QTime)recordedDuration

Return Value

The recorded time.

Discussion

If recording is in progress, this method returns the total time recorded so far. Otherwise, this method returns the time recorded in the most recent recording.

Availability

Mac OS X v10.5 and later.

Declared In

QTCaptureFileOutput.h

recordedFileSize

Returns the size, in bytes, of the data recorded by the receiver to output files.

- (UInt64)recordedFileSize

Return Value

The recorded size, in bytes.

Discussion

If a recording is in progress, this method returns the size in bytes of the data recorded so far. Otherwise, this method returns the size in the most recent recording.

Availability

Mac OS X v10.5 and later.

Declared In

QTCaptureFileOutput.h

recordToOutputFileURL:

Sets the file written to by the receiver.

```
- (void)recordToOutputFileURL:(NSURL *)outputURL
```

Parameters

outputURL

An NSURL object containing the URL of the output file, or `NIL` if the receiver should not record to any file. This method throws an `NSInvalidArgumentException` if the URL is not a valid file URL.

Discussion

The method sets the file URL to which the receiver is currently writing output media. If a file at the given URL already exists when capturing starts, the existing file is overwritten. If `NIL` is passed as the file URL, the receiver will stop recording to any file. If this method is invoked while an existing output file was already being recorded, no media samples are discarded between the old file and the new file. The sample buffer currently in flight when this method is called will always be written to the new file. Applications can specify where the sample buffer currently in flight will be recorded using the `recordToOutputFileURL:bufferDestination:` method. When the new file is set, applications cannot open the old file until it has finished recording in the background. Delegates should implement the `captureOutput:didFinishRecordingToOutputFileAtURL:forConnections:dueToError:` to be notified when the file is ready to be opened.

Availability

Mac OS X v10.5 and later.

Declared In

QTCaptureFileOutput.h

recordToOutputFileURL:bufferDestination:

Sets the file written to by the receiver, specifying where the sample buffer currently in flight should be recorded.

```
- (void)recordToOutputFileURL:(NSURL *)url
    bufferDestination:(QTCaptureFileOutputBufferDestination)bufferDestination
```

Parameters

outputURL

An NSURL object containing the URL of the output file, or `NIL` if the receiver should not record to any file. This method throws an `NSInvalidArgumentException` if the URL is not a valid file URL.

bufferDestination

A buffer destination specifying which file should contain the buffer currently in flight.

Discussion

The method sets the file URL to which the receiver is currently writing output media. If a file at the given URL already exists when capturing starts, the existing file will be overwritten. If `NIL` is passed as the file URL, the receiver will stop recording to any file. If this method is invoked while an existing output file was already being recorded, no media samples will be discarded between the old file and the new file. Applications can specify where the sample buffer currently in flight will be recorded using the `bufferDestination` argument. When the new file is set, applications will not be able to open the old file until it has finished recording in the background. Delegates should implement the `captureOutput:didFinishRecordingToOutputFileAtURL:forConnections:dueToError:` method to be notified when the file is ready to be opened.

Availability

Mac OS X v10.5 and later.

Declared In

`QTCaptureFileOutput.h`

setCompressionOptions:forConnection:

Sets the options the receiver uses to compress media on the given connection as it is being captured.

```
- (void)setCompressionOptions:(QTCompressionOptions *)compressionOptions
    forConnection:(QTCaptureConnection *)connection
```

Parameters

compressionOptions *compressionOptions*

A `QTCompressionOptions` object detailing the options being used to compress captured media, or `NIL` if the media should not be recompressed.

connection

The connection containing the media to be compressed.

Discussion

This method sets the options for compressing media as it is being captured. If compression cannot be performed in real time, the receiver will drop frames in order to remain synchronized with the session. If the receiver does not recompress the output media, this method should be passed `NIL`. The default value is `NIL`.

Availability

Mac OS X v10.5 and later.

Declared In

`QTCaptureFileOutput.h`

setDelegate:

Sets the receiver's delegate.

```
- (void)setDelegate:(id)delegate
```

Availability

Mac OS X v10.5 and later.

Declared In

`QTCaptureFileOutput.h`

setMaximumRecordedDuration:

Sets the maximum duration of the media that should be recorded by the receiver.

```
- (void)setMaximumRecordedDuration:(QTime)maximumRecordedDuration
```

Parameters

maximumRecordedDuration

The maximum time to be recorded, or `QTZeroTime` if there should be no limit.

Discussion

This method sets a soft limit on the duration of recorded files. Delegates can determine what to do when the limit is reached by implementing the

`captureOutput:shouldChangeOutputFileAtURL:forConnections:dueToError:` method. By default, the current output file is set to `NIL` when the limit is reached.

Availability

Mac OS X v10.5 and later.

Declared In

`QTCaptureFileOutput.h`

setMaximumRecordedFileSize:

Sets the maximum file size, in bytes, of the file that should be recorded by the receiver.

```
- (void)setMaximumRecordedFileSize:(UInt64)maximumRecordedFileSize
```

Parameters

maximumRecordedFileSize

The maximum size, in bytes, to be recorded, or 0 if there should be no limit.

Discussion

This method sets a soft limit on the size of recorded files. Delegates can determine what to do when the limit is reached by implementing the

`captureOutput:shouldChangeOutputFileAtURL:forConnections:dueToError:` method. By default, the current output file is set to `NIL` when the limit is reached.

Availability

Mac OS X v10.5 and later.

Declared In

`QTCaptureFileOutput.h`

Constants

QTCaptureFileOutputBufferDestination

Specifies where the media sample buffer currently in flight should be written when changing output files.


```
enum {  
    QTCaptureFileOutputBufferDestinationNewFile = 0,  
    QTCaptureFileOutputBufferDestinationOldFile = 1  
};
```

Constants

`QTCaptureFileOutputBufferDestination`
`QTCaptureFileOutputBufferDestinationNewFile` tells the output to include the buffer currently in flight in the old file. `QTCaptureFileOutputBufferDestinationOldFile` tells the output to include the buffer currently in flight in the new file.

Declared In

`QTCaptureFileOutput.h`

Document Revision History

This table describes the changes to *QTCaptureFileOutput Reference*.

Date	Notes
2007-07-23	New document that describes the Objective-C API for supporting and working with QuickTime Capture.

REVISION HISTORY

Document Revision History

Index

C

captureOutput:didFinishRecordingToOutputFileAtURL:
forConnections:dueToError: **instance method** [7](#)

captureOutput:didOutputSampleBuffer:
fromConnection: **instance method** [7](#)

captureOutput:didStartRecordingToOutputFileURL:
forConnections: **instance method** [8](#)

captureOutput:mustChangeOutputFileAtURL:
forConnections:dueToError: **instance method** [8](#)

captureOutput:shouldChangeOutputFileAtURL:
forConnections: **instance method** [9](#)

captureOutput:
willFinishRecordingToOutputFileAtURL:
forConnections:dueToError: **instance method** [10](#)

captureOutput:willStartRecordingToOutputFileURL:
forConnections: **instance method** [11](#)

compressionOptionsForConnection: **instance method** [11](#)

D

delegate **instance method** [12](#)

M

maximumRecordedDuration **instance method** [12](#)
maximumRecordedFileSize **instance method** [12](#)

O

outputFileURL **instance method** [13](#)

Q

QTCaptureFileOutputBufferDestination [16](#)
QTCaptureFileOutputBufferDestination **constant** [17](#)

R

recordedDuration **instance method** [13](#)
recordedFileSize **instance method** [13](#)
recordToOutputFileURL: **instance method** [14](#)
recordToOutputFileURL:bufferDestination:
instance method [14](#)

S

setCompressionOptions:forConnection: **instance method** [15](#)
setDelegate: **instance method** [15](#)
setMaximumRecordedDuration: **instance method** [16](#)
setMaximumRecordedFileSize: **instance method** [16](#)