QTCaptureFileOutput Class Reference

Audio & Video

2010-08-03

Contents

QTCaptureFileOutput 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; QuickTime 7.2.1.
Declared in	QTCaptureFileOutput.h
Related sample code	MyRecorder QT Capture Widget QTCompressionOptionsWindow 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 `recordToOutputFileURL:` (page 12) and `recordToOutputFileURL:bufferDestination:` (page 12) methods. On successive invocations of these methods, the output file can by 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 recorded 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:` (page 17) method. Currently, the only concrete subclass of this class is `QTCaptureMovieFileOutput`.

Tasks

Recording File Outputs

– `outputFileURL` (page 10)
  Returns the file URL of the file to which the receiver is currently recording incoming buffers.

– `recordToOutputFileURL:` (page 12)
  Calls `recordToOutputFileURL:bufferDestination:` with a buffer destination of `QTCaptureFileOutputBufferDestinationNewFile`.

- `recordToOutputFileURL:bufferDestination:` (page 12)
  Sets the file written to by the receiver, specifying where the sample buffer currently in flight should be recorded.
- `recordedDuration` (page 11)
  Returns the duration of the media recorded by the receiver.
- `recordedFileSize` (page 11)
  Returns the size, in bytes, of the data recorded by the receiver to output files.
- `maximumRecordedDuration` (page 9)
  Returns the maximum duration of the media that should be recorded by the receiver.
- `setMaximumRecordedDuration:` (page 14)
  Sets the maximum duration of the media that should be recorded by the receiver.
- `maximumRecordedFileSize` (page 9)
  Returns the maximum file size, in bytes, of the file that should be recorded by the receiver.
- `setMaximumRecordedFileSize:` (page 15)
  Sets the maximum file size, in bytes, of the file that should be recorded by the receiver.
- `compressionOptionsForConnection:` (page 7)
  Returns the options the receiver uses to compress media on the given connection as it is being captured.
- `setCompressionOptions:forConnection:` (page 13)
  Sets the options the receiver uses to compress media on the given connection as it is being captured.
- `delegate` (page 8)
  Returns the receiver's delegate.
- `setDelegate:` (page 14)
  Sets the receiver's delegate.

Methods That Control Recording

- `isRecordingPaused` (page 8)
  Returns whether recording to the current output file is paused.
- `pauseRecording` (page 11)
  Pauses recording to the current output file.
- `resumeRecording` (page 13)
  Resumes recording to the current output file after it was previously paused using pauseRecording.
- `maximumVideoSize` (page 9)
  Returns the maximum dimensions within which the receiver will record video.
- `setMaximumVideoSize:` (page 15)
  Sets the maximum dimensions within which the receiver should record video.
- `minimumVideoFrameInterval` (page 10)
  Returns the minimum time interval between which the receiver will record consecutive video frames.
- `setMinimumVideoFrameInterval:` (page 16)
  Sets the minimum time interval between which the receiver should record consecutive video frames.

Methods Implemented by the Delegate

`– captureOutput:didOutputSampleBuffer:fromConnection:` (page 17) *delegate method*
: Gives the delegate the opportunity to inspect samples as they are received by the output and start and stop capturing at exact times.

`– captureOutput:willStartRecordingToOutputFileAtURL:forConnections:` (page 22) *delegate method*
: Informs the delegate when the output is about to start writing to a file.

`– captureOutput:didStartRecordingToOutputFileAtURL:forConnections:` (page 19) *delegate method*
: Informs the delegate when the output has started writing to a file.

`– captureOutput:shouldChangeOutputFileAtURL:forConnections:dueToError:` (page 20) *delegate method*
: Gives the delegate the opportunity to determine what should happen when an output file has reached a soft limit.

`– captureOutput:mustChangeOutputFileAtURL:forConnections:dueToError:` (page 19) *delegate method*
: Informs the delegate when an output file can no longer be written using the incoming media.

`– captureOutput:willFinishRecordingToOutputFileAtURL:forConnections:dueToError:` (page 21) *delegate method*
: Informs the delegate when the output will stop writing new samples to a file.

`– captureOutput:didFinishRecordingToOutputFileAtURL:forConnections:dueToError:` (page 16) *delegate method*
: Informs the delegate when an output file is ready to be opened by applications.

`– captureOutput:didPauseRecordingToOutputFileAtURL:forConnections:` (page 18) *delegate method*
: Called whenever the output is recording to a file and successfully pauses the recording at the request of the client.

`– captureOutput:didResumeRecordingToOutputFileAtURL:forConnections:` (page 18) *delegate method*
: Called whenever the output, at the request of the client, successfully resumes a file recording that was paused.

Instance Methods

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 re-compressed.

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

Availability
Mac OS X v10.5 and later; QuickTime 7.2.1.

Declared In
`QTCaptureFileOutput.h`

delegate

Returns the receiver's delegate.

```
- (id)delegate
```

Discussion
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; QuickTime 7.2.1.

Declared In
`QTCaptureFileOutput.h`

isRecordingPaused

Returns whether recording to the current output file is paused.

```
- (BOOL)isRecordingPaused
```

Return Value
Returns `YES` if recording to the current output file is paused and returns `NO` otherwise.

Discussion
This method returns whether recording to the file returned by `outputFileURL` has been previously paused using the `pauseRecording` method. When a recording is paused, captured samples are not written to the output file, but new samples can be written to the same file in the future by calling `resumeRecording`. The value of this method is key value observable using the key `@"recordingPaused"`.

Availability
QuickTime 7.6.3 or later; QuickTime 7.2.1.

Declared In
`QTCaptureFileOutput.h`

maximumRecordedDuration

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

```
- (QTTime)maximumRecordedDuration
```

Return Value
The 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; QuickTime 7.2.1.

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; QuickTime 7.2.1.

Declared In
`QTCaptureFileOutput.h`

maximumVideoSize

Returns the maximum dimensions within which the receiver will record video.

```
- (NSSize)maximumVideoSize
```

Return Value
An `NSSize` specifying the maximum dimensions at which the receiver should record video. Returns `NSZeroSize` if there is no limit.

Discussion
This method returns the maximum limit on the dimensions of video that the receiver records to a file previously set by `setMaximumVideoSize:`. When a size is set, all video recorded by the receiver will be no larger than the specified size, while still preserving the original aspect ratio of the content. A value of `NSZeroSize` indicates that there should be no limit. If this is set to a value other than `NSZeroSize`, device native compressed video, such as DV video, will be decompressed so that it can be resized. By default, there is no limit on the maximum recorded video size.

Availability
QuickTime 7.6.3 or later.

Declared In
`QTCaptureFileOutput.h`

minimumVideoFrameInterval

Returns the minimum time interval between which the receiver will record consecutive video frames.

```
- (NSTimeInterval)minimumVideoFrameInterval
```

Return Value
An `NSTimeInterval` specifying the minimum interval between video frames. Returns 0 if there is no frame rate limit set.

Discussion
This method returns the minimum amount of time that should separate consecutive frames recorded by the receiver. This is equivalent to the inverse of the maximum frame rate. A value of 0 indicates an unlimited maximum frame rate. If this is set to a value other than 0, device native compressed video, such as DV video, will be decompressed so that its frame rate can be adjusted. The default value is 0.

Availability
QuickTime 7.6.3 or later.

Declared In
`QTCaptureFileOutput.h`

outputFileURL

Returns the file URL of the file to which the receiver is currently recording incoming buffers.

```
- (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; QuickTime 7.2.1.

Declared In
`QTCaptureFileOutput.h`

pauseRecording

Pauses recording to the current output file.

```
- (void)pauseRecording
```

Discussion
This method causes the receiver to stop writing captured samples to the current output file returned by `outputFileURL`, but leaves the file open so that samples can be written to it in the future, when resumeRecording is called. This allows clients to record multiple media segments that are not contiguous in time to a single file.

When clients stop recording or change files using `recordToOutputFileURL:bufferDestination:` or recording automatically stops due to an error condition while recording is paused, the output file will be finished and closed normally without requiring a matching call to `resumeRecording`. When there is no current output file, or when recording is already paused, this method does nothing. This method can be called within the `captureOutput:didOutputSampleBuffer:fromConnection:` delegate method to pause recording after an exact media sample.

Availability
QuickTime 7.6.3 or later.

Declared In
`QTCaptureFileOutput.h`

recordedDuration

Returns the duration of the media recorded by the receiver.

```
- (QTTime)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; QuickTime 7.2.1.

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; QuickTime 7.2.1.

Declared In
`QTCaptureFileOutput.h`

recordToOutputFileURL:

Calls `recordToOutputFileURL:bufferDestination:` with a buffer destination of `QTCaptureFileOutputBufferDestinationNewFile`.

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

Parameters

`url`
: An `url` 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.

Related Sample Code
MyRecorder
QT Capture Widget
QTCompressionOptionsWindow
QTRecorder

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`

resumeRecording

Resumes recording to the current output file after it was previously paused using pauseRecording.

```
- (void)resumeRecording
```

Discussion

This method causes the receiver to resume writing captured samples to the current output file returned by `outputFileURL`, after recording was previously paused using `pauseRecording`. This allows clients to record multiple media segments that are not contiguous in time to a single file. When there is no current output file, or when recording is not paused, this method does nothing. This method can be called within the `captureOutput:didOutputSampleBuffer:fromConnection:` delegate method to resume recording at an exact media sample.

Availability

QuickTime 7.6.3 or 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

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

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 re-compress the output media, this method should be passed `NIL`. The default value is `NIL`.

Availability

Mac OS X v10.5 and later; QuickTime 7.2.1.

Declared In

`QTCaptureFileOutput.h`

setDelegate:

Sets the receiver's delegate.

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

Discussion

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; QuickTime 7.2.1.

Related Sample Code

QT Capture Widget

Declared In

`QTCaptureFileOutput.h`

setMaximumRecordedDuration:

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

```
- (void)setMaximumRecordedDuration:(QTTime)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; QuickTime 7.2.1.

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 is 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; QuickTime 7.2.1.

Declared In
`QTCaptureFileOutput.h`

setMaximumVideoSize:

Sets the maximum dimensions within which the receiver should record video.

```
- (void)setMaximumVideoSize:(NSSize)maximumVideoSize
```

Parameters

maximumVideoSize
An `NSSize` specifying the maximum dimensions at which the receiver should record video. A value of `NSZeroSize` indicates that there should be no limit.

Discussion
This method sets the maximum limit on the dimensions of video that the receiver records to a file. When a size is set, all video recorded by the receiver will be no larger than the specified size, while still preserving the original aspect ratio of the content. A value of `NSZeroSize` indicates that there should be no limit. If this is set to a value other than `NSZeroSize`, device native compressed video, such as DV video, will be decompressed so that it can be resized. By default, there is no limit on the maximum recorded video size.

Availability
QuickTime 7.6.3 or later.

Declared In
`QTCaptureFileOutput.h`

setMinimumVideoFrameInterval:

Sets the minimum time interval between which the receiver should record 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 recorded by the receiver. This is equivalent to the inverse of the maximum frame rate. A value of 0 indicates an unlimited maximum frame rate. If this is set to a value other than 0, device native compressed video, such as DV video, will be decompressed so that its frame rate can be adjusted. The default value is 0.

Availability
QuickTime 7.6.3 or later.

Declared In
`QTCaptureFileOutput.h`

Delegate 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.

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

Availability
Mac OS X v10.5 and later; QuickTime 7.2.1.

Declared In
`QTCaptureFileOutput.h`

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.

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; QuickTime 7.2.1.

Declared In
`QTCaptureFileOutput.h`

captureOutput:didPauseRecordingToOutputFileAtURL:forConnections:

Called whenever the output is recording to a file and successfully pauses the recording at the request of the client.

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

Parameters

captureOutput

The capture file output that has paused its file recording.

fileURL

The file URL of the file that is being written.

connections

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

Discussion

Delegates can use this method to be informed when a request to pause recording is actually respected. It is safe for delegates to change what the file output is currently doing (starting a new file, for example) from within this method. Clients should not assume that this method will be called on the main thread, and should also try to make this method as efficient as possible. If recording to a file is stopped, either manually or due to an error, this method is not guaranteed to be called, even if a previous call to `pauseRecording` was made.

Availability

QuickTime 7.2.1 or later.

Declared In

`QTCaptureFileOutput.h`

captureOutput:didResumeRecordingToOutputFileAtURL:forConnections:

Called whenever the output, at the request of the client, successfully resumes a file recording that was paused.

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

Parameters

captureOutput

The capture file output that has resumed its paused file recording.

fileURL

The file URL of the file that is being written.

connections

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

Discussion

Delegates can use this method to be informed when a request to resume a paused recording is actually respected. It is safe for delegates to change what the file output is currently doing (starting a new file, for example) from within this method. Clients should not assume that this method will be called on the main thread, and should also try to make this method as efficient as possible. If recording to a file is stopped, either manually or due to an error, this method is not guaranteed to be called, even if a previous call to `resumeRecording` was made.

Availability
QuickTime 7.2.1 or later.

Declared In
`QTCaptureFileOutput.h`

captureOutput:didStartRecordingToOutputFileAtURL:forConnections:

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

```
- (void)captureOutput:(QTCaptureFileOutput *)captureOutput
    didStartRecordingToOutputFileAtURL:(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; QuickTime 7.2.1.

Declared In
`QTCaptureFileOutput.h`

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.

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

Availability

Mac OS X v10.5 and later; QuickTime 7.2.1.

Declared In

`QTCaptureFileOutput.h`

captureOutput:shouldChangeOutputFileAtURL:forConnections:dueToError:

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.

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

Availability
Mac OS X v10.5 and later; QuickTime 7.2.1.

Declared In
`QTCaptureFileOutput.h`

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.

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

Availability
Mac OS X v10.5 and later; QuickTime 7.2.1.

Declared In
`QTCaptureFileOutput.h`

captureOutput:willStartRecordingToOutputFileAtURL:forConnections:

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

```
- (void)captureOutput:(QTCaptureFileOutput *)captureOutput
    willStartRecordingToOutputFileAtURL:(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; QuickTime 7.2.1.

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
};
typedef NSUInteger QTCaptureFileOutputBufferDestination;
```

Constants

`QTCaptureFileOutputBufferDestinationNewFile`
This tells the output to include the buffer currently in flight in the old file.

Available in Mac OS X v10.5 and later.

Declared in `QTCaptureFileOutput.h`.

`QTCaptureFileOutputBufferDestinationOldFile`

This tells the output to include the buffer currently in flight in the new file.

Available in Mac OS X v10.5 and later.

Declared in `QTCaptureFileOutput.h`.

Document Revision History

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

Date	Notes
2010-08-03	Corrected delegate method signatures.
2009-05-20	Updated availability information for Mac OS X v10.6. Minor fixes.
2007-07-23	New document that describes the Objective-C API for supporting and working with QuickTime Capture.