

---

# QTCaptureConnection Class Reference

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



2009-05-06



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

Numbers is a trademark of Apple Inc.

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

---

## **QTCaptureConnection Class Reference 5**

---

Overview 5

Tasks 6

    Getting and Setting Connection Attributes 6

Instance Methods 6

    attributeForKey: 6

    attributesReadOnly: 7

    connectionAttributes 7

    formatDescription 7

    isEnabled 8

    mediaType 8

    owner 8

    setAttribute:forKey: 9

    setConnectionAttributes: 9

    setEnabled: 9

Constants 10

    Audio Attributes 10

Notifications 11

    QTCaptureConnectionAttributeDidChangeNotification 11

    QTCaptureConnectionAttributeWillChangeNotification 11

    QTCaptureConnectionChangedAttributeKey 12

    QTCaptureConnectionFormatDescriptionDidChangeNotification 12

    QTCaptureConnectionFormatDescriptionWillChangeNotification 12

---

## **Document Revision History 13**

---

## **Index 15**

---



# QTCaptureConnection Class Reference

---

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

## Overview

This class represents a connection over which a single stream of media data is sent from a `QTCaptureInput` to a `QTCaptureSession` and from a `QTCaptureSession` to a `QTCaptureOutput`.

Instances of `QTCaptureConnection` wrap individual media streams that can be provided by `QTCaptureInput` objects and received by `QTCaptureOutput` objects. Connections can have a QuickTime media type, such as `QTMediaTypeVideo` and `QTMediaTypeSound`, and a format description that describes the media sent or received across the connection. Individual connections belonging to an input can be enabled or disabled to restrict what media enters a capture session, while connections belonging to an output can be enabled or disabled to restrict what media enters the output from the capture session. In addition, if a `QTCaptureConnection` wraps a stream of audio media, it provides a number of attributes to control the volume, mix, and enabled channels of the audio passing through it.

`QTCaptureConnection` objects can have extended attributes that applications can read using the `attributeForKey:` and `connectionAttributes` methods. Some attributes, for which the `attributeIsReadOnly:` method returns `NO`, can be edited using the `setAttributeForKey:` and `setConnectionAttributes:` methods. In addition to these explicit methods, applications can use key-value coding to get and set extended attributes. For an object that supports a given attribute, `valueForKey:` will be functionally identical to `attributeForKey:`, and `setValueForKey:` will be identical to `setAttributeForKey:`. 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 and Setting Connection Attributes

- [attributeForKey:](#) (page 6)  
Returns the current value of the connection attribute for key.
- [attributeIsReadOnly:](#) (page 7)  
Returns a Boolean value indicating whether the given attribute for the connection cannot be modified.
- [connectionAttributes](#) (page 7)  
Returns a dictionary of all attributes set for the receiver.
- [formatDescription](#) (page 7)  
Returns the format description of the receiver.
- [isEnabled](#) (page 8)  
Returns a Boolean value indicating whether the receiver is enabled.
- [mediaType](#) (page 8)  
Returns the QuickTime media type of the receiver.
- [owner](#) (page 8)  
Returns the `QTCaptureInput` or `QTCaptureOutput` object that owns the receiver.
- [setAttribute:forKey:](#) (page 9)  
Sets a connection attribute for the given key.
- [setConnectionAttributes:](#) (page 9)  
Sets the connection's attributes from the key-value pairs specified in the given dictionary.
- [setEnabled:](#) (page 9)  
Sets whether the receiver is enabled.

## Instance Methods

### **attributeForKey:**

Returns the current value of the connection attribute for key.

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

#### **Discussion**

Use this method to get attributes of a connection. 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; QuickTime 7.2.1.

#### **Related Sample Code**

QTRecorder

**Declared In**

QTCaptureConnection.h

**attributeIsReadOnly:**

Returns a Boolean value indicating whether the given attribute for the connection cannot be modified.

- (BOOL)attributeIsReadOnly:(NSString \*)*attributeKey***Return Value**

Returns YES if the attribute cannot be modified; otherwise, NO.

**Availability**

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

**Declared In**

QTCaptureConnection.h

**connectionAttributes**

Returns a dictionary of all attributes set for the receiver.

- (NSDictionary \*)connectionAttributes

**Discussion**

Applications can use this method to determine what attributes a specific connection supports.

**Availability**

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

**Declared In**

QTCaptureConnection.h

**formatDescription**

Returns the format description of the receiver.

- (QTFormatDescription \*)formatDescription

**Discussion**

This method returns the format description of the connection, allowing applications to monitor various attributes of the media being sent or received by the connection (the display size of video media, for example). Applications can be notified of changes to the connection's format by registering to receive `QTCaptureConnectionFormatDescriptionWillChangeNotification` and `QTCaptureConnectionFormatDescriptionDidChangeNotification` notifications or by adding a key-value observer to the connection for the key `@"formatDescription"`.

**Availability**

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

**Related Sample Code**

QTRecorder

**Declared In**

QTCaptureConnection.h

**isEnabled**

Returns a Boolean value indicating whether the receiver is enabled.

- (BOOL)isEnabled

**Discussion**

This method returns a Boolean indicating whether the receiver is enabled to send or receive media data. Individual connections can be enabled or disabled using the `setEnabled:` method.

**Availability**

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

**Declared In**

QTCaptureConnection.h

**mediaType**

Returns the QuickTime media type of the receiver.

- (NSString \*)mediaType

**Return Value**A QuickTime media type, as defined in `QTMedia.h`.**Discussion**

This method returns the QuickTime media type, such as `QTMediaTypeVideo` and `QTMediaTypeSound`, of the receiver.

**Availability**

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

**Declared In**

QTCaptureConnection.h

**owner**Returns the `QTCaptureInput` or `QTCaptureOutput` object that owns the receiver.

- (id)owner

**Return Value**A `QTCaptureInput` or `QTCaptureOutput` object that uses the receiver as a media connection.**Discussion**

This method returns the input or output to which the receiver belongs. The returned input or output uses the receiver as a connection for sending or receiving a media stream.

**Availability**

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



**Declared In**

QTCaptureConnection.h

**setAttribute:forKey:**

Sets a connection attribute for the given key.

```
- (void)setAttribute:(id)attribute
    forKey:(NSString *)key
```

**Discussion**

Use this method to set attributes of a capture connection. The keys that can be used with this method are described in the Constants section. This method raises an `NSInvalidArgumentException` if the attribute is read-only or not supported by the receiver. Applications using key-value coding can also set an attribute for a given key by passing that key to the `NSObject setValue:forKey:` method.

**Availability**

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

**Declared In**

QTCaptureConnection.h

**setConnectionAttributes:**

Sets the connection's attributes from the key-value pairs specified in the given dictionary.

```
- (void)setConnectionAttributes:(NSDictionary *)connectionAttributes
```

**Discussion**

This method allows application to set multiple attributes on a connection at once. This method raises an `NSInvalidArgumentException` if any of the attributes in the dictionary are read-only or not supported by the receiver. Applications using key-value coding can also set multiple attributes using the `NSObject setValuesForKeysWithDictionary:` method using attribute keys as keys in the dictionary.

**Availability**

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

**Declared In**

QTCaptureConnection.h

**setEnabled:**

Sets whether the receiver is enabled.

```
- (void)setEnabled:(BOOL)enabled
```

**Discussion**

This method sets whether the receiver is enabled to send or receive media data.

**Availability**

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

**Declared In**

QTCaptureConnection.h

## Constants

### Audio Attributes

Applications can use the following constants to display audio level meters for specific connections and to specify the volumes of audio channels. These string values can be used in key paths for key-value coding, key-value observing, and bindings.

```
NSString * const QTCaptureConnectionAudioAveragePowerLevelsAttribute;
NSString * const QTCaptureConnectionAudioPeakHoldLevelsAttribute;
NSString * const QTCaptureConnectionAudioMasterVolumeAttribute;
NSString * const QTCaptureConnectionAudioVolumesAttribute;
NSString * const QTCaptureConnectionEnabledAudioChannelsAttribute;
```

**Constants**

QTCaptureConnectionAudioAveragePowerLevelsAttribute

An NSArray of NSNumbers that correspond to the average power, in decibels, of each audio stream sent through the connection.

Applications that wish to display audio level meters for a specific connection can periodically check the value of this attribute. Average power levels change quickly and appear jumpy on a level meter. 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 QTCaptureConnection.h.

QTCaptureConnectionAudioPeakHoldLevelsAttribute

An NSArray of NSNumbers that correspond to the peak hold level, in decibels, of each audio channel sent through the connection.

Applications that wish to display audio level meters for a specific connection can periodically check the value of this attribute. Peak hold levels remain at the maximum volume for about a second, and are often useful for displaying audio clipping. 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 QTCaptureConnection.h.

QTCaptureConnectionAudioMasterVolumeAttribute

An NSNumber that specifies the master volume of all audio channels sent through the connection.

The values are between 0.0 and 1.0 for normal volume, or greater than 1.0 for boosting the audio gain. This attribute determines the master volumes of all audio channels sent through the connection. Applications that need to set the volumes of individual channels can set the QTCaptureConnectionAudioVolumesAttribute attribute. 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 QTCaptureConnection.h.

**QTCaptureConnectionAudioVolumesAttribute**

An NSArray of NSNumbers that specify the volumes of audio channels sent through the connection.

The values are between 0.0 and 1.0 for normal volume, or greater than 1.0 for boosting the audio gain. This attribute determines the individual volumes of audio channels sent through the connection. Applications that need to set the master volume of all channels can set the QTCaptureConnectionAudioMasterVolumeAttribute attribute. 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 QTCaptureConnection.h.

**QTCaptureConnectionEnabledAudioChannelsAttribute**

An NSIndexSet that specifies which audio channels should be sent through the connection. The indices in the set should be between 0 and the number of volumes in QTCaptureConnectionAudioVolumesAttribute. This attribute allows applications to selectively disable certain audio channels from being sent through the connection. The value of this attribute should be an NSIndexSet that contains only the channels that should be used. By default, all audio channels are sent through a connection. 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 QTCaptureConnection.h.

## Notifications

The following are notifications enabling you to change attributes, keys, and format descriptions.

### **QTCaptureConnectionAttributeDidChangeNotification**

Posted when one of the connection's attributes has changed.

The notification's user info dictionary will contain the attribute key of the changed attribute for the key QTCaptureConnectionChangedAttributeKey.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

QTCaptureConnection.h

### **QTCaptureConnectionAttributeWillChangeNotification**

Posted when one of the connection's attributes is about to change.

The notification's user info dictionary will contain the attribute key of the changed attribute for the key QTCaptureConnectionChangedAttributeKey.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

QTCaptureConnection.h

### **QTCaptureConnectionChangedAttributeKey**

Used as a key in the user info dictionary passed to `QTCaptureConnectionAttributeWillChangeNotification`, and `QTCaptureConnectionAttributeDidChangeNotification` to indicate the key of that attribute that changed.

#### **Availability**

Available in Mac OS X v10.5 and later.

#### **Declared In**

`QTCaptureConnection.h`

### **QTCaptureConnectionFormatDescriptionDidChangeNotification**

Posted when the format description of a connection has changed.

Applications can be notified of changes to a connection's format by registering to receive this notification.

#### **Availability**

Available in Mac OS X v10.5 and later.

#### **Declared In**

`QTCaptureConnection.h`

### **QTCaptureConnectionFormatDescriptionWillChangeNotification**

Posted when the format description of a connection is about to change.

Applications can be notified of changes to a connection's format by registering to receive this notification.

#### **Availability**

Available in Mac OS X v10.5 and later.

#### **Declared In**

`QTCaptureConnection.h`

# Document Revision History

---

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

Date	Notes
2009-05-06	Fixed constant listings. Minor fixes.
2009-03-04	Added availability information; fixed method and attribute descriptions; fixed typos.
2007-07-23	New document that describes the Objective-C API for supporting and working with QuickTime Capture.

## REVISION HISTORY

### Document Revision History

# Index

---

## A

---

`attributeForKey:` [instance method 6](#)  
`attributeIsReadOnly:` [instance method 7](#)  
[Audio Attributes 10](#)

## C

---

`connectionAttributes` [instance method 7](#)

## F

---

`formatDescription` [instance method 7](#)

## I

---

`isEnabled` [instance method 8](#)

## M

---

`mediaType` [instance method 8](#)

## O

---

`owner` [instance method 8](#)

## Q

---

`QTCaptureConnectionAttributeDidChangeNotification`  
[notification 11](#)  
`QTCaptureConnectionAttributeWillChangeNotification`  
[notification 11](#)

`QTCaptureConnectionAudioAveragePowerLevelsAttribute` [constant 10](#)  
`QTCaptureConnectionAudioMasterVolumeAttribute`  
[constant 10](#)  
`QTCaptureConnectionAudioPeakHoldLevelsAttribute`  
[constant 10](#)  
`QTCaptureConnectionAudioVolumesAttribute`  
[constant 11](#)  
`QTCaptureConnectionChangedAttributeKey`  
[notification 12](#)  
`QTCaptureConnectionEnabledAudioChannelsAttribute`  
[constant 11](#)  
`QTCaptureConnectionFormatDescriptionDidChange-`  
`Notification` [notification 12](#)  
`QTCaptureConnectionFormatDescriptionWillChange-`  
`Notification` [notification 12](#)

## S

---

`setAttributeForKey:` [instance method 9](#)  
`setConnectionAttributes:` [instance method 9](#)  
`setEnabled:` [instance method 9](#)