

---

# IMAVMManager Class Reference

[Cocoa](#) > [Apple Applications](#)



2007-10-31



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

Apple, the Apple logo, Cocoa, iChat, Mac, and Mac OS are trademarks of Apple Inc., registered in the United States and other countries.

OpenGL is a registered trademark of Silicon Graphics, 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

---

## **IMAVManager Class Reference 5**

---

Overview	5
Tasks	5
Creating an IMAVManager Object	5
Getting and Setting Properties	5
Starting and Stopping Audio/Video Content	6
Optimizing Audio/Video Performance	6
Managing Audio Channels	6
Sharing Files	6
Class Methods	6
sharedAVManager	6
Instance Methods	7
audioDeviceChannels	7
audioDeviceUID	7
numberOfAudioChannels	8
setNumberOfAudioChannels:	8
setVideoDataSource:	8
setVideoOptimizationOptions:	9
start	9
state	10
stop	10
URLToShare	11
videoDataSource	11
videoOptimizationOptions	11
Constants	12
IMAVManagerState	12
IMVideoOptimizationOptions	13
Notifications	14
IMAVManagerStateChangedNotification	14
IMAVManagerURLToShareChangedNotification	14

---

## **Document Revision History 15**

---

## **Index 17**

---



# IMAVManager Class Reference

---

<b>Inherits from</b>	NSObject
<b>Conforms to</b>	NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/InstantMessage.framework
<b>Availability</b>	Available in Mac OS X v10.5 and later.
<b>Companion guide</b>	Instant Message Programming Guide
<b>Declared in</b>	IMAVManager.h

## Overview

The `IMAVManager` class is used to manage the state and configuration of auxiliary audio/video input to iChat AV—a feature that is called iChat Theater. There is only one shared instance of the `IMAVManager` class.

The `IMAVManager` shared object allows clients to provide audio and video to a running conference in iChat AV. Video is provided by supplying a data source object to receive periodic callbacks for individual frames, and audio is provided through an audio device and channel. The state of the shared `IMAVManager` object allows clients to configure the user interface appropriately.

## Tasks

### Creating an IMAVManager Object

+ [sharedAVManager](#) (page 6)

Returns the shared instance of the `IMAVManager` object, creating it if the object doesn't exist yet.

### Getting and Setting Properties

- [state](#) (page 10)

Returns the current state of the receiver.

- [videoDataSource](#) (page 11)

Returns the receiver's video data source object.

- [setVideoDataSource:](#) (page 8)

Sets the receiver's video data source object that provides video data to iChat AV.

## Starting and Stopping Audio/Video Content

- [start](#) (page 9)  
Starts sending audio and video to iChat AV.
- [stop](#) (page 10)  
Stops sending audio and video to iChat AV.

## Optimizing Audio/Video Performance

- [setVideoOptimizationOptions:](#) (page 9)  
Sets the video optimization options.
- [videoOptimizationOptions](#) (page 11)  
Returns the video optimization options.

## Managing Audio Channels

- [setNumberOfAudioChannels:](#) (page 8)  
Sets the number of audio channels.
- [numberOfAudioChannels](#) (page 8)  
Returns the number of audio channels.
- [audioDeviceUID](#) (page 7)  
Returns the audio device UID.
- [audioDeviceChannels](#) (page 7)  
Returns an array of audio device channel numbers used by the receiver.

## Sharing Files

- [URLToShare](#) (page 11)  
Returns the file URL of the document that the user chose to share over iChat Theater.

## Class Methods

### sharedAVManager

Returns the shared instance of the `IMAVManager` object, creating it if the object doesn't exist yet.

```
+ (IMAVManager *)sharedAVManager
```

#### Return Value

The shared `IMAVManager` object.

#### Availability

Available in Mac OS X v10.5 and later.

**Declared In**

IMAVManager.h

## Instance Methods

### audioDeviceChannels

Returns an array of audio device channel numbers used by the receiver.

- (NSArray \*)audioDeviceChannels

**Return Value**

An array of audio device channel numbers. If the number of audio channels is set to 2, then the first number in the array is the left channel and the second number is the right channel. Returns `nil` if the receiver is not in the `IMAVRunning` state. Also returns `nil` if the [setNumberOfAudioChannels:](#) (page 8) method is not invoked prior to invoking this method with 1 or 2 as the argument.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [audioDeviceUID](#) (page 7)

**Declared In**

IMAVManager.h

### audioDeviceUID

Returns the audio device UID.

- (NSString \*)audioDeviceUID

**Return Value**

A valid UID when the receiver is in the `IMAVRunning` state; otherwise, `nil`. Also returns `nil` if the [setNumberOfAudioChannels:](#) (page 8) method is not invoked prior to invoking this method with 1 or 2 as the argument.

**Discussion**

You can obtain the device by calling the `AudioHardwareGetProperty` function with the returned UID and the `kAudioHardwarePropertyDeviceForUID` constant as arguments.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [audioDeviceChannels](#) (page 7)  
- [state](#) (page 10)

**Declared In**

IMAVManager.h

## numberOfAudioChannels

Returns the number of audio channels.

- (NSInteger)numberOfAudioChannels

### Return Value

The number of audio channels.

### Availability

Available in Mac OS X v10.5 and later.

### See Also

- [setNumberOfAudioChannels:](#) (page 8)

### Declared In

IMAVManager.h

## setNumberOfAudioChannels:

Sets the number of audio channels.

- (void)setNumberOfAudioChannels:(NSInteger)count

### Parameters

*count*

The number of audio channels to configure. The allowed values are 0, 1, and 2. If 0, audio is disabled. If 1, audio is set to mono, and if 2, audio is stereo.

### Discussion

Sets the number of audio channels that are configured after invoking [start](#) (page 9).

### Availability

Available in Mac OS X v10.5 and later.

### See Also

- [state](#) (page 10)

- [numberOfAudioChannels](#) (page 8)

### Declared In

IMAVManager.h

## setVideoDataSource:

Sets the receiver's video data source object that provides video data to iChat AV.

- (void)setVideoDataSource:(id)dataSource



**Parameters***dataSource*

An object that conforms to the `IMVideoDataSource` informal protocol. The object needs to respond to either the `renderIntoPixelFormat:forTime:` and `getPixelFormatPixelFormat:` methods, or the `renderIntoOpenGLBuffer:onScreen:forTime:` and `getOpenGLBufferContext:PixelFormat:` methods for OpenGL content. Any `NSView` object can also be a video data source. The Instant Message framework adds video rendering capabilities to `NSView` and all its subclasses. Pass `nil` to remove the receiver's video data source. The data source is not retained by the receiver.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [videoDataSource](#) (page 11)

**Declared In**

`IMAVManager.h`

**setVideoOptimizationOptions:**

Sets the video optimization options.

```
- (void)setVideoOptimizationOptions:(IMVideoOptimizationOptions)options
```

**Parameters***options*

Indicates the characteristics of the video content. Possible values are described in [“IMVideoOptimizationOptions”](#) (page 13).

**Discussion**

Use this method to give hints to the receiver about the type of video content so it can optimize the CPU and bandwidth usage.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [state](#) (page 10)  
- [videoOptimizationOptions](#) (page 11)

**Declared In**

`IMAVManager.h`

**start**

Starts sending audio and video to iChat AV.

```
- (void)start
```

**Discussion**

This method should be called when the receiver's state changes to [IMAVRequested](#) (page 12). If this method is successful, the receiver's state changes to [IMAVRunning](#) (page 13), after possibly changing momentarily to [IMAVStartingUp](#) (page 12) and [IMAVPending](#) (page 12).

Before invoking this method, you need to set the video source using the [setVideoDataSource:](#) (page 8) method to provide video content, or set the number of audio channels to greater than 0 using the [setNumberOfAudioChannels:](#) (page 8) method to provide audio content; otherwise, this method raises an exception.

This method has no effect if invoked when the receiver is not in the [IMAVRequested](#) (page 12) state.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [stop](#) (page 10)

**Declared In**

IMAVManager.h

**state**

Returns the current state of the receiver.

- (IMAVManagerState)state

**Return Value**

The current state of the receiver set by iChat AV. See “[IMAVManagerState](#)” (page 12) for a description of the possible return values.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

IMAVManager.h

**stop**

Stops sending audio and video to iChat AV.

- (void)stop

**Discussion**

After this method is invoked the state changes to [IMAVRequested](#) (page 12), after possibly changing momentarily to [IMAVShuttingDown](#) (page 13).

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [start](#) (page 9)

**Declared In**

IMAVManager.h

**URLToShare**

Returns the file URL of the document that the user chose to share over iChat Theater.

- (NSURL \*)URLToShare

**Return Value**

Returns the file URL of the document or `nil` if the receiver's state is [IMAVInactive](#) (page 12). Also returns `nil` if the user chose this application to share audio/video without a document.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

IMAVManager.h

**videoDataSource**

Returns the receiver's video data source object.

- (id)videoDataSource

**Return Value**

The receiver's video data source, or `nil` if it is not set.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [setVideoDataSource:](#) (page 8)

**Declared In**

IMAVManager.h

**videoOptimizationOptions**

Returns the video optimization options.

- (IMVideoOptimizationOptions)videoOptimizationOptions

**Return Value**

Video optimization options. Possible values are described in "[IMVideoOptimizationOptions](#)" (page 13).

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [setVideoOptimizationOptions:](#) (page 9)

**Declared In**

IMAVManager.h

## Constants

### IMAVManagerState

The state of an IMAVManager object.

```
enum {
    IMAVInactive           = 0,
    IMAVRequested         = 1,
    IMAVShuttingDown      = 2,
    IMAVStartingUp        = 3,
    IMAVPending            = 4,
    IMAVRunning            = 5
};
typedef NSUInteger IMAVManagerState;
```

**Constants****IMAVInactive**

An IMAVManager object is not available to send audio/video to iChat AV because the user has not started a session.

Available in Mac OS X v10.5 and later.

Declared in IMAVManager.h.

**IMAVRequested**

The user selected this client to begin iChat Theater. The client should send [start](#) (page 9) to the IMAVManager object to begin an iChat Theater session.

Available in Mac OS X v10.5 and later.

Declared in IMAVManager.h.

**IMAVStartingUp**

An IMAVManager object is starting up and will soon change to the IMAVPending or IMAVRunning state.

Available in Mac OS X v10.5 and later.

Declared in IMAVManager.h.

**IMAVPending**

iChat AV is not ready to receive content from an IMAVManager object.

An IMAVManager object may enter this state after the [start](#) (page 9) method is invoked when iChat AV is not ready to receive audio/video content. This state may be followed by IMAVRunning at any point.

Typically, this state is entered if either the user does not yet have a video chat active or some internal processing or negotiation needs to take place before auxiliary audio/video input can begin. If the user does not have a video chat active, the state changes to IMAVRunning when a chat starts.

Available in Mac OS X v10.5 and later.

Declared in IMAVManager.h.

#### IMAVRunning

An `IMAVManager` object is actively sending audio/video content to iChat AV.

You should not send audio/video content to an `IMAVManager` object until it reaches this state. For example, do not send audio/video content to an `IMAVManager` object immediately after sending `start` to the manager unless the manager is in this state.

Available in Mac OS X v10.5 and later.

Declared in `IMAVManager.h`.

#### IMAVShuttingDown

An `IMAVManager` object is shutting down and will soon change to the `IMAVInactive` state.

Available in Mac OS X v10.5 and later.

Declared in `IMAVManager.h`.

#### Declared In

`InstantMessage/IMAVManager.h`

## IMVideoOptimizationOptions

The characteristics of the video source to allow for optimization of CPU and bandwidth usage.

```
enum {
    IMVideoOptimizationDefault = 0,
    IMVideoOptimizationStills = 1 << 0,
    IMVideoOptimizationReplacement = 1 << 1,
};
typedef NSUInteger IMVideoOptimizationOptions;
```

#### Constants

##### `IMVideoOptimizationDefault`

Shared video is played alongside the user's local video, and the video is full-motion. This is the default.

Available in Mac OS X v10.5 and later.

Declared in `IMAVManager.h`.

##### `IMVideoOptimizationStills`

Shared video remains unchanged for many sequential frames (such as a photo slideshow). This is a hint that the required bandwidth is lower than that of full-motion video. Incorrectly setting this option may result in poor video quality.

Available in Mac OS X v10.5 and later.

Declared in `IMAVManager.h`.

##### `IMVideoOptimizationReplacement`

Do not send the user's local video, instead devote full CPU and bandwidth resources to the shared video.

Available in Mac OS X v10.5 and later.

Declared in `IMAVManager.h`.

#### Declared In

`InstantMessage/IMAVManager.h`

## Notifications

### **IMAVManagerStateChangedNotification**

Posted by the `IMService` class custom notification center when the iChat AV input state changes.

The notification object is the shared `IMAVManager` object. This notification does not have a user information dictionary. Observers of this notification should send `state` (page 10) to the shared `IMAVManager` object to get the new state.

When the user selects this application or one of its documents to share over iChat Theater, the state of the shared `IMAVManager` object changes to `IMAVRequested` and this notification is sent.

#### **Availability**

Available in Mac OS X v10.5 and later.

#### **Declared In**

`IMAVManager.h`

### **IMAVManagerURLToShareChangedNotification**

Posted by the `IMService` class custom notification center when a new document is selected by the user to share over iChat Theater during a running session.

The notification object is the shared `IMAVManager` object. This notification does not have a user information dictionary. Observers of this notification should send `URLToShare` (page 11) to the shared `IMAVManager` object to get the new document. This notification is not sent the first time the state of the shared `IMAVManager` object changes to `IMAVRequested` to begin the session.

#### **Availability**

Available in Mac OS X v10.5 and later.

#### **Declared In**

`IMAVManager.h`

# Document Revision History

---

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

Date	Notes
2007-10-31	New document that describes the class used to allow clients to share audio and video content through conferences with other buddies in iChat AV.

**REVISION HISTORY**

Document Revision History



# Index

---

## A

---

audioDeviceChannels **instance method** [7](#)  
audioDeviceUID **instance method** [7](#)

## I

---

IMAVInactive **constant** [12](#)  
IMAVManagerState **constant** [12](#)  
IMAVManagerStateChangedNotification **notification** [14](#)  
IMAVManagerURLToShareChangedNotification **notification** [14](#)  
IMAVPending **constant** [12](#)  
IMAVRequested **constant** [12](#)  
IMAVRunning **constant** [13](#)  
IMAVShuttingDown **constant** [13](#)  
IMAVStartingUp **constant** [12](#)  
IMVideoOptimizationDefault **constant** [13](#)  
IMVideoOptimizationOptions **constant** [13](#)  
IMVideoOptimizationReplacement **constant** [13](#)  
IMVideoOptimizationStills **constant** [13](#)

## N

---

numberOfAudioChannels **instance method** [8](#)

## S

---

setNumberOfAudioChannels: **instance method** [8](#)  
setVideoDataSource: **instance method** [8](#)  
setVideoOptimizationOptions: **instance method** [9](#)  
sharedAVManager **class method** [6](#)  
start **instance method** [9](#)  
state **instance method** [10](#)  
stop **instance method** [10](#)

## U

---

URLToShare **instance method** [11](#)

## V

---

videoDataSource **instance method** [11](#)  
videoOptimizationOptions **instance method** [11](#)