
System Sound Services Reference

Audio & Video: Audio



2010-09-01



Apple Inc.
© 2010 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, iPhone, iPod, iPod touch, Mac, and Mac OS are trademarks of Apple Inc., registered in the United States and other countries.

iOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

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

System Sound Services Reference 5

Overview	5
Functions by Task	5
Creating and Disposing of System Sound Objects	5
Playing Sounds	5
Adding and Removing System Sound Callbacks	5
Managing System Sound Services Properties	6
Functions	6
AudioServicesAddSystemSoundCompletion	6
AudioServicesCreateSystemSoundID	7
AudioServicesDisposeSystemSoundID	7
AudioServicesGetProperty	8
AudioServicesGetPropertyInfo	9
AudioServicesPlayAlertSound	10
AudioServicesPlaySystemSound	11
AudioServicesRemoveSystemSoundCompletion	12
AudioServicesSetProperty	12
Callbacks	13
AudioServicesSystemSoundCompletionProc	13
Data Types	14
AudioServicesPropertyID	14
SystemSoundID	14
Constants	14
Alert Sound Identifiers	14
System Sound Services Property Identifiers	15
Result Codes	15

Document Revision History 17

System Sound Services Reference

Framework:	AudioToolbox/AudioToolbox.h
Declared in	AudioServices.h

Overview

System Sound Services provides a C interface for playing short sounds and for invoking vibration on iOS devices that support vibration.

You can use System Sound Services to play short (30 seconds or shorter) sounds. The interface does not provide level, positioning, looping, or timing control, and does not support simultaneous playback: You can play only one sound at a time. You can use System Sound Services to provide audible alerts. On some iOS devices, alerts can include vibration.

For usage information for iOS, refer to “Using Audio” in *Multimedia Programming Guide*.

Functions by Task

Creating and Disposing of System Sound Objects

[AudioServicesCreateSystemSoundID](#) (page 7)

Creates a system sound object.

[AudioServicesDisposeSystemSoundID](#) (page 7)

Disposes of a system sound object and associated resources.

Playing Sounds

[AudioServicesPlayAlertSound](#) (page 10)

Plays a system sound as an alert.

[AudioServicesPlaySystemSound](#) (page 11)

Plays a system sound object.

Adding and Removing System Sound Callbacks

[AudioServicesAddSystemSoundCompletion](#) (page 6)

Registers a callback function that is invoked when a specified system sound finishes playing.

[AudioServicesRemoveSystemSoundCompletion](#) (page 12)

Unregisters any completion callback functions that were registered for a specified system sound.

Managing System Sound Services Properties

[AudioServicesGetPropertyInfo](#) (page 9)

Gets information about a System Sound Services property.

[AudioServicesGetProperty](#) (page 8)

Gets a specified System Sound Services property value.

[AudioServicesSetProperty](#) (page 12)

Sets the value for a specified System Sound Services property.

Functions

AudioServicesAddSystemSoundCompletion

Registers a callback function that is invoked when a specified system sound finishes playing.

```
OSStatus AudioServicesAddSystemSoundCompletion (
    SystemSoundID                inSystemSoundID,
    CFRunLoopRef                 inRunLoop,
    CFStringRef                   inRunLoopMode,
    AudioServicesSystemSoundCompletionProc inCompletionRoutine,
    void                          *inClientData
);
```

Parameters

inSystemSoundID

The system sound that your callback function is to respond to.

inRunLoop

The run loop in which the callback function should run. Pass `NULL` to use the main run loop.

inRunLoopMode

The mode for the run loop in which the callback functions should run. Pass `NULL` to use the default run loop mode.

inCompletionRoutine

The callback function to be invoked when the specified system sound has finished playing.

inClientData

Application data to be passed to your callback function when it is invoked.

Return Value

A result code.

Discussion

Because a system sound may play for several seconds, you might want to know when it has finished playing. For example, you may want to wait until a system sound has finished playing before you play another sound.

Availability

Available in iOS 2.0 and later.

See Also

[AudioServicesCreateSystemSoundID](#) (page 7)

[AudioServicesSystemSoundCompletionProc](#) (page 13)

[AudioServicesRemoveSystemSoundCompletion](#) (page 12)

Declared In

AudioServices.h

AudioServicesCreateSystemSoundID

Creates a system sound object.

```
OSStatus AudioServicesCreateSystemSoundID (
    CFURLRef      inFileURL,
    SystemSoundID *outSystemSoundID
);
```

Parameters

inFileURL

The URL of the audio file to play.

outSystemSoundID

On output, a system sound object associated with the specified audio file.

Return Value

A result code.

Availability

Available in iOS 2.0 and later.

See Also

[AudioServicesDisposeSystemSoundID](#) (page 7)

Related Sample Code

aurioTouch

BubbleLevel

Formulaic

GLPaint

PhotoPicker

Declared In

AudioServices.h

AudioServicesDisposeSystemSoundID

Disposes of a system sound object and associated resources.

```
OSStatus AudioServicesDisposeSystemSoundID (
    SystemSoundID inSystemSoundID
);
```

Parameters

inSystemSoundID

The system sound object to dispose of.

Return Value

A result code.

Availability

Available in iOS 2.0 and later.

See Also

[AudioServicesCreateSystemSoundID](#) (page 7)

Related Sample Code

Audio UI Sounds (SysSound)

BubbleLevel

GLPaint

PhotoPicker

Declared In

AudioServices.h

AudioServicesGetProperty

Gets a specified System Sound Services property value.

```
OSStatus AudioServicesGetProperty (
    AudioServicesPropertyID inPropertyID,
    UInt32 inSpecifierSize,
    const void *inSpecifier,
    UInt32 *ioPropertyDataSize,
    void *outPropertyData
);
```

Parameters

inPropertyID

The property whose value you want.

inSpecifierSize

The size of the buffer pointed to by the *inSpecifier* parameter. Pass 0 if no specifier buffer is required.

inSpecifier

A pointer to a specifier buffer, if such a buffer is required by the property about which you want information. Pass NULL if no specifier is required.

ioPropertyDataSize

On input, the size, in bytes, of the buffer pointed to by the *outPropertyData* parameter. Call the `AudioServicesGetPropertyInfo` function to find out the size required for this buffer. On output, the number of bytes written to the buffer.

outPropertyData

On output, the property value.

Return Value

A result code.

Discussion

System Sound Services properties are listed and described in [“System Sound Services Property Identifiers”](#) (page 15).

Availability

Available in iOS 2.0 and later.

See Also

[AudioServicesGetPropertyInfo](#) (page 9)

[AudioServicesSetProperty](#) (page 12)

Declared In

AudioServices.h

AudioServicesGetPropertyInfo

Gets information about a System Sound Services property.

```
OSStatus AudioServicesGetPropertyInfo (
    AudioServicesPropertyID inPropertyID,
    UInt32                  inSpecifierSize,
    const void              *inSpecifier,
    UInt32                  *outPropertyDataSize,
    Boolean                  *outWritable
);
```

Parameters

inPropertyID

The property you want information about.

inSpecifierSize

The size of the buffer pointed to by the *inSpecifier* parameter. Pass 0 if no specifier buffer is required.

inSpecifier

A pointer to a specifier buffer, if such a buffer is required by the property about which you want information. Pass NULL if no specifier is required.

outPropertyDataSize

On output, the size, in bytes, of the property value. To get the property value, you need a buffer of at least this size.

outWritable

On output, true if the property is writable, or false if the property is read only.

Return Value

A result code.

Discussion

System Sound Services properties are listed and described in [“System Sound Services Property Identifiers”](#) (page 15).

Availability

Available in iOS 2.0 and later.

See Also

[AudioServicesGetProperty](#) (page 8)

Declared In

AudioServices.h

AudioServicesPlayAlertSound

Plays a system sound as an alert.

```
void AudioServicesPlayAlertSound (
    SystemSoundID inSystemSoundID
);
```

Parameters

inSystemSoundID

The system sound object to play as an alert.

Before using this function, call the [AudioServicesCreateSystemSoundID](#) (page 7) function to obtain a system sound.

Discussion

Depending on the particular iOS device, this function plays a short sound and may invoke vibration. Calling this function does the following on various iOS devices:

- **iPhone**—plays the specified sound and, if the user has configured the Settings application for vibration on ring, invokes vibration.
- **iPod touch, original**—plays a short alert melody.
- **iPod touch, 2nd generation**—plays the specified sound.

In iOS, the duration of the sound to be played must not be more than 30 seconds.

Note: System-supplied alert sounds and system-supplied user-interface sound effects are not available to your iOS application. For example, using the `kSystemSoundID_UserPreferredAlert` constant as a parameter to the `AudioServicesPlayAlertSound` function will not play anything.

In Mac OS X, when a user has configured System Preferences to flash the screen for alerts, or if sound cannot be rendered, calling this function will result in the screen flashing. In Mac OS X, pass the constant `kSystemSoundID_UserPreferredAlert` to play the alert sound selected by the user in System Preferences. In iOS there is no preferred user alert sound.

To play a short sound not used as an alert, use `AudioServicesPlaySystemSound`.

Availability

Available in iOS 2.0 and later.

See Also

[AudioServicesCreateSystemSoundID](#) (page 7)

[AudioServicesPlaySystemSound](#) (page 11)

Related Sample Code

Audio UI Sounds (SysSound)

Declared In

AudioServices.h

AudioServicesPlaySystemSound

Plays a system sound object.

```
void AudioServicesPlaySystemSound (
    SystemSoundID inSystemSoundID
);
```

Parameters*inSystemSoundID*

The system sound to play. Before using this function, call the [AudioServicesCreateSystemSoundID](#) (page 7) function to obtain a system sound.

Discussion

This function plays a short sound (30 seconds or less in duration). Because sound might play for several seconds, this function is executed asynchronously. To know when a sound has finished playing, call the [AudioServicesAddSystemSoundCompletion](#) function to register a callback function.

On some iOS devices, you can pass the [kSystemSoundID_Vibrate](#) (page 15) constant to invoke vibration. On other iOS devices, calling this function with that constant does nothing.

Sound files that you play using this function must be:

- No longer than 30 seconds in duration
- In linear PCM or IMA4 (IMA/ADPCM) format
- Packaged in a .caf, .aif, or .wav file

In addition, when you use the [AudioServicesPlaySystemSound](#) function:

- Sounds play at the current system audio volume, with no programmatic volume control available
- Sounds play immediately
- Looping and stereo positioning are unavailable
- Simultaneous playback is unavailable: You can play only one sound at a time

Availability

Available in iOS 2.0 and later.

See Also[AudioServicesCreateSystemSoundID](#) (page 7)[AudioServicesAddSystemSoundCompletion](#) (page 6)[AudioServicesPlayAlertSound](#) (page 10)**Related Sample Code**

Audio UI Sounds (SysSound)

aurioTouch

BubbleLevel
 Formulaic
 PhotoPicker

Declared In

AudioServices.h

AudioServicesRemoveSystemSoundCompletion

Unregisters any completion callback functions that were registered for a specified system sound.

```
void AudioServicesRemoveSystemSoundCompletion (
    SystemSoundID inSystemSoundID
);
```

Parameters

inSystemSoundID

The system sound for which callback functions should be removed.

Availability

Available in iOS 2.0 and later.

See Also

[AudioServicesAddSystemSoundCompletion](#) (page 6)

Declared In

AudioServices.h

AudioServicesSetProperty

Sets the value for a specified System Sound Services property.

```
OSStatus AudioServicesSetProperty (
    AudioServicesPropertyID inPropertyID,
    UInt32 inSpecifierSize,
    const void *inSpecifier,
    UInt32 inPropertyDataSize,
    const void *inPropertyData
);
```

Parameters

inPropertyID

The property whose value you want to set.

inSpecifierSize

The size of the buffer pointed to by the *inSpecifier* parameter. Pass 0 if no specifier buffer is required.

inSpecifier

A pointer to a specifier buffer, if such a buffer is required by the property about which you want information. Pass NULL if no specifier is required.

inPropertyDataSize

The size, in bytes, of the buffer pointed to by the *inPropertyData* parameter.

inPropertyData

The property value you want to set.

Return Value

A result code.

Discussion

System Sound Services properties are listed and described in [“System Sound Services Property Identifiers”](#) (page 15).

Availability

Available in iOS 2.0 and later.

See Also

[AudioServicesGetProperty](#) (page 8)

Declared In

AudioServices.h

Callbacks

AudioServicesSystemSoundCompletionProc

Invoked when a system sound finishes playing.

```
typedef void (*AudioServicesSystemSoundCompletionProc) (
    SystemSoundID ssID,
    void *clientData
);
```

If you named your function `MyAudioServicesSystemSoundCompletionProc`, you would declare it like this:

```
void MyAudioServicesSystemSoundCompletionProc (
    SystemSoundID ssID,
    void *clientData
);
```

Parameters

ssID

The system sound that has finished playing.

clientData

Application data that you specified when registering the callback function.

Discussion

Because a system sound may play for up to 30 seconds, the `AudioServicesPlaySystemSound` function executes asynchronously (that is, it returns immediately). This callback gets invoked when a specified system sound has finished playing. You can use this callback, for example, to help you avoid playing a second sound while a first sound is still playing.

Availability

Available in iOS 2.0 and later.

See Also

[AudioServicesCreateSystemSoundID](#) (page 7)

[AudioServicesAddSystemSoundCompletion](#) (page 6)

[AudioServicesRemoveSystemSoundCompletion](#) (page 12)

[AudioServicesPlaySystemSound](#) (page 11)

Declared In

AudioServices.h

Data Types

AudioServicesPropertyID

The data type for a system sound property identifier.

```
typedef UInt32 AudioServicesPropertyID;
```

Discussion

System Audio Services properties are listed in [“System Sound Services Property Identifiers”](#) (page 15).

Availability

Available in iOS 2.0 and later.

Declared In

AudioServices.h

SystemSoundID

A system sound object, identified with a sound file you want to play.

```
typedef UInt32 SystemSoundID;
```

Discussion

Call the [AudioServicesCreateSystemSoundID](#) (page 7) function to obtain a system sound object.

Availability

Available in iOS 2.0 and later.

Declared In

AudioServices.h

Constants

Alert Sound Identifiers

Identifiers for alert sounds and alternatives to sounds, for use with the [AudioServicesPlayAlertSound](#) (page 10) function.

```
enum {
    kSystemSoundID_Vibrate          = 0x00000FFF
};
```

Constants

`kSystemSoundID_Vibrate`

On the iPhone, use this constant with the [AudioServicesPlayAlertSound](#) (page 10) function to invoke a brief vibration. On the iPod touch, does nothing.

Available in iOS 2.0 and later.

Declared in `AudioServices.h`.

System Sound Services Property Identifiers

Property identifiers used when playing alerts with System Sound Services.

```
enum {
    kAudioServicesPropertyIsUISound          = 'isui',
    kAudioServicesPropertyCompletePlaybackIfAppDies = 'ifdi'
};
```

Constants

`kAudioServicesPropertyIsUISound`

A `UInt32` value, where 1 means that, for the audio file specified by a system sound passed in the `inSpecifier` parameter, the System Sound server respects the user setting in the Sound Effects preference and is silent when the user turns off sound effects.

This property is set to 1 by default. Set it to 0 for the system sound to always play when passed to [AudioServicesPlaySystemSound](#) (page 11), regardless of the user's setting in sound preferences.

Available in iOS 2.0 and later.

Declared in `AudioServices.h`.

`kAudioServicesPropertyCompletePlaybackIfAppDies`

A `UInt32` value, where 1 means that the audio file specified by a system sound passed in the `inSpecifier` parameter should finish playing even if the client application terminates. This could happen, for example, if the user quits or the application terminates unexpectedly while the sound is playing. The default is 0. That is, you must explicitly set this property's value to 1 if you want the sound to complete playing even if the application terminates.

Available in iOS 2.0 and later.

Declared in `AudioServices.h`.

Result Codes

This table lists the result codes defined for System Sound Services.

Result Code	Value	Description
<code>kAudioServicesNoError</code>	0	No error has occurred. Available in iOS 2.0 and later.

Result Code	Value	Description
kAudioServicesUnsupportedPropertyError	'pty?'	The property is not supported. Available in iOS 2.0 and later.
kAudioServicesBadPropertySizeError	'!siz'	The size of the property data was not correct. Available in iOS 2.0 and later.
kAudioServicesBadSpecifierSizeError	'!spc'	The size of the specifier data was not correct. Available in iOS 2.0 and later.
kAudioServicesSystemSoundUnspecifiedError	-1500	An unspecified error has occurred. Available in iOS 2.0 and later.
kAudioServicesSystemSoundClientTimedOutError	-1501	System sound client message timed out. Available in iOS 2.0 and later.

Document Revision History

This table describes the changes to *System Sound Services Reference*.

Date	Notes
2010-09-01	Clarified usage of the AudioServicesPlaySystemSound (page 11) function.
2009-10-05	Minor changes.
2009-02-04	Clarified description of AudioServicesPlayAlertSound (page 10) to describe behavior on various iOS devices.
2008-09-09	New document that describes how to play short sounds and, on iPhone, invoke vibration.

REVISION HISTORY

Document Revision History