NSSound Class Reference

Cocoa > Audio

2008-10-15

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

Simultaneously published in the United States and Canada.

Contents

NSSound Class Reference

Inherits from	NSObject
Conforms to	NSCoding NSCopying NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.0 and later.
Companion guide	Sound Programming Topics for Cocoa
Declared in	NSSound.h
Related sample code	AttachAScript BundleLoader CustomSave TrackIt

Overview

The `NSSound` class provides a simple interface for loading and playing audio files. This class supports the same audio encodings and file formats that are supported by Core Audio and QuickTime.

To use this class, initialize a new instance with the desired file or audio data. You can configure assorted aspects of the audio playback, including the volume and whether the sound loops before you play it. Depending on the type of the audio data, this class may use either Core Audio or QuickTime to handle the actual playback. (Typically, it uses Core Audio to play files in the AIFF, WAVE, NeXT, SD2, AU, and MP3 formats and may use it for other formats in the future as well.) Playback occurs asynchronously so that your application can continue doing work.

You should retain `NSSound` objects before initiating playback or make sure you have a strong reference to them in a garbage-collected environment. Upon deallocation, a sound object stops playback of the sound (as needed) so that it can free up the corresponding audio resources. If you want to deallocate a sound object immediately after playback, assign a delegate and use the `sound:didFinishPlaying:` method to deallocate it.

If you want to play the system beep sound, use the `NSBeep` function.

Adopted Protocols

NSCoding
- `encodeWithCoder:`
- `initWithCoder:`

NSCopying
- `NSCopying`

Tasks

Creating Sounds

`+ canInitWithPasteboard:` (page 8)
Indicates whether the receiver can create an instance of itself from the data in a pasteboard.

`- initWithContentsOfFile:byReference:` (page 11)
Initializes the receiver with the the audio data located at a given filepath.

`- initWithContentsOfURL:byReference:` (page 11)
Initializes the receiver with the audio data located at a given URL.

`- initWithData:` (page 12)
Initializes the receiver with a given audio data.

`- initWithPasteboard:` (page 12)
Initializes the receiver with data from a pasteboard. The pasteboard should contain a type returned by `soundUnfilteredPasteboardTypes` (page 21). `NSSound` expects the data to have a proper magic number, sound header, and data for the formats it supports.

Configuring Sounds

`- name` (page 13)
Returns the name assigned to the receiver.

`- setName:` (page 17)
Registers the receiver under a given name.

`- volume` (page 18)
Provides the volume of the receiver.

`- setVolume:` (page 18)
Specifies the volume of the receiver.

`- currentTime` (page 10)
Provides the receiver's playback progress in seconds.

`- setCurrentTime:` (page 15)
Specifies the receivers playback progress in seconds.

`- loops` (page 13)
Indicates whether the receiver restarts playback when it reaches the end of its content. Default: `NO`.

- `setLoops:` (page 16)
  Specifies whether the receiver restarts playback when it reaches the end of its content.
- `playbackDeviceIdentifier` (page 14)
  Identifies the receiver's output device.
- `setPlaybackDeviceIdentifier:` (page 17)
  Specifies the receiver's output device.
- `channelMapping` (page 9)
  Provides the receiver's channel map.
- `setChannelMapping:` (page 15)
  Specifies the receiver's channel map.
- `delegate` (page 10)
  Returns the receiver's delegate.
- `setDelegate:` (page 16)
  Set the receiver's delegate.

Getting Sound Information

+ `soundUnfilteredTypes` (page 9)
  Provides the file types the `NSSound` class understands.
+ `soundNamed:` (page 8)
  Returns the `NSSound` instance associated with a given name.
- `duration` (page 11)
  Provides the duration of the receiver in seconds.

Playing Sounds

- `isPlaying` (page 13)
  Indicates whether the receiver is playing its audio data.
- `pause` (page 14)
  Pauses audio playback.
- `play` (page 14)
  Initiates audio playback.
- `resume` (page 15)
  Resumes audio playback.
- `stop` (page 18)
  Concludes audio playback.
- `sound:didFinishPlaying:` (page 19) *delegate method*
  This delegate method is called when an `NSSound` instance has completed playback of its sound data.

Writing Sounds

- `writeToPasteboard:` (page 19)
  Writes the receiver's data to a pasteboard.

Deprecated

`+ soundUnfilteredFileTypes` (page 21) Deprecated in Mac OS X v10.5
: Provides the list of file types the `NSSound` class understands. (Deprecated. Use `soundUnfilteredTypes` (page 9).)

`+ soundUnfilteredPasteboardTypes` (page 21) Deprecated in Mac OS X v10.5
: Provides a list of the pasteboard types that the `NSSound` class can accept. (Deprecated. Use `soundUnfilteredTypes` (page 9).)

Class Methods

canInitWithPasteboard:

Indicates whether the receiver can create an instance of itself from the data in a pasteboard.

```
+ (BOOL)canInitWithPasteboard:(NSPasteboard *)pasteboard
```

Parameters

pasteboard
: Pasteboard containing sound data.

Return Value
`YES` when the receiver can handle the data represented by *pasteboard*; `NO` otherwise.

Discussion
The `soundUnfilteredPasteboardTypes` (page 21) method is used to find out whether the class can handle the data in *pasteboard*.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`NSSound.h`

soundNamed:

Returns the `NSSound` instance associated with a given name.

```
+ (id)soundNamed:(NSString *)soundName
```

Parameters

soundName
: Name that identifies sound data.

Return Value
`NSSound` instance initialized with the sound data identified by *soundName*.

Discussion
The returned object can be one of the following:

- One that's been assigned a name with `setName:` (page 17)

- One of the named system sounds provided by the Application Kit framework

If there's no known `NSSound` object with *`soundName`*, this method tries to create one by searching for sound files in the application's main bundle (see `NSBundle` for a description of how the bundle's contents are searched). If no sound file can be located in the application main bundle, the following directories are searched in order:

```
~/Library/Sounds
/Library/Sounds
/Network/Library/Sounds
/System/Library/Sounds
```

If no data can be found for *`soundName`*, no object is created, and `nil` is returned.

The preferred way to locate a sound is to pass a name without the file extension. See the class description for a list of the supported sound file extensions.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
BundleLoader
CustomSave
TrackIt

Declared In
`NSSound.h`

soundUnfilteredTypes

Provides the file types the `NSSound` class understands.

```
+ (NSArray*)soundUnfilteredTypes
```

Return Value
Array of UTIs identifying the file types the `NSSound` class understands.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSSound.h`

Instance Methods

channelMapping

Provides the receiver's channel map.

```
- (NSArray *)channelMapping
```

Return Value
The receiver's audio-channel–to–device–channel mappings.

Discussion
A **channel map** correlates a sound's channels to the the output-device's channels. For example, a two-channel sound being played on a five-channel device should have a channel map to optimize the sound-playing experience. The default map, correlates the first sound channel to the first output channel, the second sound channel to the second output channel, and so on.

For details about channel maps, see *Core Audio Overview* > "An Overview of Common Tasks."

Availability
Available in Mac OS X v10.5 and later.

See Also
`- setChannelMapping:` (page 15)

Declared In
`NSSound.h`

currentTime

Provides the receiver's playback progress in seconds.

```
- (NSTimeInterval)currentTime
```

Return Value
Receiver's playback progress in seconds.

Discussion
Sounds start with `currentTime == 0` and end with `currentTime == ([<sound> duration] - 1)`.

Availability
Available in Mac OS X v10.5 and later.

See Also
`- setCurrentTime:` (page 15)
`- duration` (page 11)

Declared In
`NSSound.h`

delegate

Returns the receiver's delegate.

```
- (id)delegate
```

Return Value
The receiver's delegate.

Availability
Available in Mac OS X v10.0 and later.

See Also
`- setDelegate:` (page 16)

Declared In
`NSSound.h`

duration

Provides the duration of the receiver in seconds.

```
- (NSTimeInterval)duration
```

Return Value
Duration of the receiver in seconds.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSSound.h`

initWithContentsOfFile:byReference:

Initializes the receiver with the the audio data located at a given filepath.

```
- (id)initWithContentsOfFile:(NSString *)filepath byReference:(BOOL)byRef
```

Parameters

filepath
Path to the sound file with which the receiver is to be initialized.

byRef
When `YES` only the name of the sound is stored with the `NSSound` instance when archived using `encodeWithCoder:`; otherwise the audio data is archived along with the instance.

Return Value
Initialized `NSSound` instance.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
AttachAScript

Declared In
`NSSound.h`

initWithContentsOfURL:byReference:

Initializes the receiver with the audio data located at a given URL.

```
- (id)initWithContentsOfURL:(NSURL *)fileUrl byReference:(BOOL)byRef
```

Parameters

fileUrl

URL to the sound file with which the receiver is to be initialized.

byRef

When `YES` only the name of the sound is stored with the `NSSound` instance when archived using `encodeWithCoder:`; otherwise the audio data is archived along with the instance.

Return Value
Initialized `NSSound` instance.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`NSSound.h`

initWithData:

Initializes the receiver with a given audio data.

```
- (id)initWithData:(NSData *)audioData
```

Parameters

audioData

Audio data with which the receiver is to be initialized. The data must have a proper magic number, sound header, and data for the formats the `NSSound` class supports.

Return Value
Initialized `NSSound` instance.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`NSSound.h`

initWithPasteboard:

Initializes the receiver with data from a pasteboard. The pasteboard should contain a type returned by `soundUnfilteredPasteboardTypes` (page 21). `NSSound` expects the data to have a proper magic number, sound header, and data for the formats it supports.

```
- (id)initWithPasteboard:(NSPasteboard *)pasteboard
```

Parameters

pasteboard

The pasteboard containing the audio data with which the receiver is to be initialized. The pasteboard must contain a type returned by `soundUnfilteredPasteboardTypes` (page 21). The contained data must have a proper magic number, sound header, and data for the formats the `NSSound` class supports.

Return Value
Initialized `NSSound` instance.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`NSSound.h`

isPlaying

Indicates whether the receiver is playing its audio data.

```
- (BOOL)isPlaying
```

Return Value
`YES` when the receiver is playing its audio data, `NO` otherwise.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`NSSound.h`

loops

Indicates whether the receiver restarts playback when it reaches the end of its content. Default: `NO`.

```
- (BOOL)loops
```

Return Value
`YES` when the receiver restarts playback when it finishes, `NO` otherwise.

Availability
Available in Mac OS X v10.5 and later.

See Also
`- setLoops:` (page 16)

Declared In
`NSSound.h`

name

Returns the name assigned to the receiver.

```
- (NSString *)name
```

Return Value
Name assigned to the receiver; `nil` when no name has been assigned.

Availability
Available in Mac OS X v10.0 and later.

See Also
- `setName:` (page 17)

Declared In
`NSSound.h`

pause

Pauses audio playback.

```
- (BOOL)pause
```

Return Value
`YES` when playback is paused successfully, `NO` when playback is already paused or when an error occurred.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`NSSound.h`

play

Initiates audio playback.

```
- (BOOL)play
```

Return Value
`YES` when playback is initiated, `NO` when playback is already in progress or when an error occurred.

Discussion
This method initiates playback asynchronously and returns control to your application. Therefore, your application can continue doing work while the audio is playing.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
BundleLoader

Declared In
`NSSound.h`

playbackDeviceIdentifier

Identifies the receiver's output device.

```
- (NSString *)playbackDeviceIdentifier
```

Return Value
Unique identifier of a sound output device.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `setPlaybackDeviceIdentifier:` (page 17)

Declared In
`NSSound.h`

resume

Resumes audio playback.

```
- (BOOL)resume
```

Return Value
`YES` when playback is resumed, `NO` when playback is in progress or when an error occurred.

Discussion
Assumes the receiver has been previously paused by sending it `pause` (page 14).

Availability
Available in Mac OS X v10.0 and later.

Declared In
`NSSound.h`

setChannelMapping:

Specifies the receiver's channel map.

```
- (void)setChannelMapping:(NSArray *)channelMapping
```

Parameters
channelMapping
Audio-channel—to—device–channel mappings for the receiver.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `channelMapping` (page 9)

Declared In
`NSSound.h`

setCurrentTime:

Specifies the receivers playback progress in seconds.

```
- (void)setCurrentTime:(NSTimeInterval)currentTime
```

Parameters

currentTime

Playback progress for the receiver.

Discussion

This property is not archived, copied, or stored on the pasteboard.

Availability

Available in Mac OS X v10.5 and later.

See Also

- currentTime (page 10)

Declared In

NSSound.h

setDelegate:

Set the receiver's delegate.

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

Parameters

delegate

Object to serve as the receiver's delegate.

Availability

Available in Mac OS X v10.0 and later.

See Also

- delegate (page 10)

Declared In

NSSound.h

setLoops:

Specifies whether the receiver restarts playback when it reaches the end of its content.

```
- (void)setLoops:(BOOL)loops
```

Parameters

Term

YES to have the receiver restart playback when it reaches the end of its content.

NO to have the receiver conclude playback, instead.

Discussion

When *loops* is YES, the receiver does not send sound:didFinishPlaying: (page 19) to its delegate when it reaches the end of its content and restarts playback.

Availability

Available in Mac OS X v10.5 and later.

See Also
- `loops` (page 13)
- `stop` (page 18)

Declared In
`NSSound.h`

setName:

Registers the receiver under a given name.

```
- (BOOL)setName:(NSString *)soundName
```

Parameters

soundName
Name to assign the receiver. The name must be unused by other `NSSound` instances.

Return Value
`YES` when successful; `NO` otherwise.

Discussion
If the receiver is already registered under another name, this method first unregisters the prior name.

Availability
Available in Mac OS X v10.0 and later.

See Also
- `name` (page 13)
- `+ soundNamed:` (page 8)

Declared In
`NSSound.h`

setPlaybackDeviceIdentifier:

Specifies the receiver's output device.

```
- (void)setPlaybackDeviceIdentifier:(NSString *)playbackDeviceIdentifier
```

Parameters

playbackDeviceIdentifier
Unique identifier of a sound output device.

Availability
Available in Mac OS X v10.5 and later.

See Also
- `playbackDeviceIdentifier` (page 14)

Declared In
`NSSound.h`

setVolume:

Specifies the volume of the receiver.

```
- (void)setVolume:(float)volume
```

Parameters

volume

Volume at which the receiver is to play.

Discussion

This method does not affect the systemwide volume.

Availability

Available in Mac OS X v10.5 and later.

See Also

- volume (page 18)

Declared In

NSSound.h

stop

Concludes audio playback.

```
- (BOOL)stop
```

Return Value

YES when playback is concluded successfully or if it's paused, NO otherwise.

Availability

Available in Mac OS X v10.0 and later.

See Also

- sound:didFinishPlaying: (page 19)

Declared In

NSSound.h

volume

Provides the volume of the receiver.

```
- (float)volume
```

Return Value

Volume of the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- setVolume: (page 18)

Declared In
NSSound.h

writeToPasteboard:

Writes the receiver's data to a pasteboard.

```
- (void)writeToPasteboard:(NSPasteboard *)pasteboard
```

Parameters

pasteboard
Pasteboard to which the receiver is to write its data.

Availability
Available in Mac OS X v10.0 and later.

Declared In
NSSound.h

Delegate Methods

sound:didFinishPlaying:

This delegate method is called when an NSSound instance has completed playback of its sound data.

```
- (void)sound:(NSSound *)sound didFinishPlaying:(BOOL)finishedPlaying
```

Parameters

sound
The NSSound that has completed playback of its sound data.

finishedPlaying
YES when playback was successful; NO otherwise.

Availability
Available in Mac OS X v10.0 and later.

Declared In
NSSound.h

Constants

NSPasteboard Type for Sound Data

The NSSound class defines this common pasteboard data type.

```
NSString *NSSoundPboardType;
```

Constants

`NSSoundPboardType`

`NSSound` data

Available in Mac OS X v10.0 and later.

Declared in `NSSound.h`.

Declared In

`NSSound.h`

Deprecated NSSound Methods

A method identified as deprecated has been superseded and may become unsupported in the future.

Deprecated in Mac OS X v10.5

soundUnfilteredFileTypes

Provides the list of file types the `NSSound` class understands. (Deprecated in Mac OS X v10.5. Use `soundUnfilteredTypes` (page 9).)

```
+ (NSArray *)soundUnfilteredFileTypes
```

Return Value
Array of strings representing the file types the `NSSound` class understands.

Discussion
The returned array may be passed directly to the `runModalForTypes:` method of the `NSOpenPanel` class.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`NSSound.h`

soundUnfilteredPasteboardTypes

Provides a list of the pasteboard types that the `NSSound` class can accept. (Deprecated in Mac OS X v10.5. Use `soundUnfilteredTypes` (page 9).)

```
+ (NSArray *)soundUnfilteredPasteboardTypes
```

Return Value
Array of pasteboard types that the `NSSound` class can accept.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.5.

Declared In
`NSSound.h`

Document Revision History

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

Date	Notes
2008-10-15	Updated the list of supported sound formats.
2007-04-27	Updated for Mac OS X v10.5.
	Added sound file details to `soundNamed:` (page 8).
	Added detail to methods in "Playing Sounds" (page 7).
	Added search policy and file format information for sound files. Added detail to playback methods.
2006-05-23	First publication of this content as a separate document.

Document Revision History

Index