

[Audio Toolbox](#) / Audio Services

API Collection

Audio Services

Play short sounds or trigger a vibration effect on iOS devices with the appropriate hardware.

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.

Topics

Creating and Disposing of System Sound Objects

```
func AudioServicesCreateSystemSoundID(CFURL, UnsafeMutablePointer<SystemSoundID>) -> OSStatus
```

Creates a system sound object.

```
func AudioServicesDisposeSystemSoundID(SystemSoundID) -> OSStatus
```

Disposes of a system sound object and associated resources.

```
typealias SystemSoundID
```

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

☰ System Sounds

☰ Alert Sound Identifiers

Identifiers for alert sounds and alternatives to sounds, for use with the AudioServicesPlayAlertSound(_:) function.

Playing Sounds

```
func AudioServicesPlayAlertSoundWithCompletion(SystemSoundID, (() -> Void)?)
```

```
func AudioServicesPlaySystemSoundWithCompletion(SystemSoundID, (() -> Void)?)
```

```
func AudioServicesPlayAlertSound(SystemSoundID)
```

Plays a system sound as an alert.

```
func AudioServicesPlaySystemSound(SystemSoundID)
```

Plays a system sound object.

Adding and Removing System Sound Callbacks

```
func AudioServicesAddSystemSoundCompletion(SystemSoundID, CFRunLoop?, CFString?, AudioServicesSystemSoundCompletionProc, UnsafeMutableRaw Pointer?) -> OSStatus
```

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

```
func AudioServicesRemoveSystemSoundCompletion(SystemSoundID)
```

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

```
 typealias AudioServicesSystemSoundCompletionProc
```

A function the system invokes when a system sound finishes playing.

Managing System Sound Services Properties

```
func AudioServicesGetPropertyInfo(AudioServicesPropertyID, UInt32, UnsafeRawPointer?, UnsafeMutablePointer<UInt32>?, UnsafeMutablePointer< DarwinBoolean>?) -> OSStatus
```

Gets information about a System Sound Services property.

```
func AudioServicesGetProperty(AudioServicesPropertyID, UInt32, UnsafeRawPointer?, UnsafeMutablePointer<UInt32>, UnsafeMutableRawPointer?) -> OSStatus
```

Gets a specified System Sound Services property value.

```
func AudioServicesSetProperty(AudioServicesPropertyID, UInt32, UnsafeRawPointer?, UInt32, UnsafeRawPointer) -> OSStatus
```

Sets the value for a specified System Sound Services property.

```
typealias AudioServicesPropertyID
```

The data type for a system sound property identifier.

⌵ System Sound Services Property Identifiers

Property identifiers used when playing alerts with System Sound Services.

⌵ Audio Hardware Services Properties

Property identifiers that apply to HAL audio objects only when accessed via the Audio Hardware Services.

Getting Error Codes

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

⌵ Audio Services Errors

```
var kAudioServicesNoError: OSStatus
```

No error has occurred.

```
var kAudioServicesUnsupportedPropertyError: OSStatus
```

The property is not supported.

```
var kAudioServicesBadPropertySizeError: OSStatus
```

The size of the property data was not correct.

```
var kAudioServicesBadSpecifierSizeError: OSStatus
```

The size of the specifier data was not correct.

```
var kAudioServicesSystemSoundUnspecifiedError: OSStatus
```

An unspecified error has occurred.

```
var kAudioServicesSystemSoundClientTimedOutError: OSStatus
```

System sound client message timed out.

See Also

Playback and Recording



Audio Queue Services

Connect to audio hardware and manage the recording or playback process.



Music Player

Create and play a sequence of tracks, and manage aspects of playback in response to standard events.



Anchoring sound to a window or volume

Provide unique app experiences by attaching sounds to windows and volumes in 3D space.