System Sound Service Reference

사운드 콘텐츠 응용

Dept. of Multimedia Science, Sookmyung Women's University. prof. JongWoo Lee

Index

- Overview
- Functions by Task
  - Creating and Disposing of System Sound Objects
  - Playing Sounds
  - Adding and Removing System Sound Callbacks
  - Managing System Sound Services Properties
- Callbacks
- Data Types
- Constants
- Result Codes

Overview

System Sound Services

- Provides a C interface for playing short sounds
- for invoking vibration on iOS devices that support vibration
- not provide level, positioning, looping or timing control
- not support simultaneous playback
- playing only one sound at a time
- Sound files
  - ✓ No longer than 30 seconds in duration
  - ✓ In linear PCM or IMA4 (IMA/ADPCM) format
  - ✓ Packaged in a .caf, .aif, or .wav file

Functions by Task

Creating and Disposing of System Sound Objects

Creating and Disposing of System Sound Objects

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 2.2 and later.

Creating and Disposing of System Sound Objects

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 2.2 and later.

Functions by Task

Playing Sounds

Playing Sounds

AudioServicesPlaySound

Plays a system sound as an alert

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

- Parameters

inSystemSoundID	• The system sound object to play as an alert. • Call the AudioServicesCreateSystemSoundID function to obtain a system sound (before using this function)

- Discussion

Depending on the particular iOS device, this function plays a short sound and may invoke vibration.

- Availability

Available in iOS 2.0 and later.

Playing Sounds

AudioServicesPlaySystemSound

Playing a system sound object

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

- Parameters

inSystemSoundID	• The system sound to play. • Call the AudioServicesCreateSystemSoundID function to obtain a system sound (before using this function)

- Discussion
  - playing a short sound (30 seconds or less in duration)
  - To know when a sound has finished playing, call the `AudioServicesAddSystemSoundCompletion` function to register a callback function.
  - Sound files
    - ✓ No longer than 30 seconds in duration
    - ✓ In linear PCM or IMA4 (IMA/ADPCM) format
    - ✓ Packaged in a `.caf`, `.aif`, or `.wav` file
  - 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 2.2 and later.

Functions by Task

Adding and Removing System Sound Callbacks

Adding and Removing System Sound Callbacks

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. • NULL : to use the main run loop.
inRunLoopMode	• The mode for the run loop in which the callback functions should run. • 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

- Availability

  Available in iOS 2.2 and later.

Adding and Removing System Sound Callbacks

AudioServicesAddSystemSoundCompletion 사용 예제

```
-(void)playAndRelease{
   AudioServicesAddSystemSoundCompletion (_soundID,NULL,NULL,
      completionCallback,
      (void*) self);
   AudioServicesPlaySystemSound(_soundID);
}

static void completionCallback (SystemSoundID  mySSID, void* myself) {
   //NSLog(@"completion Callback");
   AudioServicesRemoveSystemSoundCompletion (mySSID);

   [(SoundEffect*)myself release];
}
```

Adding and Removing System Sound Callbacks

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 2.2 and later.

Functions by Task

Managing System Sound Services Properties

Managing System Sound Services Properties

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 : writable, or false : read only

- Return Value

  A result code

- Availability

  Available in iOS 2.2 and later.

Managing System Sound Services Properties

AudioServicesGetProperty

Gets a specified System Sound Services property value.

```
OSStatus AudioServicesGetProperty (
       AudioServicesPropertyID inPropertyID,
       UInt32                  inSpecifierSize,
       const void              *inSpecifier,
       UInt32                  *outPropertyDataSize,
       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)
outPropertyDataSize	• 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

- Availability

  Available in iOS 2.2 and later.

Managing System Sound Services Properties

AudioServicesSetProperty

Sets the value for a specified System Sound Services property.

```
OSStatus AudioServicesGetProperty (
      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

- Availability

  Available in iOS 2.2 and later.

Callbacks

Data Types

Constants

Result Codes

Callbacks

AudioServicesSystemSoundCompletionProc

Invoked when a system sound finishes playing.

```
typedef void (*AudioServicesSystemSoundCompletionProc) (
                 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).
  - ex) To help you avoid playing a second sound while a first sound is still playing.
- Availability

  Available in iOS 2.2 and later.

Data Types

AudioServicesPropertyID

The data type for a system sound property identifier.

```
typedef UInt32 AudioServicesPropertyID;
```

SystemSoundID

The data type for a system sound property identifier.

```
typedef UInt32 SystemSoundID;
```

- Availability

  Available in iOS 2.2 and later.

Constants

Alert Sound Identifiers

Identifiers for alert sounds and alternatives to sounds, for use with the AudioServicesAlert Sound function.

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

- Constants

kSystemSoundID_Vibrate	• On the iPhone, use this constant with the AudioServicesPlayAlertSound function to invoke a brief vibration. • On the iPod touch, does nothing.

- Availability

  Available in iOS 2.0 and later.

Constants

System Sound Services Property Identifiers

Property identifiers used when playing alerts with System Sounds 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. • 1 : default • 0 : for the system sound to always play when passed to AudioServicesPlaySystemSound.
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. • 0 : default • 1 : you must explicitly set this property's value if you want the sound to complete playing.

- Availability

  Available in iOS 2.0 and later.

Result Codes

Result Code	Value	Description
`kAudioServicesNoError`	0	No error has occurred. Available in iOS 2.0 and later.
`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.