

[Audio Toolbox](#) / [Audio Components](#)

API Collection

Audio Components

Find, load, and configure audio components, such as Audio Units and audio codecs.

Overview

Use the Audio Components API to register and discover audio units, codecs, and other loadable code modules. This API replaces the Component Manager API used prior to macOS 10.6. The system searches for loadable bundles with a `.audiocomp` or `.component` filename extension in the following locations:

- `~/Library/Audio/Plug-Ins/Components`
- `/Library/Audio/Plug-Ins/Components`
- `/System/Library/Components`

The bundle `Info.plist` file needs to contain an `AudioComponents` item whose value is an array of dictionaries. For example:

```
<key>AudioComponents</key>
<array>
    <dict>
        <key>type</key>
        <string>aufx</string>
        <key>subtype</key>
        <string>XMPL</string>
        <key>manufacturer</key>
        <string>ACME</string>
        <key>name</key>
        <string>AUExample</string>
        <key>version</key>
```

```
<integer>12345</integer>
<key>factoryFunction</key>
<string>AUExampleFactory</string>

<!-- An AudioComponent is sandbox safe. -->

<key>sandboxSafe</key>
<true/>

<!-- Or it can describe its resource usage. -->

<key>resourceUsage</key>
<dict>
    <key>iokit.user-client</key>
    <array>
        <string>CustomUserClient1</string>
        <string>CustomUserClient2</string>
    </array>
    <key>mach-lookup.global-name</key>
    <array>
        <string>MachServiceName1</string>
        <string>MachServiceName2</string>
    </array>
    <key>network.client</key>
    <true/>
    <key>temporary-exception.files.all.read-write</key>
    </true>
</dict>

<!-- An AudioComponent can define its tags. -->

<key>tags</key>
<array>
    <string>Effect</string>
    <string>Equalizer</string>
</array>
</dict>
</array>
```

Topics

Creating an Audio Component Instance

```
func AudioComponentInstanceNew(AudioComponent, UnsafeMutablePointer<AudioComponentInstance?>) -> OSStatus
```

Creates a new instance of an audio component.

```
func AudioComponentInstantiate(AudioComponent, AudioComponentInstantiationOptions, (AudioComponentInstance?, OSStatus) -> Void)
```

```
func AudioComponentInstanceDispose(AudioComponentInstance) -> OSStatus
```

Disposes of an audio component instance.

```
typealias AudioComponent
```

An audio component.

```
struct AudioComponentInstantiationOptions
```

☰ Audio Component Errors

Creating an Audio Component Dynamically

```
func AudioComponentRegister(UnsafePointer<AudioComponentDescription>, CFString, UInt32, AudioComponentFactoryFunction) -> AudioComponent
```

```
func AudioComponentCount(UnsafePointer<AudioComponentDescription>) -> UInt32
```

Returns the number of audio components that match a specified `AudioComponentDescription` structure.

```
func AudioComponentFindNext(AudioComponent?, UnsafePointer<AudioComponentDescription>) -> AudioComponent?
```

Finds the next component that matches a specified `AudioComponentDescription` structure after a specified audio component.

```
func AudioComponentInstanceGetComponent(AudioComponentInstance) -> AudioComponent
```

Retrieves a reference to an audio component from an instance of that audio component.

```
struct AudioComponentDescription
```

Identifying information for an audio component.

```
typealias AudioComponentInstance
```

A component instance, or object, is an audio unit or audio codec.

```
struct AudioComponentFlags
```

```
typealias AudioComponentFactoryFunction
```

Getting Information About a Component

```
func AudioComponentInstanceCanDo(AudioComponentInstance, Int16) -> Bool
```

Determines if an audio component instance implements a particular function.

```
func AudioComponentGetDescription(AudioComponent, UnsafeMutablePointer<AudioComponentDescription>) -> OSStatus
```

Gets the class description, as an `AudioComponentDescription` structure, of an audio component.

```
func AudioComponentCopyName(AudioComponent, UnsafeMutablePointer<Unmanaged<CFString>?>) -> OSStatus
```

Returns the generic name of an audio component.

```
func AudioComponentGetVersion(AudioComponent, UnsafeMutablePointer<UInt32>) -> OSStatus
```

Gets the version of an audio component in hexadecimal form as 0xMMMMmmDD (major, minor, dot).

```
func AudioComponentCopyIcon(AudioComponent) -> UIImage?
```

```
func AudioComponentCopyConfigurationInfo(AudioComponent, UnsafeMutablePointer<Unmanaged<CFDictionary>?>) -> OSStatus
```

```
struct AudioComponentPlugInInterface
```

```
typealias AudioComponentMethod
```

Validating an Audio Component

```
func AudioComponentValidate(AudioComponent, CFDictionary?, UnsafeMutablePointer<AudioComponentValidationResult>) -> OSStatus
```

```
var kAudioComponentValidationParameter_LoadOutOfProcess: String
```

```
enum AudioComponentValidationResult
```

Constants

```
var kAudioComponentConfigurationInfo_ValidationResult: String  
let kAudioComponentInstanceInvalidationNotification: CFString  
let kAudioComponentRegistrationsChangedNotification: CFString  
var kAudioComponentValidationParameter_ForceValidation: String  
var kAudioComponentValidationParameter_TimeOut: String
```

See Also

Audio Units

- { } Generating spatial audio from a multichannel audio stream
 - Convert 8-channel audio to 2-channel spatial audio by using a spatial mixer audio unit.
- :≡ Audio Unit v3 Plug-Ins
 - Deliver custom audio effects, instruments, and other audio behaviors using an Audio Unit v3 app extension.
- :≡ Audio Unit v2 (C) API
 - Configure an Audio Unit and prepare it to render audio.
- :≡ Audio Unit Properties
 - Obtain information about the built-in mixers, equalizers, filters, effects, and other Audio Unit app extensions.
- :≡ Audio Unit Voice I/O
 - Configure system voice processing and respond to speech events.