API Collection

Audio Queue Services

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

Overview

This document describes Audio Queue Services, a C programming interface in the Audio Toolbox framework, which is part of Core Audio.

An audio queue is a software object you use for recording or playing audio. An audio queue does the work of:

- Connecting to audio hardware
- Managing memory
- Employing codecs, as needed, for compressed audio formats
- Mediating playback or recording

Audio Queue Services enables you to record and play audio in linear PCM, in compressed formats (such as Apple Lossless and AAC), and in other formats for which users have installed codecs. Audio Queue Services also supports scheduled playback and synchronization of multiple audio queues and synchronization of audio with video.

> **Note**
>
> Audio Queue Services provides features similar to those previously offered by the Sound Manager and in macOS. It adds additional features such as synchronization. The Sound Manager is deprecated in OS X v10.5 and does not work with 64-bit applications. Audio Queue Services is recommended for all new development and as a replacement for the Sound Manager in existing Mac apps.

Topics

Creating and Disposing of Audio Queues

`func AudioQueueNewOutputWithDispatchQueue(UnsafeMutablePointer<AudioQueueRef?>, UnsafePointer<AudioStreamBasicDescription>, UInt32, dispatch_queue_t, AudioQueueOutputCallbackBlock) -> OSStatus`

`func AudioQueueNewInputWithDispatchQueue(UnsafeMutablePointer<AudioQueueRef?>, UnsafePointer<AudioStreamBasicDescription>, UInt32, dispatch_queue_t, AudioQueueInputCallbackBlock) -> OSStatus`

`func AudioQueueNewOutput(UnsafePointer<AudioStreamBasicDescription>, AudioQueueOutputCallback, UnsafeMutableRawPointer?, CFRunLoop?, CFString?, UInt32, UnsafeMutablePointer<AudioQueueRef?>) -> OSStatus`

Creates a new playback audio queue object.

`func AudioQueueNewInput(UnsafePointer<AudioStreamBasicDescription>, AudioQueueInputCallback, UnsafeMutableRawPointer?, CFRunLoop?, CFString?, UInt32, UnsafeMutablePointer<AudioQueueRef?>) -> OSStatus`

Creates a new recording audio queue object.

`func AudioQueueDispose(AudioQueueRef, Bool) -> OSStatus`

Disposes of an audio queue.

`typealias AudioQueueRef`

Defines an opaque data type that represents an audio queue.

`typealias AudioQueueInputCallbackBlock`

`typealias AudioQueueOutputCallbackBlock`

Controlling Audio Queues

`func AudioQueueStart(AudioQueueRef, UnsafePointer<AudioTimeStamp>?) -> OSStatus`

Begins playing or recording audio.

`func AudioQueuePrime(AudioQueueRef, UInt32, UnsafeMutablePointer<UInt32>?) -> OSStatus`

Decodes enqueued buffers in preparation for playback.

`func AudioQueueFlush(AudioQueueRef) -> OSStatus`

Resets an audio queue's decoder state.

`func AudioQueueStop(AudioQueueRef, Bool) -> OSStatus`

Stops playing or recording audio.

`func AudioQueuePause(AudioQueueRef) -> OSStatus`

Pauses audio playback or recording.

`func AudioQueueReset(AudioQueueRef) -> OSStatus`

Resets an audio queue.

Handling Audio Queue Buffers

`func AudioQueueAllocateBuffer(AudioQueueRef, UInt32, UnsafeMutablePointer<AudioQueueBufferRef?>) -> OSStatus`

Asks an audio queue object to allocate an audio queue buffer.

`func AudioQueueAllocateBufferWithPacketDescriptions(AudioQueueRef, UInt32, UInt32, UnsafeMutablePointer<AudioQueueBufferRef?>) -> OSStatus`

Asks an audio queue object to allocate an audio queue buffer with space for packet descriptions.

`func AudioQueueFreeBuffer(AudioQueueRef, AudioQueueBufferRef) -> OSStatus`

Asks an audio queue to dispose of an audio queue buffer.

`func AudioQueueEnqueueBuffer(AudioQueueRef, AudioQueueBufferRef, UInt32, UnsafePointer<AudioStreamPacketDescription>?) -> OSStatus`

Adds a buffer to the buffer queue of a recording or playback audio queue.

`func AudioQueueEnqueueBufferWithParameters(AudioQueueRef, AudioQueueBufferRef, UInt32, UnsafePointer<AudioStreamPacketDescription>?, UInt32, UInt32, UInt32, UnsafePointer<AudioQueueParameterEvent>?, UnsafePointer<AudioTimeStamp>?, UnsafeMutablePointer<AudioTimeStamp>?) -> OSStatus`

Adds a buffer to the buffer queue of a playback audio queue object, specifying start time and other settings.

Tapping the Queue's Audio

```
func AudioQueueProcessingTapNew(AudioQueueRef, AudioQueueProcessingTap
Callback, UnsafeMutableRawPointer?, AudioQueueProcessingTapFlags,
UnsafeMutablePointer<UInt32>, UnsafeMutablePointer<AudioStreamBasic
Description>, UnsafeMutablePointer<AudioQueueProcessingTapRef?>) ->
OSStatus
```

```
func AudioQueueProcessingTapGetQueueTime(AudioQueueProcessingTapRef,
UnsafeMutablePointer<Float64>, UnsafeMutablePointer<UInt32>) ->
OSStatus
```

```
func AudioQueueProcessingTapGetSourceAudio(AudioQueueProcessingTapRef,
UInt32, UnsafeMutablePointer<AudioTimeStamp>, UnsafeMutablePointer<
AudioQueueProcessingTapFlags>, UnsafeMutablePointer<UInt32>, Unsafe
MutablePointer<AudioBufferList>) -> OSStatus
```

```
func AudioQueueProcessingTapDispose(AudioQueueProcessingTapRef) ->
OSStatus
```

Manipulating Audio Queue Parameters

```
func AudioQueueGetParameter(AudioQueueRef, AudioQueueParameterID,
UnsafeMutablePointer<AudioQueueParameterValue>) -> OSStatus
```

Gets an audio queue parameter value.

```
func AudioQueueSetParameter(AudioQueueRef, AudioQueueParameterID, Audio
QueueParameterValue) -> OSStatus
```

Sets a playback audio queue parameter value.

Manipulating Audio Queue Properties

```
func AudioQueueGetProperty(AudioQueueRef, AudioQueuePropertyID, Unsafe
MutableRawPointer, UnsafeMutablePointer<UInt32>) -> OSStatus
```

Gets an audio queue property value.

```
func AudioQueueSetProperty(AudioQueueRef, AudioQueuePropertyID, Unsafe
RawPointer, UInt32) -> OSStatus
```

Sets an audio queue property value.

```
func AudioQueueGetPropertySize(AudioQueueRef, AudioQueuePropertyID,
UnsafeMutablePointer<UInt32>) -> OSStatus
```

Gets the size of the value of an audio queue property.

`func AudioQueueAddPropertyListener(AudioQueueRef, AudioQueuePropertyID, AudioQueuePropertyListenerProc, UnsafeMutableRawPointer?) -> OSStatus`

Adds a property listener callback to an audio queue.

`func AudioQueueRemovePropertyListener(AudioQueueRef, AudioQueuePropertyID, AudioQueuePropertyListenerProc, UnsafeMutableRawPointer?) -> OSStatus`

Removes a property listener callback from an audio queue.

Managing the Timeline

`func AudioQueueCreateTimeline(AudioQueueRef, UnsafeMutablePointer<AudioQueueTimelineRef?>) -> OSStatus`

Creates a timeline object for an audio queue.

`func AudioQueueDisposeTimeline(AudioQueueRef, AudioQueueTimelineRef) -> OSStatus`

Disposes of an audio queue's timeline object.

`func AudioQueueDeviceGetCurrentTime(AudioQueueRef, UnsafeMutablePointer<AudioTimeStamp>) -> OSStatus`

Gets the current time of the audio hardware device associated with an audio queue.

`func AudioQueueDeviceGetNearestStartTime(AudioQueueRef, UnsafeMutablePointer<AudioTimeStamp>, UInt32) -> OSStatus`

Gets the start time, for an audio hardware device, that is closest to a requested start time.

`func AudioQueueDeviceTranslateTime(AudioQueueRef, UnsafePointer<AudioTimeStamp>, UnsafeMutablePointer<AudioTimeStamp>) -> OSStatus`

Converts the time for an audio queue's associated audio hardware device from one time base representation to another.

`func AudioQueueGetCurrentTime(AudioQueueRef, AudioQueueTimelineRef?, UnsafeMutablePointer<AudioTimeStamp>?, UnsafeMutablePointer<DarwinBoolean>?) -> OSStatus`

Gets the current audio queue time.

`typealias AudioQueueTimelineRef`

Defines an opaque data type that represents an audio queue timeline object.

Performing Offline Rendering

`func AudioQueueSetOfflineRenderFormat(AudioQueueRef, UnsafePointer<AudioStreamBasicDescription>?, UnsafePointer<AudioChannelLayout>?) -> OSStatus`

Sets the rendering mode and audio format for a playback audio queue.

`func AudioQueueOfflineRender(AudioQueueRef, UnsafePointer<AudioTimeStamp>, AudioQueueBufferRef, UInt32) -> OSStatus`

Exports audio to a buffer, instead of to a device, using a playback audio queue.

Callbacks

`typealias AudioQueueInputCallback`

Called by the system when a recording audio queue has finished filling an audio queue buffer.

`typealias AudioQueueOutputCallback`

Called by the system when an audio queue buffer is available for reuse.

`typealias AudioQueuePropertyListenerProc`

Called by the system when a specified audio queue property changes value.

Data Types

`struct AudioQueueChannelAssignment`

`struct AudioQueueProcessingTapFlags`

`struct AudioQueueBuffer`

Defines an audio queue buffer.

`typealias AudioQueueBufferRef`

A pointer to an audio queue buffer.

`struct AudioQueueLevelMeterState`

Specifies the current level metering information for one channel of an audio queue.

`struct AudioQueueParameterEvent`

Specifies an audio queue parameter and associated value.

`typealias AudioQueueParameterID`

A `UInt32` value that uniquely identifies an audio queue parameter.

`typealias AudioQueueParameterValue`

A `Float32` value for an audio queue parameter.

`typealias` `AudioQueueProcessingTapCallback`

`typealias` `AudioQueueProcessingTapRef`

Structures

`struct` `AudioUnitConnection`

An audio unit source-to-destination connection specification.

`struct` `AudioUnitEvent`

`struct` `AudioUnitExternalBuffer`

Allows an audio unit host application to tell an audio unit to use a specified buffer for its input callback.

`struct` `AudioUnitFrequencyResponseBin`

An audio unit's audio level at a particular frequency.

`struct` `AudioUnitMeterClipping`

Audio clipping that has occurred in a mixer unit.

`struct` `AudioUnitMIDIControlMapping`

`struct` `AudioUnitOtherPluginDesc`

`struct` `AudioUnitParameter`

An adjustable audio unit attribute such as volume, pitch, or filter cutoff frequency.

`struct` `AudioUnitParameterEvent`

A scheduled change to an audio unit parameter's value.

`struct` `AudioUnitParameterHistoryInfo`

The suggested update rate and history duration for parameters which have the `flag_PlotHistory` flag set.

`struct` `AudioUnitParameterNameInfo`

A short version of the name for an audio unit parameter.

`typealias` `AudioUnitParameterIDName`

A type definition for a data type that defines the short version of the name for an audio unit parameter.

`struct` `AudioUnitParameterInfo`

`struct` AudioUnitParameterOptions

Value options for audio unit parameters.

`struct` AudioUnitParameterStringFromValue

A string representation of a parameter's value.

`struct` AudioUnitParameterValueFromString

A parameter's value based on a string representation of the value.

`struct` AudioUnitParameterValueName

`struct` AudioUnitParameterValueTranslation

`struct` AudioUnitPresetMAS_SettingData

`struct` AudioUnitPresetMAS_Settings

`struct` AudioUnitProperty

A key-value pair that declares an attribute or behavior for an audio unit.

`struct` AudioUnitRenderActionFlags

Flags for configuring audio unit rendering.

`struct` AU3DMixerRenderingFlags

`struct` AUChannelInfo

The audio input and output channel capabilities for an audio unit.

`struct` AUDependentParameter

An audio unit parameter whose value can change in response to a change in its parent metaparameter.

~~`struct` AUDistanceAttenuationData~~ Deprecated

`struct` AUHostIdentifier

`struct` AUHostTransportStateFlags

`struct` AUHostVersionIdentifier

The name and version of an audio unit's host application.

`struct` AUInputSamplesInOutputCallbackStruct

The callback function and custom data for providing input-to-output sample mapping for an audio unit.

`struct` AUMIDIEvent

A structure that describes a scheduled MIDI event.

`struct` AUMIDIOutputCallbackStruct

The callback function and custom data for an audio unit that provides MIDI output.

`struct` AUNumVersion

`struct` AUParameterAutomationEvent

`struct` AUParameterEvent

A structure that describes a scheduled parameter event.

`struct` AUParameterMIDIMapping

`struct` AUParameterMIDIMappingFlags

`struct` AUPreset

Used to set factory presets for an audio unit.

`struct` AUPresetEvent

Describes an audio unit preset.

`struct` AURecordedParameterEvent

An event recording the changing of a parameter at a particular host time.

`struct` AURenderCallbackStruct

Used for registering an input callback function with an audio unit.

`struct` AURenderEvent

A union of the various specific render event types.

`struct` AURenderEventHeader

The common header for a render event.

`struct` AUSamplerBankPresetData

`struct` AUSamplerInstrumentData

`struct` AUScheduledAudioSliceFlags

`struct` AUSpatialMixerRenderingFlags

Enumerations

`struct` AudioQueueProcessingTapFlags

- Anonymous
- Audio Queue Time Pitch Algorithms
- Audio Queue Property IDs
- Audio Queue Property IDs
- Audio Queue Hardware Codec Policy

Constants

`typealias` `AudioQueuePropertyID`

Identifiers for audio queue properties.

- Audio Queue Parameters

  Identifiers for audio queue parameters.

- Hardware Codec Policy Keys

  Indicates how an audio queue should choose between hardware and software implementations of a codec.

Result Codes

This table lists result codes defined for Audio Queue Services.

`var kAudioQueueErr_InvalidBuffer: OSStatus`

The specified audio queue buffer does not belong to the specified audio queue.

`var kAudioQueueErr_BufferEmpty: OSStatus`

The audio queue buffer is empty (that is, the `mAudioDataByteSize` field = `0`).

`var kAudioQueueErr_DisposalPending: OSStatus`

The function cannot act on the audio queue because it is being asynchronously disposed of.

`var kAudioQueueErr_InvalidProperty: OSStatus`

The specified property ID is invalid.

`var kAudioQueueErr_InvalidPropertySize: OSStatus`

The size of the specified property is invalid.

`var kAudioQueueErr_InvalidParameter: OSStatus`

The specified parameter ID is invalid.

`var kAudioQueueErr_CannotStart: OSStatus`

The audio queue has encountered a problem and cannot start.

`var kAudioQueueErr_InvalidDevice: OSStatus`

The specified audio hardware device could not be located.

`var kAudioQueueErr_BufferInQueue: OSStatus`

The audio queue buffer cannot be disposed of when it is enqueued.

`var kAudioQueueErr_InvalidRunState: OSStatus`

The queue is running but the function can only operate on the queue when it is stopped, or vice versa.

`var kAudioQueueErr_InvalidQueueType: OSStatus`

The queue is an input queue but the function can only operate on an output queue, or vice versa.

`var kAudioQueueErr_Permissions: OSStatus`

You do not have the required permissions to call the function.

`var kAudioQueueErr_InvalidPropertyValue: OSStatus`

The property value used is not valid.

`var kAudioQueueErr_PrimeTimedOut: OSStatus`

During a call to the `AudioQueuePrime(_:_:_:)` function, the audio queue's audio converter failed to convert the requested number of sample frames.

`var kAudioQueueErr_CodecNotFound: OSStatus`

The requested codec was not found.

`var kAudioQueueErr_InvalidCodecAccess: OSStatus`

The codec could not be accessed.

`var kAudioQueueErr_QueueInvalidated: OSStatus`

In iOS, the audio server has exited, causing the audio queue to become invalid.

`var kAudioQueueErr_RecordUnderrun: OSStatus`

During recording, data was lost because there was no enqueued buffer to store it in.

`var kAudioQueueErr_EnqueueDuringReset: OSStatus`

During a call to the `AudioQueueReset(_:)`, `AudioQueueStop(_:_:)`, or `AudioQueueDispose(_:_:)` functions, the system does not allow you to enqueue buffers.

`var kAudioQueueErr_InvalidOfflineMode: OSStatus`

The operation requires the audio queue to be in offline mode but it isn't, or vice versa.

`var kAudioFormatUnsupportedDataFormatError: OSStatus`

The playback data format is unsupported (declared in `AudioFormat.h`).

See Also

Playback and Recording

- Audio Services

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

- 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.