API Collection

Audio File Services

Read or write a variety of audio data to or from disk or a memory buffer.

Overview

This document describes Audio File Services, a C programming interface that enables you to read or write a wide variety of audio data to or from disk or a memory buffer.

With Audio File Services you can:

- Create, initialize, open, and close audio files
- Read and write audio files
- Optimize audio files
- Work with user data and global information

Topics

Creating and Initializing Audio Files

```
func AudioFileCreateWithURL(CFURL, AudioFileTypeID, UnsafePointer<AudioStreamBasicDescription>, AudioFileFlags, UnsafeMutablePointer<AudioFileID?>) -> OSStatus
```

Creates a new audio file, or initializes an existing file, specified by a URL.

```
func AudioFileInitializeWithCallbacks(UnsafeMutableRawPointer, Audio
File_ReadProc, AudioFile_WriteProc, AudioFile_GetSizeProc, AudioFile
_SetSizeProc, AudioFileTypeID, UnsafePointer<AudioStreamBasic
Description>, AudioFileFlags, UnsafeMutablePointer<AudioFileID?>) ->
OSStatus
```

Deletes the content of an existing file and assigns callbacks to the audio file object.

Opening and Closing Audio Files

```
func AudioFileOpenURL(CFURL, AudioFilePermissions, AudioFileTypeID,
UnsafeMutablePointer<AudioFileID?>) -> OSStatus
```

Open an existing audio file specified by a URL.

```
func AudioFileOpenWithCallbacks(UnsafeMutableRawPointer, AudioFile_Read
Proc, AudioFile_WriteProc?, AudioFile_GetSizeProc, AudioFile_SetSize
Proc?, AudioFileTypeID, UnsafeMutablePointer<AudioFileID?>) -> OSStatus
```

Opens an existing file with callbacks you provide.

```
func AudioFileClose(AudioFileID) -> OSStatus
```

Closes an audio file.

Reading and Writing Audio Files

```
func AudioFileReadBytes(AudioFileID, Bool, Int64, UnsafeMutablePointer<
UInt32>, UnsafeMutableRawPointer) -> OSStatus
```

Reads bytes of audio data from an audio file.

```
func AudioFileWriteBytes(AudioFileID, Bool, Int64, UnsafeMutablePointer
<UInt32>, UnsafeRawPointer) -> OSStatus
```

Writes bytes of audio data to an audio file.

```
func AudioFileReadPacketData(AudioFileID, Bool, UnsafeMutablePointer<
UInt32>, UnsafeMutablePointer<AudioStreamPacketDescription>?, Int64,
UnsafeMutablePointer<UInt32>, UnsafeMutableRawPointer?) -> OSStatus
```

Reads packets of audio data from an audio file.

```
func AudioFileWritePackets(AudioFileID, Bool, UInt32, UnsafePointer<
AudioStreamPacketDescription>?, Int64, UnsafeMutablePointer<UInt32>,
UnsafeRawPointer) -> OSStatus
```

Writes packets of audio data to an audio data file.

Getting and Setting Audio File Properties

`func AudioFileGetProperty(AudioFileID, AudioFilePropertyID, UnsafeMutablePointer<UInt32>, UnsafeMutableRawPointer) -> OSStatus`

Gets the value of an audio file property.

`func AudioFileGetPropertyInfo(AudioFileID, AudioFilePropertyID, UnsafeMutablePointer<UInt32>?, UnsafeMutablePointer<UInt32>?) -> OSStatus`

Gets information about an audio file property, including the size of the property value and whether the value is writable.

`func AudioFileSetProperty(AudioFileID, AudioFilePropertyID, UInt32, UnsafeRawPointer) -> OSStatus`

Sets the value of an audio file property

Working with User Data

`func AudioFileCountUserData(AudioFileID, UInt32, UnsafeMutablePointer<UInt32>) -> OSStatus`

Gets the number of user data items with a specified ID in a file.

`func AudioFileGetUserDataSize(AudioFileID, UInt32, UInt32, UnsafeMutablePointer<UInt32>) -> OSStatus`

Gets the size of a user data item in an audio file.

`func AudioFileGetUserDataSize64(AudioFileID, UInt32, UInt32, UnsafeMutablePointer<UInt64>) -> OSStatus`

Gets the size of a user data item in an audio file.

`func AudioFileGetUserData(AudioFileID, UInt32, UInt32, UnsafeMutablePointer<UInt32>, UnsafeMutableRawPointer) -> OSStatus`

Gets a chunk from an audio file.

`func AudioFileGetUserDataAtOffset(AudioFileID, UInt32, UInt32, Int64, UnsafeMutablePointer<UInt32>, UnsafeMutableRawPointer) -> OSStatus`

Gets part of the data from a chunk in an audio file.

`func AudioFileSetUserData(AudioFileID, UInt32, UInt32, UInt32, UnsafeRawPointer) -> OSStatus`

Sets a user data item in an audio file.

`func AudioFileRemoveUserData(AudioFileID, UInt32, UInt32) -> OSStatus`

Removes a user data item from an audio file.

Working with Global Information

`func AudioFileGetGlobalInfoSize(AudioFilePropertyID, UInt32, UnsafeMutableRawPointer?, UnsafeMutablePointer<UInt32>) -> OSStatus`

Gets the size of a global audio file property.

`func AudioFileGetGlobalInfo(AudioFilePropertyID, UInt32, UnsafeMutableRawPointer?, UnsafeMutablePointer<UInt32>, UnsafeMutableRawPointer) -> OSStatus`

Copies the value of a global property into a buffer.

Optimizing Audio Files

`func AudioFileOptimize(AudioFileID) -> OSStatus`

Consolidates audio data and performs other internal optimizations of the file structure.

Parsing Audio File Content

`func NextAudioFileRegion(UnsafePointer<AudioFileRegion>) -> UnsafeMutablePointer<AudioFileRegion>`

Finds the next audio file region in a region list.

`func NumAudioFileMarkersToNumBytes(Int) -> Int`

Returns the number of bytes corresponding to a specified number of audio file markers.

`func NumBytesToNumAudioFileMarkers(Int) -> Int`

A macro that returns the number of audio file markers represented by a specified number of bytes.

Callbacks

`typealias AudioFile_ReadProc`

Reads audio data when used in conjunction with the `AudioFileOpenWithCallbacks(_:_:_:_:_:_:)` or `AudioFileInitializeWithCallbacks(_:_:_:_:_:_:_:_:)` functions.)

`typealias AudioFile_WriteProc`

A callback for writing file data when used in conjunction with the AudioFileOpenWithCallbacks(_:_:_:_:_:_:) or AudioFileCreateWithURL(_:_:_:_:) functions.

typealias AudioFile_GetSizeProc

Gets file data size.

typealias AudioFile_SetSizeProc

Sets file data size.

Data Types

struct AudioBytePacketTranslationFlags

struct AudioFileFlags

struct AudioFileRegionFlags

Flags that specify a playback direction for an audio file region structure.

struct AudioFileStreamParseFlags

struct AudioFileStreamPropertyFlags

struct AudioFileStreamSeekFlags

typealias AudioFileID

An opaque data type that represents an audio file object.

typealias AudioFilePropertyID

An audio file property identifier.

struct AudioFile_SMPTE_Time

A data structure for describing SMPTE (Society of Motion Picture and Television Engineers) time.

struct AudioFileMarker

Annotates a position in an audio file.

struct AudioFileMarkerList

A list of markers associated with an audio file, including their SMPTE time type, the number of markers, and the markers themselves.

struct AudioFileRegion

An audio file region specifies a segment of audio data.

`struct` AudioFileRegionList

A list of the audio file regions in a file.

`struct` AudioFramePacketTranslation

A structure that specifies frame and packet translations.

`struct` AudioBytePacketTranslation

A data structure used by the kAudioFilePropertyByteToPacket and kAudioFilePropertyPacketToByte properties.

`struct` AudioFilePacketTableInfo

Contains information about the number of valid frames in a file and where they begin and end.

`struct` AudioFileTypeAndFormatID

A specifier for the constantkAudioFileGlobalInfo_AvailableStreamDescriptionsForFormat.

`struct` AudioIndependentPacketTranslation

`struct` AudioPacketDependencyInfoTranslation

`struct` AudioPacketRangeByteCountTranslation

`struct` AudioPacketRollDistanceTranslation

Enumerations

`struct` AudioBytePacketTranslationFlags

`struct` AudioFileFlags

`enum` AudioFilePermissions

Flags for use when opening an audio file.

`struct` AudioFileRegionFlags

Flags that specify a playback direction for an audio file region structure.

`struct` AudioFileStreamParseFlags

`struct` AudioFileStreamPropertyFlags

`struct` AudioFileStreamSeekFlags

Constants

`typealias` `AudioFileTypeID`

Operating system constants that indicate the type of file to be written or a hint about what type of file to expect from data provided.

Audio File Creation Flags

Flags to set when creating an audio file.

`enum` `AudioFilePermissions`

Flags for use when opening an audio file.

Audio File Loop Direction Constants

The playback direction of a looped segment of an audio file.

Audio File Marker Types

A type of marker within a file used in the `mType` field of the `AudioFileMarker` structure.

`struct` `AudioFileRegionFlags`

Flags that specify a playback direction for an audio file region structure.

Audio File Packet Translation Flags

Flags specified in a packet translation structure.

Info String Keys

Key values of properties to get and set using Audio File Services functions and provide a common way to get the same information out of several different kinds of files.

Audio File Properties

Properties used by the functions described in getting and setting pieces of data in audio files. See Working with Global Information for details.

Audio File Global Info Properties

Access these properties using the functions described in Working with Global Information.

Result Codes

This table lists the result codes defined for Audio File Services.

`var` `kAudioFileUnspecifiedError: OSStatus`

An unspecified error has occurred.

`var` `kAudioFileUnsupportedFileTypeError: OSStatus`

The file type is not supported.

`var kAudioFileUnsupportedDataFormatError: OSStatus`

The data format is not supported by this file type.

`var kAudioFileUnsupportedPropertyError: OSStatus`

The property is not supported.

`var kAudioFileBadPropertySizeError: OSStatus`

The size of the property data was not correct.

`var kAudioFilePermissionsError: OSStatus`

The operation violated the file permissions. For example, an attempt was made to write to a file opened with the `kAudioFileReadPermission` constant.

`var kAudioFileNotOptimizedError: OSStatus`

The chunks following the audio data chunk are preventing the extension of the audio data chunk. To write more data, you must optimize the file.

`var kAudioFileInvalidChunkError: OSStatus`

Either the chunk does not exist in the file or it is not supported by the file.

`var kAudioFileDoesNotAllow64BitDataSizeError: OSStatus`

The file offset was too large for the file type. The AIFF and WAVE file format types have 32-bit file size limits.

`var kAudioFileInvalidPacketOffsetError: OSStatus`

A packet offset was past the end of the file, or not at the end of the file when a VBR format was written, or a corrupt packet size was read when the packet table was built.

`var kAudioFileInvalidFileError: OSStatus`

The file is malformed, or otherwise not a valid instance of an audio file of its type.

`var kAudioFileOperationNotSupportedError: OSStatus`

The operation cannot be performed.

`var kAudioFileNotOpenError: OSStatus`

The file is closed.

`var kAudioFileEndOfFileError: OSStatus`

End of file.

`var kAudioFilePositionError: OSStatus`

Invalid file position.

```
var kAudioFileFileNotFoundError: OSStatus
```

File not found.

See Also

Audio Files and Formats

- Audio Format Services

  Access information about audio formats and codecs.

- Extended Audio File Services

  Read and write compressed files and linear PCM audio files using a simplified interface.

- Audio File Stream Services

  Parse streamed audio files as the data arrives on the user's computer.

- Audio File Components

  Get information about audio file formats, and about files containing audio data.

- Core Audio File Format

  Parse the structure of Core Audio files.