API Collection

Audio File Stream Services

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

Overview

Audio File Stream Services provides the interface for parsing streamed audio files—in which only a limited window of data is available at a time.

Audio file streams, by nature, are not random access. When you request data from a stream, earlier data might no longer be accessible and later data might not yet be available. In addition, the data you obtain (and then provide to a parser) might include partial packets. To parse streamed audio data, then, a parser must remember data from partially satisfied requests, and must be able to wait for the remainder of that data. In other words, a parser must be able to suspend parsing as needed and then resume where it left off.

To use a parser, you pass data from a streamed audio file, as you acquire it, to the parser. When the parser has a complete packet of audio data or a complete property, it invokes a callback function. Your callbacks then process the parsed data—such as by playing it or writing it to disk.

Here, in outline form, is a typical usage pattern for an audio file stream parser:

1. Create a new audio file stream parser by calling the `AudioFileStreamOpen(_:_:_:_:_:)` function. Pass pointers to your callback functions for audio data and metadata (`AudioFileStream_PacketsProc` and `AudioFileStream_PropertyListenerProc`). The `AudioFileStreamOpen(_:_:_:_:_:)` function gives you a reference to the new parser.

2. Acquire some streamed data. Call the `AudioFileStreamParseBytes(_:_:_:_:)` function when you have data to pass to the parser. Send the data to the parser sequentially and, if possible, without gaps.

3. When the parser acquires a usable buffer of audio data, it invokes your audio data callback. Your callback can then play the data, write it to a file, or otherwise process it.

4. When the parser acquires metadata, it invokes your property callback—which in turn can obtain the property value by calling the `AudioFileStreamGetPropertyInfo(_:_:_:_:)` and `AudioFileStreamGetProperty(_:_:_:_:)` functions.

5. When finished parsing a stream, call the `AudioFileStreamClose(_:)` function to close and deallocate the parser.

Audio File Stream Services supports the following audio data types:

- AIFF
- AIFC
- WAVE
- CAF
- NeXT
- ADTS
- MPEG Audio Layer 3
- AAC

Topics

Opening Audio File Streams

```
func AudioFileStreamOpen(UnsafeMutableRawPointer?, AudioFileStream
_PropertyListenerProc, AudioFileStream_PacketsProc, AudioFileTypeID,
UnsafeMutablePointer<AudioFileStreamID?>) -> OSStatus
```

Creates and opens a new audio file stream parser.

Supplying Data to the Parser

```
func AudioFileStreamParseBytes(AudioFileStreamID, UInt32, UnsafeRaw
Pointer?, AudioFileStreamParseFlags) -> OSStatus
```

Passes audio file stream data to the parser.

Seeking Packets in the Data Stream

```
func AudioFileStreamSeek(AudioFileStreamID, Int64, UnsafeMutablePointer
<Int64>, UnsafeMutablePointer<AudioFileStreamSeekFlags>) -> OSStatus
```

Provides a byte offset for a specified packet in the data stream.

Working with Data Stream Property Information

```
func AudioFileStreamGetPropertyInfo(AudioFileStreamID, AudioFileStream
PropertyID, UnsafeMutablePointer<UInt32>?, UnsafeMutablePointer<Darwin
Boolean>?) -> OSStatus
```

Retrieves information about a property value.

```
func AudioFileStreamGetProperty(AudioFileStreamID, AudioFileStream
PropertyID, UnsafeMutablePointer<UInt32>, UnsafeMutableRawPointer) ->
OSStatus
```

Retrieves the value of the specified property.

```
func AudioFileStreamSetProperty(AudioFileStreamID, AudioFileStream
PropertyID, UInt32, UnsafeRawPointer) -> OSStatus
```

Sets the value of the specified property.

Closing an Audio File Stream

```
func AudioFileStreamClose(AudioFileStreamID) -> OSStatus
```

Closes and deallocates the specified audio file stream parser.

Callbacks

```
typealias AudioFileStream_PropertyListenerProc
```

Invoked by an audio file stream parser when it finds a property value in the audio file stream.

```
typealias AudioFileStream_PacketsProc
```

Invoked by an audio file stream parser when it finds audio data in the audio file stream.

Data Types

```
typealias AudioFileStreamPropertyID
```

Uniquely identifies an audio file stream property.

```
typealias AudioFileStreamID
```

Defines an opaque data type that represents an audio file stream parser.

Enumerations

Audio File Stream Errors

Audio File Types

Constants

Audio File Stream Flags

Flags set by the property listener callback and the `AudioFileStreamParseBytes(_:_:_:_:)` function.

Audio File Stream Properties

Audio file stream properties contain information that you can use to help interpret the audio data in a stream.

Result Codes

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

Audio File Errors

```
var kAudioFileStreamError_UnsupportedFileType: OSStatus
```

The specified file type is not supported.

```
var kAudioFileStreamError_UnsupportedDataFormat: OSStatus
```

The data format is not supported by the specified file type.

```
var kAudioFileStreamError_UnsupportedProperty: OSStatus
```

The property is not supported.

```
var kAudioFileStreamError_BadPropertySize: OSStatus
```

The size of the buffer you provided for property data was not correct.

```
var kAudioFileStreamError_NotOptimized: OSStatus
```

It is not possible to produce output packets because the streamed audio file's packet table or other defining information is not present or appears after the audio data.

```
var kAudioFileStreamError_InvalidPacketOffset: OSStatus
```

A packet offset was less than `0`, or past the end of the file, or a corrupt packet size was read when building the packet table.

`var kAudioFileStreamError_InvalidFile: OSStatus`

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

`var kAudioFileStreamError_ValueUnknown: OSStatus`

The property value is not present in this file before the audio data.

`var kAudioFileStreamError_DataUnavailable: OSStatus`

The amount of data provided to the parser was insufficient to produce any result.

`var kAudioFileStreamError_IllegalOperation: OSStatus`

An illegal operation was attempted.

`var kAudioFileStreamError_UnspecifiedError: OSStatus`

An unspecified error has occurred.

`var kAudioFileStreamError_DiscontinuityCantRecover: OSStatus`

A discontinuity has occurred in the audio data, and Audio File Stream Services cannot recover.

See Also

Audio Files and Formats

- Audio Format Services

  Access information about audio formats and codecs.

- Audio File Services

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

- Extended Audio File Services

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

- 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.