
Audio File Stream Services Reference

[Audio](#) > [Core Audio](#)



2008-10-15



Apple Inc.
© 2008 Apple Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Apple, the Apple logo, Mac, and Mac OS are trademarks of Apple Inc., registered in the United States and other countries.

iPhone is a trademark of Apple Inc.

NeXT is a trademark of NeXT Software, Inc., registered in the United States and other countries.

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE

ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

Audio File Stream Services Reference 5

| | |
|---|----|
| Overview | 5 |
| Functions by Task | 6 |
| Opening Audio File Streams | 6 |
| Supplying Data to the Parser | 6 |
| Seeking Packets in the Data Stream | 6 |
| Working with Data Stream Property Information | 6 |
| Closing an Audio File Stream | 7 |
| Functions | 7 |
| AudioFileStreamClose | 7 |
| AudioFileStreamGetProperty | 7 |
| AudioFileStreamGetPropertyInfo | 8 |
| AudioFileStreamOpen | 9 |
| AudioFileStreamParseBytes | 10 |
| AudioFileStreamSeek | 11 |
| AudioFileStreamSetProperty | 12 |
| Callbacks by Task | 12 |
| Processing Property Values | 12 |
| Handling Packets of Audio File Stream Data | 12 |
| Callbacks | 13 |
| AudioFileStream_PacketsProc | 13 |
| AudioFileStream_PropertyListenerProc | 14 |
| Data Types | 15 |
| AudioFileStreamPropertyID | 15 |
| AudioFileStreamID | 15 |
| Constants | 15 |
| Audio File Stream Flags | 15 |
| Audio File Stream Properties | 16 |
| Result Codes | 19 |

Document Revision History 21

Index 23

Audio File Stream Services Reference

| | |
|--------------------|-----------------------------|
| Framework: | AudioToolbox/AudioToolbox.h |
| Declared in | AudioFileStream.h |

Overview

This document describes Audio File Stream Services, a C programming interface in Core Audio's Audio Toolbox framework.

The Audio File Stream Services API is used for parsing streamed audio files—including cases where only a limited window of the streamed data is available at any time—and returning packets of audio data along with property data (that is, metadata that characterizes the audio data).

In audio files on disk or in memory, any request to read contiguous data always returns all the data requested (as long as the data doesn't reach the end of the file). Thus, the random-access nature of audio files makes parsing straightforward and inexpensive. Audio file streams, on the other hand, are not random access. When a request for data is made, earlier data in the stream might no longer be accessible, and later data might not yet be available. Any request the parser makes for data supplied from the stream might be only partially satisfied. What's more, the data provided might bear no relationship to the boundaries (such as data chunks) in the audio file from which the data was streamed. In order to be able to parse streamed audio file data, therefore, the parser must remember any partially satisfied requests and retry them until all the requested data has been acquired. Only then can it request additional data. Hence, the parser must be able to suspend work at any point and resume parsing where it left off.

Audio File Stream Services solves the parsing problem for audio file streams by remembering the data and necessary state information from previous buffers and not attempting to parse data that is not yet accessible. Audio File Stream Services parses audio file streams to find property data and packets of audio data. You must provide callback functions to process the audio data and the property values. You pass data from a streamed audio file to the Audio File Stream Services parser as you acquire it. When the parser has (at least) either a complete property value or a complete packet of audio data, it calls your callbacks.

Audio File Stream Services provides a straightforward way to read and parse audio file streams:

1. Call the [AudioFileStreamOpen](#) (page 9) function to create a new audio file stream parser. You pass pointers to your callback functions for properties and audio data and the function returns an ID to use in subsequent calls to the parser.
2. Call the [AudioFileStreamParseBytes](#) (page 10) function whenever you have data to pass to the parser. The data should be sent to the parser sequentially and, if possible, without gaps.
 - a. If the parser finds property data, it calls your property callback with a property ID. Your callback can then call the [AudioFileStreamGetPropertyInfo](#) (page 8) and [AudioFileStreamGetProperty](#) (page 7) functions to get the property value.

- b. If the parser finds audio data, it calls your audio data callback with the numbers of bytes and packets and a pointer to the data. Your callback can then write the data to a file or process it as you wish.
3. When you are finished parsing the audio file stream, call the [AudioFileStreamClose](#) (page 7) function to close and deallocate the parser.

Audio File Stream Services currently supports the following audio data types:

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

Functions by Task

Opening Audio File Streams

[AudioFileStreamOpen](#) (page 9)

Creates and opens a new audio file stream parser.

Supplying Data to the Parser

[AudioFileStreamParseBytes](#) (page 10)

Passes audio file stream data to the parser.

Seeking Packets in the Data Stream

[AudioFileStreamSeek](#) (page 11)

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

Working with Data Stream Property Information

[AudioFileStreamGetPropertyInfo](#) (page 8)

Retrieves information about a property value.

[AudioFileStreamGetProperty](#) (page 7)

Retrieves the value of the specified property.

[AudioFileStreamSetProperty](#) (page 12)

Sets the value of the specified property.

Closing an Audio File Stream

[AudioFileStreamClose](#) (page 7)

Closes and deallocates the specified audio file stream parser.

Functions

AudioFileStreamClose

Closes and deallocates the specified audio file stream parser.

```
OSStatus AudioFileStreamClose (
    AudioFileStreamID inAudioFileStream
);
```

Parameters

inAudioFileStream

The ID of the parser you wish to close. The parser ID is returned by the [AudioFileStreamOpen](#) function.

Return Value

A result code. See [“Audio File Stream Result Codes”](#) (page 19).

Availability

Available in Mac OS X v10.5 and later.

See Also

[AudioFileStreamOpen](#) (page 9)

Declared In

AudioFileStream.h

AudioFileStreamGetProperty

Retrieves the value of the specified property.

```
OSStatus AudioFileStreamGetProperty (
    AudioFileStreamID          inAudioFileStream,
    AudioFileStreamPropertyID  inPropertyID,
    UInt32                    *ioPropertyDataSize,
    void                       *outPropertyData
);
```

Parameters

inAudioFileStream

The ID of the parser from which you wish to obtain data. The parser ID is returned by the [AudioFileStreamOpen](#) function.

inPropertyID

A four-character ID indicating the audio file stream property whose value you want to read. See [“Audio File Stream Properties”](#) (page 16) for possible values.

ioPropertyDataSize

On input, the size of the buffer in the `outPropertyData` parameter. Call the `AudioFileStreamGetPropertyInfo` function to obtain the size of the property value. On output, the number of bytes of the property value returned.

outPropertyData

On output, the value of the specified property.

Return Value

A result code. See [“Audio File Stream Result Codes”](#) (page 19).

Availability

Available in Mac OS X v10.5 and later.

See Also

[AudioFileStreamOpen](#) (page 9)

[AudioFileStreamGetPropertyInfo](#) (page 8)

[AudioFileStreamSetProperty](#) (page 12)

Declared In

AudioFileStream.h

AudioFileStreamGetPropertyInfo

Retrieves information about a property value.

```
OSStatus AudioFileStreamGetPropertyInfo (
    AudioFileStreamID          inAudioFileStream,
    AudioFileStreamPropertyID  inPropertyID,
    UInt32                    *outPropertyDataSize,
    Boolean                    *outWritable
);
```

Parameters

inAudioFileStream

The ID of the parser from which you wish to obtain information. The parser ID is returned by the `AudioFileStreamOpen` function.

inPropertyID

A four-character ID indicating the audio file stream property about which you want information. See [“Audio File Stream Properties”](#) (page 16) for possible values.

outPropertyDataSize

On output, the size, in bytes, of the current value of the specified property.

outWritable

On output, `true` if the property can be written. Currently, there are no writable audio file stream properties.

Return Value

A result code. See [“Audio File Stream Result Codes”](#) (page 19).

Availability

Available in Mac OS X v10.5 and later.

See Also

[AudioFileStreamOpen](#) (page 9)

[AudioFileStreamGetProperty](#) (page 7)

Declared In

AudioFileStream.h

AudioFileStreamOpen

Creates and opens a new audio file stream parser.

```
OSStatus AudioFileStreamOpen (
    void
    AudioFileStream_PropertyListenerProc
    AudioFileStream_PacketsProc
    AudioFileTypeID
    AudioFileStreamID
    *inClientData,
    inPropertyListenerProc,
    inPacketsProc,
    inFileTypeHint,
    *outAudioFileStream
);
```

Parameters

inClientData

A pointer to a value or structure to be passed to your callback functions.

inPropertyListenerProc

Your property-listener callback. Whenever the parser finds the value of a property in the data stream, it calls your property listener with the property ID. You can then call the `AudioFileStreamGetPropertyInfo` and `AudioFileStreamGetProperty` functions to get the value of the property.

inPacketsProc

Your audio-data callback. Whenever the parser finds audio data packets in the data stream, it passes the data to your audio-data callback.

inFileTypeHint

An audio file type hint. If the audio file stream that you intend to pass to the parser is of a type that the parser cannot easily or uniquely determine from the data (such as ADTS or AC3), you can use this parameter to indicate the type. Possible values are listed in the `Built-In Audio File Types` enumeration in *Audio File Services Reference*.

If you do not know the audio file type, pass 0.

outAudioFileStream

On output, an opaque object representing the audio file stream parser. This object is referred to in this document as the audio file stream parser ID. You need to pass this ID in to other functions in the Audio File Stream API.

Return Value

A result code. See [“Audio File Stream Result Codes”](#) (page 19).

Availability

Available in Mac OS X v10.5 and later.

See Also

[AudioFileStream_PacketsProc](#) (page 13)

[AudioFileStream_PropertyListenerProc](#) (page 14)

[AudioFileStreamGetPropertyInfo](#) (page 8)

[AudioFileStreamGetProperty](#) (page 7)

Declared In

AudioFileStream.h

AudioFileStreamParseBytes

Passes audio file stream data to the parser.

```
OSStatus AudioFileStreamParseBytes (
    AudioFileStreamID  inAudioFileStream,
    UInt32             inDataByteSize,
    const void         *inData,
    UInt32             inFlags
);
```

Parameters

inAudioFileStream

The ID of the parser to which you wish to pass data. The parser ID is returned by the `AudioFileStreamOpen` function.

inDataByteSize

The number of bytes of data to be parsed.

inData

The data to be parsed.

inFlags

An audio file stream flag. If there is a discontinuity from the last data you passed to the parser, set the `kAudioFileStreamParseFlag_Discontinuity` (page 16) flag.

Return Value

A result code. See [“Audio File Stream Result Codes”](#) (page 19).

Discussion

Streamed audio file data is expected to be passed to the parser in the same sequence in which it appears in the audio file, from the beginning of the audio file stream, without gaps. However, if you called the `AudioFileStreamSeek` function, the parser assumes that the data passed to the `AudioFileStreamParseBytes` function starts from the byte offset returned by the `AudioFileStreamSeek` function.

When you provide data to the parser, the parser looks for property data and audio data packets and, when it has data ready, calls your `AudioFileStream_PropertyListenerProc` and `AudioFileStream_PacketsProc` callback functions to process the data. You should provide at least more than a single packet’s worth of audio file data, but it is better to provide a few packets to a few seconds data at a time.

Availability

Available in Mac OS X v10.5 and later.

See Also

[AudioFileStreamOpen](#) (page 9)

[AudioFileStreamSeek](#) (page 11)

[AudioFileStream_PropertyListenerProc](#) (page 14)

[AudioFileStream_PacketsProc](#) (page 13)

Declared In

AudioFileStream.h

AudioFileStreamSeek

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

```
OSStatus AudioFileStreamSeek (
    AudioFileStreamID  inAudioFileStream,
    SInt64             inAbsolutePacketOffset,
    SInt64             *outAbsoluteByteOffset,
    UInt32             *ioFlags
);
```

Parameters

inAudioFileStream

The ID of the parser to which you wish to provide a byte offset. The parser ID is returned by the `AudioFileStreamOpen` function.

inAbsolutePacketOffset

The number of packets from the beginning of the file of the packet whose byte offset you wish to have returned.

outAbsoluteByteOffset

On output, the absolute byte offset of the packet whose offset you specify in the `inAbsolutePacketOffset` parameter. For audio file formats that do not contain packet tables, the returned offset may be an estimate.

ioFlags

On output, if the `outAbsoluteByteOffset` parameter returns an estimate, this parameter returns the constant `kAudioFileStreamSeekFlag_OffsetIsEstimated`. Currently, no input flags are defined for this call.

Return Value

A result code. See [“Audio File Stream Result Codes”](#) (page 19).

Discussion

After you call this function, the parser assumes the next data passed to the `AudioFileStreamParseBytes` function starts from the byte offset returned in the `outAbsoluteByteOffset` parameter.

Availability

Available in Mac OS X v10.5 and later.

See Also

[AudioFileStreamOpen](#) (page 9)

[AudioFileStreamParseBytes](#) (page 10)

Declared In

AudioFileStream.h

AudioFileStreamSetProperty

Sets the value of the specified property.

```
OSStatus AudioFileStreamSetProperty (
    AudioFileStreamID      inAudioFileStream,
    AudioFileStreamPropertyID inPropertyID,
    UInt32                 inPropertyDataSize,
    const void             *inPropertyData
);
```

Parameters

inAudioFileStream

The ID of the parser to which you wish to pass data. The parser ID is returned by the `AudioFileStreamOpen` function.

inPropertyID

The ID of the audio file stream property whose value is to be set.

inPropertyDataSize

The size, in bytes, of the property data.

inPropertyData

The property data.

Return Value

A result code. See “[Audio File Stream Result Codes](#)” (page 19).

Discussion

Currently, there are no settable properties.

Availability

Available in Mac OS X v10.5 and later.

See Also

[AudioFileStreamOpen](#) (page 9)

[AudioFileStreamGetProperty](#) (page 7)

Declared In

`AudioFileStream.h`

Callbacks by Task

Processing Property Values

[AudioFileStream_PropertyListenerProc](#) (page 14)

A callback function that the parser calls when it finds a property value in the audio file stream.

Handling Packets of Audio File Stream Data

[AudioFileStream_PacketsProc](#) (page 13)

A callback function that the parser calls when it finds audio data in the audio file stream.

Callbacks

AudioFileStream_PacketsProc

A callback function that the parser calls when it finds audio data in the audio file stream.

```
typedef void (*AudioFileStream_PacketsProc) (
    void                *inClientData,
    UInt32              inNumberBytes,
    UInt32              inNumberPackets,
    const void          *inInputData,
    AudioStreamPacketDescription *inPacketDescriptions
);
```

If you named your function `MyAudioFileStream_PacketsProc`, you would declare it like this:

```
void MyAudioFileStream_PacketsProc (
    void                *inClientData,
    UInt32              inNumberBytes,
    UInt32              inNumberPackets,
    const void          *inInputData,
    AudioStreamPacketDescription *inPacketDescriptions
);
```

Parameters

inClientData

The value you provided in the `inClientData` parameter when you called the [AudioFileStreamOpen](#) (page 9) function.

inNumberBytes

The number of bytes of data in the `inInputData` buffer.

inNumberPackets

The number of packets of audio data in the `inInputData` buffer.

inInputData

The audio data.

inPacketDescriptions

An array of audio file stream packet description structures describing the data. Audio file stream packet description structures are described in *Core Audio Data Types Reference*.

Discussion

For constant-bit-rate (CBR) audio data, your callback is typically called with as much data as you passed to the [AudioFileStreamParseBytes](#) (page 10) function. At times, however, only a single packet might be passed because of boundaries in the input data. For variable-bit-rate (VBR) audio data, your callback might be called several times for each time you called the [AudioFileStreamParseBytes](#) (page 10) function.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`AudioFileStream.h`

AudioFileStream_PropertyListenerProc

A callback function that the parser calls when it finds a property value in the audio file stream.

```
typedef void (*AudioFileStream_PropertyListenerProc) (
    void                *inClientData,
    AudioFileStreamID   inAudioFileStream,
    AudioFileStreamPropertyID inPropertyID,
    UInt32              *ioFlags
);
```

If you named your function `MyAudioFileStream_PropertyListenerProc`, you would declare it like this:

```
void MyAudioFileStream_PropertyListenerProc (
    void                *inClientData,
    AudioFileStreamID   inAudioFileStream,
    AudioFileStreamPropertyID inPropertyID,
    UInt32              *ioFlags
);
```

Parameters

inClientData

The value you provided in the `inClientData` parameter when you called the [AudioFileStreamOpen](#) (page 9) function.

inAudioFileStream

The ID of the audio file stream parser that invoked the callback. The parser ID is returned by the `AudioFileStreamOpen` function.

inPropertyID

The four-character ID of the property that the parser found in the audio file data stream. See [“Audio File Stream Properties”](#) (page 16) for possible values.

ioFlags

On input, if the `kAudioFileStreamPropertyFlag_PropertyIsCached` value is set, the parser is caching the property value. If not, on output you can set the `kAudioFileStreamPropertyFlag_CacheProperty` flag to cause the parser to cache the value. See [“Audio File Stream Flags”](#) (page 15).

Discussion

When the parser calls your property listener, check the `ioFlags` value to see if the property value is being cached. If not, you can call the [AudioFileStreamGetPropertyInfo](#) (page 8) and [AudioFileStreamGetProperty](#) (page 7) functions to obtain the value of the property from inside the property listener, or you can set the `kAudioFileStreamPropertyFlag_CacheProperty` flag on return to cause the parser to cache the value.

In some cases when you call the `AudioFileStreamGetProperty` function from inside the property listener, because of boundaries in the input data, the parser returns the result code `"kAudioFileStreamError_DataUnavailable"` indicating the value is not yet available. When unavailable data is requested from within the property listener, the parser begins caching the property value and calls the property listener again when the property value is available. If the `kAudioFileStreamPropertyFlag_PropertyIsCached` flag is not set, this is your only opportunity to get the value of the property, as the data is disposed of when the property listener callback returns.

Availability

Available in Mac OS X v10.5 and later.

Declared In
AudioFileStream.h

Data Types

AudioFileStreamPropertyID

Uniquely identifies an audio file stream property.

```
typedef UInt32 AudioFileStreamPropertyID;
```

Discussion

See “[Audio File Stream Properties](#)” (page 16) for possible values.

Availability

Available in Mac OS X v10.5 and later.

Declared In
AudioFileStream.h

AudioFileStreamID

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

```
typedef struct OpaqueAudioFileStreamID *AudioFileStreamID;
```

Availability

Available in Mac OS X v10.5 and later.

Declared In
AudioFileStream.h

Constants

Audio File Stream Flags

Flags set by the property listener callback and the [AudioFileStreamParseBytes](#) (page 10) function.

```
enum {
    kAudioFileStreamPropertyFlag_PropertyIsCached = 1,
    kAudioFileStreamPropertyFlag_CacheProperty = 2,
    kAudioFileStreamParseFlag_Discontinuity = 1,
    kAudioFileStreamSeekFlag_OffsetIsEstimated = 1
};
```

Constants

`kAudioFileStreamPropertyFlag_PropertyIsCached`

This flag is set when the callback [AudioFileStream_PropertyListenerProc](#) (page 14) is invoked in the case that the value of the property has been cached and can be obtained later.

If this flag is not set, get the value of the property from within this callback or set the [kAudioFileStreamPropertyFlag_CacheProperty](#) (page 16) flag to instruct the parser to begin caching the property data. Otherwise, the value will not be available after the callback returns.

Available in Mac OS X v10.5 and later.

Declared in `AudioFileStream.h`.

`kAudioFileStreamPropertyFlag_CacheProperty`

A property listener sets this flag to instruct the parser to cache the property value so that it remains available after the callback returns.

Available in Mac OS X v10.5 and later.

Declared in `AudioFileStream.h`.

`kAudioFileStreamParseFlag_Discontinuity`

Pass this flag to the [AudioFileStreamParseBytes](#) (page 10) function to signal a discontinuity in the audio data.

Any partial packet straddling a buffer boundary is discarded to avoid having the parser call your callback with a corrupt packet. After a discontinuity occurs, the [AudioFileStreamSeek](#) (page 11) function might return approximate values for some data formats.

Available in Mac OS X v10.5 and later.

Declared in `AudioFileStream.h`.

`kAudioFileStreamSeekFlag_OffsetIsEstimated`

This flag is returned by the [AudioFileStreamSeek](#) (page 11) function if the byte offset is only an estimate.

Available in Mac OS X v10.5 and later.

Declared in `AudioFileStream.h`.

Declared In

`AudioFileStream.h`

Audio File Stream Properties

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


```
enum {
    kAudioFileStreamProperty_ReadyToProducePackets    = 'redy',
    kAudioFileStreamProperty_FileFormat              = 'ffmt',
    kAudioFileStreamProperty_DataFormat              = 'dfmt',
    kAudioFileStreamProperty_FormatList              = 'flst',
    kAudioFileStreamProperty_MagicCookieData         = 'mgic',
    kAudioFileStreamProperty_AudioDataByteCount      = 'bcnt',
    kAudioFileStreamProperty_AudioDataPacketCount    = 'pcnt',
    kAudioFileStreamProperty_MaximumPacketSize       = 'psze',
    kAudioFileStreamProperty_DataOffset              = 'doff',
    kAudioFileStreamProperty_ChannelLayout           = 'cmap',
    kAudioFileStreamProperty_PacketToFrame           = 'pkfr',
    kAudioFileStreamProperty_FrameToPacket           = 'frpk',
    kAudioFileStreamProperty_PacketTableInfo         = 'pnfo',
    kAudioFileStreamProperty_PacketSizeUpperBound    = 'pkub',
    kAudioFileStreamProperty_BitRate                 = 'brat'
};
```

Constants

`kAudioFileStreamProperty_ReadyToProducePackets`

A UInt32 value that is 0 until the parser has parsed up to the beginning of the audio data. Once the parser has reached the audio data, the value of this property is set to 1, at which point all the audio file stream properties that can be known are known.

Available in Mac OS X v10.5 and later.

Declared in `AudioFileStream.h`.

`kAudioFileStreamProperty_FileFormat`

A UInt32 four-character code that identifies the audio data format. For a list of audio format IDs, see “Audio Data Format IDs” in *Core Audio Data Types Reference*.

Available in Mac OS X v10.5 and later.

Declared in `AudioFileStream.h`.

`kAudioFileStreamProperty_DataFormat`

An `AudioStreamBasicDescription` structure describing the format of the audio data in the stream. For more information on audio stream basic descriptions, see *Core Audio Data Types Reference*.

Available in Mac OS X v10.5 and later.

Declared in `AudioFileStream.h`.

`kAudioFileStreamProperty_FormatList`

To support formats such as AAC with SBR where an encoded data stream can be decoded to multiple destination formats, this property returns an array of `AudioFormatListItem` structures (declared in `AudioFormat.h`)—one for each of the destination formats. The default behavior is to return an `AudioFormatListItem` structure that has the same `AudioStreamBasicDescription` structure as that returned by the `kAudioFileStreamProperty_DataFormat` (page 17) property.

Available in Mac OS X v10.5 and later.

Declared in `AudioFileStream.h`.

`kAudioFileStreamProperty_MagicCookieData`

A pointer (`void *`) to a magic cookie. For audio file types that require a magic cookie before packets can be written to a file, you should get this property value before calling the `AudioFileWriteBytes` or `AudioFileWritePackets` functions.

Available in Mac OS X v10.5 and later.

Declared in `AudioFileStream.h`.

`kAudioFileStreamProperty_AudioDataByteCount`

A `UInt64` value indicating the number of bytes of audio data in the streamed file. This property is valid only if the number of bytes for the entire stream is known from the data parsed in the header. For some kinds of streams this property may have no value.

Available in Mac OS X v10.5 and later.

Declared in `AudioFileStream.h`.

`kAudioFileStreamProperty_AudioDataPacketCount`

A `UInt64` value indicating the number of packets of audio data in the streamed file.

Available in Mac OS X v10.5 and later.

Declared in `AudioFileStream.h`.

`kAudioFileStreamProperty_MaximumPacketSize`

A `UInt64` value indicating the maximum packet size of the data in the streamed file.

Available in Mac OS X v10.5 and later.

Declared in `AudioFileStream.h`.

`kAudioFileStreamProperty_DataOffset`

An `SInt64` value indicating the byte offset in the streamed file at which the audio data starts.

Available in Mac OS X v10.5 and later.

Declared in `AudioFileStream.h`.

`kAudioFileStreamProperty_ChannelLayout`

An `AudioChannelLayout` structure. For details, see *Core Audio Data Types Reference*.

Available in Mac OS X v10.5 and later.

Declared in `AudioFileStream.h`.

`kAudioFileStreamProperty_PacketToFrame`

Obtains the frame number corresponding to a packet number. Pass an `AudioFramePacketTranslation` structure with the `mPacket` field filled in, and a value in the `mFrame` field is returned on output. (The `mFrameOffsetInPacket` field of the `AudioFramePacketTranslation` structure is ignored.) For more information on the audio frame packet translation structure, see *Audio File Services Reference*.

Available in Mac OS X v10.5 and later.

Declared in `AudioFileStream.h`.

`kAudioFileStreamProperty_FrameToPacket`

Obtains the packet number corresponding to a frame number. Pass an `AudioFramePacketTranslation` structure with the `mFrame` field filled in, and values in the `mPacket` and `mFrameOffsetInPacket` fields are returned on output. For more information on the audio frame packet translation structure, see *Audio File Services Reference*.

Available in Mac OS X v10.5 and later.

Declared in `AudioFileStream.h`.

`kAudioFileStreamProperty_PacketTableInfo`

An `AudioFilePacketTableInfo` structure. For more information on the audio file packet table info structure, see *Audio File Services Reference*.

Available in Mac OS X v10.5 and later.

Declared in `AudioFileStream.h`.

`kAudioFileStreamProperty_PacketSizeUpperBound`

A `UInt32` value indicating the theoretical maximum packet size in the streamed file. This value is useful for determining minimum buffer sizes, for example.

Available in Mac OS X v10.5 and later.

Declared in `AudioFileStream.h`.

`kAudioFileStreamProperty_BitRate`

A `UInt32` value indicating the bit rate of a stream in bits per second.

Available in Mac OS X v10.5 and later.

Declared in `AudioFileStream.h`.

Discussion

Use these property IDs when calling the `AudioFileStreamGetProperty` function.

Declared In

`AudioStream.h`

Result Codes

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

| Result Code | Value | Description |
|--|--------|---|
| <code>kAudioFileStreamError_UnsupportedFileType</code> | 'typ?' | The specified file type is not supported. Available in Mac OS X v10.5 and later. |
| <code>kAudioFileStreamError_UnsupportedDataFormat</code> | 'fmt?' | The data format is not supported by the specified file type. Available in Mac OS X v10.5 and later. |
| <code>kAudioFileStreamError_UnsupportedProperty</code> | 'pty?' | The property is not supported. Available in Mac OS X v10.5 and later. |
| <code>kAudioFileStreamError_BadPropertySize</code> | 'siz' | The size of the buffer you provided for property data was not correct. Available in Mac OS X v10.5 and later. |
| <code>kAudioFileStreamError_NotOptimized</code> | 'optm' | 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. Available in Mac OS X v10.5 and later. |

| Result Code | Value | Description |
|--|--------|---|
| kAudioFileStreamError_InvalidPacketOffset | 'pck?' | 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. Available in Mac OS X v10.5 and later. |
| kAudioFileStreamError_InvalidFile | 'dta?' | The file is malformed, not a valid instance of an audio file of its type, or not recognized as an audio file. Available in Mac OS X v10.5 and later. |
| kAudioFileStreamError_ValueUnknown | 'unk?' | The property value is not present in this file before the audio data. Available in Mac OS X v10.5 and later. |
| kAudioFileStreamError_DataUnavailable | 'more' | The amount of data provided to the parser was insufficient to produce any result. Available in Mac OS X v10.5 and later. |
| kAudioFileStreamError_IllegalOperation | 'nope' | An illegal operation was attempted. Available in Mac OS X v10.5 and later. |
| kAudioFileStreamError_UnspecifiedError | 'wht?' | An unspecified error has occurred. Available in Mac OS X v10.5 and later. |
| kAudioFileStreamError_DiscontinuityCantRecover | 'dsc!' | A discontinuity has occurred in the audio data, and Audio File Stream Services cannot recover. Available in Mac OS X v10.5 and later. |

Document Revision History

This table describes the changes to *Audio File Stream Services Reference*.

| Date | Notes |
|------------|---|
| 2008-09-09 | Minor corrections. |
| 2008-07-08 | Updated for iPhone OS 2.0. |
| 2007-10-31 | New document that describes a C programming interface for reading non-random-access audio file streams. |

REVISION HISTORY

Document Revision History

Index

A

Audio File Stream Flags [15](#)
Audio File Stream Properties [16](#)
AudioFileStreamClose [function 7](#)
AudioFileStreamGetProperty [function 7](#)
AudioFileStreamGetPropertyInfo [function 8](#)
AudioFileStreamID [data type 15](#)
AudioFileStreamOpen [function 9](#)
AudioFileStreamParseBytes [function 10](#)
AudioFileStreamPropertyID [data type 15](#)
AudioFileStreamSeek [function 11](#)
AudioFileStreamSetProperty [function 12](#)
AudioFileStream_PacketsProc [callback 13](#)
AudioFileStream_PropertyListenerProc [callback 14](#)

K

kAudioFileStreamError_BadPropertySize [constant 19](#)
kAudioFileStreamError_DataUnavailable [constant 20](#)
kAudioFileStreamError_DiscontinuityCantRecover [constant 20](#)
kAudioFileStreamError_IllegalOperation [constant 20](#)
kAudioFileStreamError_InvalidFile [constant 20](#)
kAudioFileStreamError_InvalidPacketOffset [constant 20](#)
kAudioFileStreamError_NotOptimized [constant 19](#)
kAudioFileStreamError_UnspecifiedError [constant 20](#)
kAudioFileStreamError_UnsupportedDataFormat [constant 19](#)
kAudioFileStreamError_UnsupportedFileType [constant 19](#)
kAudioFileStreamError_UnsupportedProperty [constant 19](#)
kAudioFileStreamError_ValueUnknown [constant 20](#)

kAudioFileStreamParseFlag_Discontinuity [constant 16](#)
kAudioFileStreamPropertyFlag_CacheProperty [constant 16](#)
kAudioFileStreamPropertyFlag_PropertyIsCached [constant 16](#)
kAudioFileStreamProperty_AudioDataByteCount [constant 18](#)
kAudioFileStreamProperty_AudioDataPacketCount [constant 18](#)
kAudioFileStreamProperty_BitRate [constant 19](#)
kAudioFileStreamProperty_ChannelLayout [constant 18](#)
kAudioFileStreamProperty_DataFormat [constant 17](#)
kAudioFileStreamProperty_DataOffset [constant 18](#)
kAudioFileStreamProperty_FileFormat [constant 17](#)
kAudioFileStreamProperty_FormatList [constant 17](#)
kAudioFileStreamProperty_FrameToPacket [constant 18](#)
kAudioFileStreamProperty_MagicCookieData [constant 17](#)
kAudioFileStreamProperty_MaximumPacketSize [constant 18](#)
kAudioFileStreamProperty_PacketSizeUpperBound [constant 19](#)
kAudioFileStreamProperty_PacketTableInfo [constant 18](#)
kAudioFileStreamProperty_PacketToFrame [constant 18](#)
kAudioFileStreamProperty_ReadyToProducePackets [constant 17](#)
kAudioFileStreamSeekFlag_OffsetIsEstimated [constant 16](#)