Audio File Stream Services Reference

Audio & Video: Audio

2010-02-24

Contents

Audio File Stream Services Reference

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

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` (page 9) function. Pass pointers to your callback functions for audio data and metadata (`AudioFileStream_PacketsProc` (page 13) and `AudioFileStream_PropertyListenerProc` (page 14)). The `AudioFileStreamOpen` function gives you a reference to the new parser.
2. Acquire some streamed data. Call the `AudioFileStreamParseBytes` (page 10) function when you have data to pass to the parser. Send the data to the parser sequentially and, if possible, without gaps.
   a. 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.
   b. When the parser acquires metadata, it invokes your property callback—which in turn can obtain the property value by calling the `AudioFileStreamGetPropertyInfo` (page 8) and `AudioFileStreamGetProperty` (page 7) functions.
3. When finished parsing a stream, call the `AudioFileStreamClose` (page 7) 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

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 iOS 2.0 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 17) 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 iOS 2.0 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 17) 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 iOS 2.0 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                                 *inClientData,
    AudioFileStream_PropertyListenerProc  inPropertyListenerProc,
    AudioFileStream_PacketsProc           inPacketsProc,
    AudioFileTypeID                       inFileTypeHint,
    AudioFileStreamID                    *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 iOS 2.0 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 iOS 2.0 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 iOS 2.0 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 iOS 2.0 and later.

See Also
`AudioFileStreamOpen` (page 9)
`AudioFileStreamGetProperty` (page 7)

Declared In
`AudioFileStream.h`

Callbacks by Task

Processing Property Values

`AudioFileStream_PropertyListenerProc` (page 14)
Invoked by an audio file stream parser when it finds a property value in the audio file stream.

Handling Packets of Audio File Stream Data

`AudioFileStream_PacketsProc` (page 13)
Invoked by an audio file stream parser when it finds audio data in the audio file stream.

Callbacks

AudioFileStream_PacketsProc

Invoked by an audio file stream parser 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 iOS 2.0 and later.

Declared In

`AudioFileStream.h`

AudioFileStream_PropertyListenerProc

Invoked by an audio file stream parser 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 17) 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 16).

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 iOS 2.0 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 17) for possible values.

Availability

Available in iOS 2.0 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 iOS 2.0 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 iOS 2.0 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 iOS 2.0 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 iOS 2.0 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 iOS 2.0 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_PacketToByte              = 'pkby',
    kAudioFileStreamProperty_ByteToPacket              = 'bypk',
    kAudioFileStreamProperty_PacketTableInfo           = 'pnfo',
    kAudioFileStreamProperty_PacketSizeUpperBound      = 'pkub',
    kAudioFileStreamProperty_AverageBytesPerPacket     = 'abpp',
    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 iOS 2.0 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 iOS 2.0 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 iOS 2.0 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 iOS 2.0 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 iOS 2.0 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 iOS 2.0 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 iOS 2.0 and later.

  Declared in `AudioFileStream.h`.

`kAudioFileStreamProperty_MaximumPacketSize`
: A `UInt32` value indicating the maximum packet size of the data in the streamed file.

  Available in iOS 2.0 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 iOS 2.0 and later.

  Declared in `AudioFileStream.h`.

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

  Available in iOS 2.0 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 iOS 2.0 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 iOS 2.0 and later.

  Declared in `AudioFileStream.h`.

`kAudioFileStreamProperty_PacketToByte`
: Obtains the byte number corresponding to a packet number. Pass an `AudioBytePacketTranslation` structure with the `mPacket` field filled in, and a value is returned in the `mByte` field. The `mByteOffsetInPacket` field of the `AudioBytePacketTranslation` structure is ignored. If the `mByte` value is an estimate, then the `kBytePacketTranslationFlag_IsEstimate` value will be set in the `mFlags` field.

  Available in iOS 2.0 and later.

  Declared in `AudioFileStream.h`.

`kAudioFileStreamProperty_ByteToPacket`
: Obtains the packet number corresponding to a byte number. Pass an `AudioBytePacketTranslation` structure with the `mByte` field filled in, and values are returned in the `mPacket` and `mByteOffsetInPacket` fields. If the `mPacket` value is an estimate, then the `kBytePacketTranslationFlag_IsEstimate` value will be set in the `mFlags` field.

  Available in iOS 2.0 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 iOS 2.0 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 iOS 2.0 and later.

  Declared in `AudioFileStream.h`.

`kAudioFileStreamProperty_AverageBytesPerPacket`
: A `Float64` value indicating the average bytes per packet. For CBR and files with packet tables, this number will be exact. Otherwise, it is a running average of packets parsed.

  Available in iOS 2.0 and later.

  Declared in `AudioFileStream.h`.

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

  Available in iOS 2.0 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
`kAudioFileStreamError_UnsupportedFileType`	'typ?'	The specified file type is not supported. Available in iOS 2.0 and later.
`kAudioFileStreamError_UnsupportedDataFormat`	'fmt?'	The data format is not supported by the specified file type. Available in iOS 2.0 and later.
`kAudioFileStreamError_UnsupportedProperty`	'pty?'	The property is not supported. Available in iOS 2.0 and later.
`kAudioFileStreamError_BadPropertySize`	'!siz'	The size of the buffer you provided for property data was not correct. Available in iOS 2.0 and later.
`kAudioFileStreamError_NotOptimized`	'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 iOS 2.0 and later.
`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 iOS 2.0 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 iOS 2.0 and later.
`kAudioFileStreamError_ValueUnknown`	'unk?'	The property value is not present in this file before the audio data. Available in iOS 2.0 and later.
`kAudioFileStreamError_DataUnavailable`	'more'	The amount of data provided to the parser was insufficient to produce any result. Available in iOS 2.0 and later.
`kAudioFileStreamError_IllegalOperation`	'nope'	An illegal operation was attempted. Available in iOS 2.0 and later.
`kAudioFileStreamError_UnspecifiedError`	'wht?'	An unspecified error has occurred. Available in iOS 2.0 and later.

Result Code	Value	Description
`kAudioFileStreamError_DiscontinuityCantRecover`	'dsc!'	A discontinuity has occurred in the audio data, and Audio File Stream Services cannot recover. Available in iOS 2.0 and later.

Document Revision History

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

Date	Notes
2010-02-24	Corrected description for `kAudioFileStreamProperty_MaximumPacketSize` (page 18).
	Clarified "Introduction" (page 5).
2008-09-09	Minor corrections.
2008-07-08	Updated for iOS 2.0.
2007-10-31	New document that describes a C programming interface for reading non-random-access audio file streams.