22.2. `aifc` — Read and write AIFF and AIFC files

Source code: Lib/aifc.py

This module provides support for reading and writing AIFF and AIFF-C files. AIFF is Audio Interchange File Format, a format for storing digital audio samples in a file. AIFF-C is a newer version of the format that includes the ability to compress the audio data.

Audio files have a number of parameters that describe the audio data. The sampling rate or frame rate is the number of times per second the sound is sampled. The number of channels indicate if the audio is mono, stereo, or quadro. Each frame consists of one sample per channel. The sample size is the size in bytes of each sample. Thus a frame consists of `nchannels * samplesize` bytes, and a second's worth of audio consists of `nchannels * samplesize * framerate` bytes.

For example, CD quality audio has a sample size of two bytes (16 bits), uses two channels (stereo) and has a frame rate of 44,100 frames/second. This gives a frame size of 4 bytes (2*2), and a second's worth occupies 2*2*44100 bytes (176,400 bytes).

Module `aifc` defines the following function:

`aifc.`**`open`**(*file*, *mode=None*)

Open an AIFF or AIFF-C file and return an object instance with methods that are described below. The argument *file* is either a string naming a file or a file object. *mode* must be `'r'` or `'rb'` when the file must be opened for reading, or `'w'` or `'wb'` when the file must be opened for writing. If omitted, `file.mode` is used if it exists, otherwise `'rb'` is used. When used for writing, the file object should be seekable, unless you know ahead of time how many samples you are going to write in total and use `writeframesraw()` and `setnframes()`. The `open()` function may be used in a `with` statement. When the `with` block completes, the `close()` method is called.

Changed in version 3.4: Support for the `with` statement was added.

Objects returned by `open()` when a file is opened for reading have the following methods:

`aifc.`**`getnchannels`**()

Return the number of audio channels (1 for mono, 2 for stereo).

`aifc.`**`getsampwidth`**()

Return the size in bytes of individual samples.

`aifc.`**`getframerate`**()

Return the sampling rate (number of audio frames per second).

`aifc.`**`getnframes`**()

Return the number of audio frames in the file.

`aifc.`**`getcomptype`**()

Return a bytes array of length 4 describing the type of compression used in the audio file. For AIFF files, the returned value is `b'NONE'`.

`aifc.`**`getcompname`**()

Return a bytes array convertible to a human-readable description of the type of compression used in the audio file. For AIFF files, the returned value is `b'not compressed'`.

`aifc.`**`getparams`**()

Returns a `namedtuple()` `(nchannels, sampwidth, framerate, nframes, comptype, compname)`, equivalent to output of the `get*()` methods.

`aifc.`**`getmarkers`**()

Return a list of markers in the audio file. A marker consists of a tuple of three elements. The first is the mark ID (an integer), the second is the mark position in frames from the beginning of the data (an integer), the third is the name of the mark (a string).

`aifc.`**`getmark`**(*id*)

Return the tuple as described in `getmarkers()` for the mark with the given *id*.

`aifc.`**`readframes`**(*nframes*)

Read and return the next *nframes* frames from the audio file. The returned data is a string containing for each frame the uncompressed samples of all channels.

`aifc.`**`rewind`**()

Rewind the read pointer. The next `readframes()` will start from the beginning.

`aifc.`**`setpos`**(*pos*)

Seek to the specified frame number.

`aifc.`**`tell`**()

Return the current frame number.

`aifc.`**`close`**()

Close the AIFF file. After calling this method, the object can no longer be used.

Objects returned by `open()` when a file is opened for writing have all the above methods, except for `readframes()` and `setpos()`. In addition the following methods exist. The `get*()` methods can only be called after the corresponding `set*()` methods have been called. Before the first `writeframes()` or `writeframesraw()`, all parameters except for the number of frames must be filled in.

`aifc.`**`aiff`**()

Create an AIFF file. The default is that an AIFF-C file is created, unless the name of the file ends in `'.aiff'` in which case the default is an AIFF file.

`aifc.`**`aifc`**()

Create an AIFF-C file. The default is that an AIFF-C file is created, unless the name of the file ends in `'.aiff'` in which case the default is an AIFF file.

`aifc.`**`setnchannels`**(*nchannels*)

Specify the number of channels in the audio file.

`aifc.`**`setsampwidth`**(*width*)

Specify the size in bytes of audio samples.

`aifc.`**`setframerate`**(*rate*)

Specify the sampling frequency in frames per second.

`aifc.`**`setnframes`**(*nframes*)

Specify the number of frames that are to be written to the audio file. If this parameter is not set, or not set correctly, the file needs to support seeking.

`aifc.`**`setcomptype`**(*type*, *name*)

Specify the compression type. If not specified, the audio data will not be compressed. In AIFF files, compression is not possible. The name parameter should be a human-readable description of the compression type as a bytes array, the type parameter should be a bytes array of length 4. Currently the following compression types are supported: `b'NONE'`, `b'ULAW'`, `b'ALAW'`, `b'G722'`.

`aifc.`**`setparams`**(*nchannels*, *sampwidth*, *framerate*, *comptype*, *compname*)

Set all the above parameters at once. The argument is a tuple consisting of the various parameters. This means that it is possible to use the result of a `getparams()` call as argument to `setparams()`.

`aifc.`**`setmark`**(*id*, *pos*, *name*)

Add a mark with the given id (larger than 0), and the given name at the given position. This method can be called at any time before `close()`.

`aifc.`**`tell`**()

Return the current write position in the output file. Useful in combination with `setmark()`.

`aifc.`**`writeframes`**(*data*)

Write data to the output file. This method can only be called after the audio file parameters have been set.

Changed in version 3.4: Any bytes-like object is now accepted.

`aifc.`**`writeframesraw`**(*data*)

Like `writeframes()`, except that the header of the audio file is not updated.

Changed in version 3.4: Any bytes-like object is now accepted.

`aifc.`**`close`**()

Close the AIFF file. The header of the file is updated to reflect the actual size of the audio data. After calling this method, the object can no longer be used.