Wavy Documentation

Andrea Celletti

Feb 15, 2019

Contents

CHAPTER 1

Introduction

A pure python module for working with WAVE files with support for all common file formats for both RIFF and RIFX.

When working with WAVE files, there are two main pure python modules available:

- ***builtin.wave*** Python built-in module, lacks support for float and 24bit integer. Provides raw data instead of an array of values.
- ***scipy.wave*** Scipy does not support 24bit integer files. The module strength and weakness is its simplicity, if all you need to do is read and write, this might be for you.

The **wave** module provides a fully featured, dedicated module that can be used as an alternative to the above if flexibility and ease of use are desirable.

CHAPTER 2

Comparison

The following table shows a comparison of supported functionality:

Functionality	builtin.wave	scipy.wave	wavy
RIFF Format Support			
RIFX Format Support			
Read Audio Information			
Read Data As Array			
Read Tag Information			

The following table shows a comparison of supported formats for uncompressed WAVE files:

Sample Width	Format Tag	builtin.wave	scipy.wave	wavy
8 bit	PCM			
	EXTENSIBLE			
16 bit	PCM			
	EXTENSIBLE			
24 bit	PCM			
	EXTENSIBLE			
32 bit	PCM			
	EXTENSIBLE			
	FLOAT			
64 bit	FLOAT			

CHAPTER 3

Installation

The latest stable version is available on PyPI.

Either add `wavy` to your `requirements.txt` file or install with pip:

```
pip install wavy
```

CHAPTER 4

Usage

4.1 Read File

Open a file using the module use `wavy.read`:

```
>>> import wavy
>>> file = wavy.read("audio.wav")
>>> file
WaveFile(sample_width=16, framerate=44100, n_channels=2, n_frames=286653)
```

Get the data for the file:

```
>>> rate, data = file.framerate, file.data

>>> rate
44100

>>> data.shape
(286653, 2)

>>> data.dtype
int16
```

4.2 Get File Info

To read the file information without loading the data use `wavy.info`:

```
>>> wavy.info("audio.wav")
WaveFileInfo(sample_width=16, framerate=44100, n_channels=2, n_frames=286653,␣
↪tags=None)
```

CHAPTER 5

API

5.1 Functions

wavy.**read**(*file*)
Read the the audio file.

Parameters **file** (*str or File*) – Either the path to the file or an instance of File.

Returns An object that represents the file.

Return type *WaveFile*

wavy.**info**(*file*)
Returns information about the audio file.

Parameters **file** (*str or File*) – Either the path to the file or an instance of File.

Returns Information about the file.

Return type WaveFileInfo

5.2 Objects

class wavy.**WaveFile**(*sample_width*, *framerate*, *data*, *tags=None*)
Class that represents a WAVE file.

data
numpy.ndarray – Audio data stored in numpy.ndarray. If the number of channels is one, the array will be one dimensional. Otherwise, the returned array will be two dimensional array of shape (n_frames, n_channels).

framerate
int – Sampling frequency (Hz).

n_channels
int – Number of audio channels.

n_frames
int – Number of audio frames.

sample_width
int – Sample width in bits.

tags
TODO

CHAPTER 6

Introduction

A pure python module for working with WAVE files with support for all common file formats for both RIFF and RIFX.

When working with WAVE files, there are two main pure python modules available:

- ***builtin.wave*** Python built-in module, lacks support for float and 24bit integer. Provides raw data instead of an array of values.
- ***scipy.wave*** Scipy does not support 24bit integer files. The module strength and weakness is its simplicity, if all you need to do is read and write, this might be for you.

The **wave** module provides a fully featured, dedicated module that can be used as an alternative to the above if flexibility and ease of use are desirable.

CHAPTER 7

Comparison

The following table shows a comparison of supported functionality:

Functionality	builtin.wave	scipy.wave	wavy
RIFF Format Support			
RIFX Format Support			
Read Audio Information			
Read Data As Array			
Read Tag Information			

The following table shows a comparison of supported formats for uncompressed WAVE files:

Sample Width	Format Tag	builtin.wave	scipy.wave	wavy
8 bit	PCM			
	EXTENSIBLE			
16 bit	PCM			
	EXTENSIBLE			
24 bit	PCM			
	EXTENSIBLE			
32 bit	PCM			
	EXTENSIBLE			
	FLOAT			
64 bit	FLOAT			

CHAPTER 8

Installation

The latest stable version is available on PyPI.

Either add `wavy` to your `requirements.txt` file or install with pip:

```
pip install wavy
```

CHAPTER 9

Usage

9.1 Read File

Open a file using the module use `wavy.read`:

```
>>> import wavy
>>> file = wavy.read("audio.wav")
>>> file
WaveFile(sample_width=16, framerate=44100, n_channels=2, n_frames=286653)
```

Get the data for the file:

```
>>> rate, data = file.framerate, file.data

>>> rate
44100

>>> data.shape
(286653, 2)

>>> data.dtype
int16
```

9.2 Get File Info

To read the file information without loading the data use `wavy.info`:

```
>>> wavy.info("audio.wav")
WaveFileInfo(sample_width=16, framerate=44100, n_channels=2, n_frames=286653,␣
↪tags=None)
```

Index