
audiodiff Documentation

Release 0.2

Choongmin Lee

February 17, 2014

1	Supported audio formats	3
2	Caveats	5
3	Install	7
4	Dependencies	9
5	API reference	11
6	Indices and tables	13
	Python Module Index	15

audiodiff is a library that can be used to compare audio files. Two audio files are considered equal if they have the same audio streams and normalized tags.

Examples:

```
>>> import audiodiff
>>> audiodiff.equal('airplane.flac', 'airplane.m4a')
False
>>> audiodiff.audio_equal('airplane.flac', 'airplane.m4a')
True
>>> audiodiff.tags_equal('airplane.flac', 'airplane.m4a')
False
```

If you want more, you can get audio checksums and tags:

```
>>> audiodiff.checksum('airplane.flac')
'ffa0d242f8642b20e90f521a898a0ab5'
>>> audiodiff.checksum('airplane.m4a')
'ffa0d242f8642b20e90f521a898a0ab5'
>>> tags1 = audiodiff.tags('airplane.flac')
>>> tags1
{'artist': 'f(x)', 'album': 'Pink Tape', 'title': 'Airplane'}
>>> tags2 = audiodiff.tags('airplane.m4a')
>>> tags2
{'title': 'f(x) - Pink Tape - Airplane'}
```

It can be also used as a commandline tool. When used as a commandline tool, it supports comparing audio files in two directories recursively. Audio files with the same name except for the extensions are considered to be compared.

Commandline examples:

```
$ ls -R
mylib1:
a.flac  b.flac  cover.jpg

mylib2:
a.m4a  b.m4a  cover.jpg
$ audiodiff mylib1 mylib2
Audio streams in mylib1/a.flac and mylib2/a.m4a differ
Audio streams in mylib1/b.flac and mylib2/b.m4a differ
--- mylib1/b.flac
+++ mylib2/b.m4a
-album: [u'Purple Heart']
+album: [u'Blue Jean']
+date: [u'2001']
Binary files mylib1/cover.jpg and mylib2/cover.jpg differ
```

Supported audio formats

Currently audiodiff can only read FLAC, M4A, MP3 files. They must have *flac*, *m4a*, *mp3* file extensions respectively.

Caveats

Tag reading is done by [mutagenwrapper](#) for which there isn't a stable version yet. It may omit some tags, thus incorrectly reporting tags in files being compared are equal while they are not.

Install

audiodiff can be installed with *pip*. To install, run:

```
pip install audiodiff
```

For help using the commandline tool, run `audiodiff -h`.

Dependencies

audiodiff requires *ffmpeg* to be installed in your system. The path is `ffmpeg` by default, but you can change it by following ways (later rules take precedence over earlier ones):

1. `audiodiff.FFMPEG_BIN` module property
2. `FFMPEG_BIN` environment variable
3. `--ffmpeg_bin` flag (commandline tool only)

API reference

`audiodiff.AUDIO_FORMATS = ['flac', 'm4a', 'mp3']`

Supported audio formats

`audiodiff.FFMPEG_BIN = 'ffmpeg'`

Default *ffmpeg* path

`audiodiff.audio_equal(name1, name2, ffmpeg_bin=None)`

Compares two audio files and returns True if they have the same audio streams.

`audiodiff.checksum(name, ffmpeg_bin=None)`

Returns an MD5 checksum of the uncompressed WAVE data stream of the audio file.

`audiodiff.equal(name1, name2, ffmpeg_bin=None)`

Compares two files and returns True if they are considered equal. For audio files, they are equal if their uncompressed audio streams and tags (reported by *mutagenwrapper*, except for *encodedby* which is ignored) are equal. Otherwise, two files must have the same content to be equal.

`audiodiff.ffmpeg_path()`

Returns the path to *ffmpeg* binary.

`audiodiff.is_supported_format(name)`

Returns True if the name has an extension that is one of the supported formats.

`audiodiff.tags(name)`

Returns tags in the audio file as *dict*. It converts tags returned by *mutagenwrapper.read_tags* by unwrapping single valued items (i.e. without enclosing lists) and removing *encodedby* tag. To read unmodified, but still normalized tags, use *mutagenwrapper.read_tags*. For unmodified and unnormalized tags, use the *mutagen* library.

`audiodiff.tags_equal(name1, name2)`

Compares two audio files and returns True if they have the same tags reported by *mutagenwrapper*. It ignores *encodedby* tag.

Indices and tables

- *genindex*
- *modindex*
- *search*

a

audiodiff, [11](#)