

---

# **audiodiff Documentation**

***Release 0.3.0***

**Choongmin Lee**

February 17, 2014



<b>1</b>	<b>Dependencies</b>	<b>3</b>
<b>2</b>	<b>Install</b>	<b>5</b>
<b>3</b>	<b>Examples</b>	<b>7</b>
<b>4</b>	<b>Supported audio formats</b>	<b>9</b>
<b>5</b>	<b>Caveats</b>	<b>11</b>
<b>6</b>	<b>Changes</b>	<b>13</b>
6.1	Version 0.3 . . . . .	13
6.2	Version 0.2 . . . . .	13
<b>7</b>	<b>API reference</b>	<b>15</b>
7.1	audiodiff . . . . .	15
7.2	audiodiff.commandlinetool . . . . .	16
<b>8</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>



audiodiff is a small Python library for comparing audio files. Two audio files are considered equal if they have the same audio streams and normalized tags.



---

## 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)

You can install `ffmpeg` with following commands.

- Debian/Ubuntu: `sudo apt-get install ffmpeg`
- OS X (with [Homebrew](#)): `brew install ffmpeg`





---

### Install

---

audiodiff can be installed with `pip`:

```
$ pip install audiodiff
```

This will also install the commandline tool. Run `audiodiff -h` for help.



---

## Examples

---

Suppose you have two files, `airplane.flac` and `airplane.m4a`. The second one is obtained by converting the first one with an ALAC encoder, so its audio stream should be identical with the first one's. After the conversion, you changed the tags in the FLAC file. Then you may get the following results with `audiodiff`:

```
>>> 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
```

It means the two files are not the same because the tags differ, but the audio streams are identical.

If you want more information about those files, you can get stream checksums and tags:

```
>>> audiodiff.checksum('airplane.flac')
'ed871b3c164998cf243e39d4b97d21f93bba9427'
>>> audiodiff.checksum('airplane.m4a')
'ed871b3c164998cf243e39d4b97d21f93bba9427'
>>> 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 also be 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 extension are compared to each other.

```
$ 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 recognizes only WAV, FLAC, M4A, and MP3 files as audiofiles. They must have `wav`, `flac`, `m4a`, `mp3` file extensions, respectively. Note that WAV files are assumed to have no tags, because tagging WAV files are inconsistent among many applications.



---

### 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.





---

## Changes

---

### 6.1 Version 0.3

(release date to be announced)

- Improved Unicode support for tags and filenames.
- Change the stream checksum algorithm from MD5 to SHA1.
- Support Python 2.6 and PyPy, in addition to Python 2.7.

### 6.2 Version 0.2

Initial release on September 10th 2013.



---

## API reference

---

### 7.1 audiodiff

This module contains functions for comparing audio files.

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

Supported audio formats (extensions)

`audiodiff.FFMPEG_BIN = 'ffmpeg'`

Default FFmpeg path

`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 (as reported by `mutagenwrapper`, except for `encodedby` which is ignored) are equal. For non-audio files, they must have the same content to be equal.

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

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

`audiodiff.tags_equal(name1, name2)`

Compares two audio files and returns `True` if they have the same tags reported by `mutagenwrapper`.

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

Returns an SHA1 checksum of the uncompressed PCM (signed 24-bit little-endian) data stream of the audio file. Note that the checksums for the same file may differ across different platforms if the file format is lossy, due to floating point problems and different implementations of decoders.

`audiodiff.tags(name)`

Returns tags in the audio file as a `dict`. Its return value is the same as `mutagenwrapper.read_tags`, except that single valued items (lists with length 1) are unwrapped and `encodedby` tag is removed. To read unmodified, but still normalized tags, use `mutagenwrapper.read_tags`. For raw tags, use the `mutagen` library.

`audiodiff.get_extension(path)`

Returns the file extension of the specified path. Example:

```
>>> get_extension('a.pdf')
'pdf'
>>> get_extension('b.js.coffee')
'coffee'
>>> get_extension('c')
''
>>> get_extension('d/e.txt')
'txt'
```

`audiodiff.is_supported_format(path)`

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

`audiodiff.ffmpeg_path()`

Returns the path to FFmpeg binary.

**exception** `audiodiff.AudiodiffException`

The root class of all audiodiff-related exceptions.

**exception** `audiodiff.UnsupportedFileError`

Raised when you pass a non-audio file to a function that expects audio files.

**exception** `audiodiff.ExternalLibraryError`

Raised when there is an error during running FFmpeg.

## 7.2 audiodiff.commandlinetool

This module contains functions for the `audiodiff` commandline tool.

`audiodiff.commandlinetool.FALLBACK_ENCODING = 'UTF-8'`

Fallback encoding for output. Encoding resolution is done as follows:

- `sys.stdout.encoding` (or `sys.stderr.encoding`)
- `PYTHONIOENCODING` environment variable
- Second item of the return value of `locale.getdefaultlocale()`
- `FALLBACK_ENCODING`

`audiodiff.commandlinetool.parser = ArgumentParser(prog='audiodiff', usage=None, description='\nCompare two fi`  
 An `argparse.ArgumentParser`

`audiodiff.commandlinetool.main_func(args=None)`

The entry point for the `audiodiff` command line tool. Parses the command arguments and calls `diff_checked()`.

`audiodiff.commandlinetool.diff_checked(path1, path2, options)`

Calls `diff_recurse()` and handles exceptions if raised.

`audiodiff.commandlinetool.diff_recurse(path1, path2, options)`

Recursively compares files in the specified paths.

`audiodiff.commandlinetool.diff_files(path1, path2, options)`

Compares the two files and prints the results.

`audiodiff.commandlinetool.diff_dirs(path1, path2, options)`

Compares the two directories and prints the results.

`audiodiff.commandlinetool.diff_streams(path1, path2, verbose=False, ffmpeg_bin=None)`

Prints whether the two audio files' streams differ or are identical.

`audiodiff.commandlinetool.diff_tags(path1, path2, verbose=False, brief=False)`

Prints whether the two audio files' tags differ or are identical.

`audiodiff.commandlinetool.diff_binary(path1, path2, verbose=False)`

Prints whether the two non-audio files differ or are identical.

---

## Indices and tables

---

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



## **a**

`audiodiff`, [15](#)

`audiodiff.commandlinetool`, [16](#)