
phrydy
Release 0+unknown

Apr 28, 2019

Contents

1	phrydy	3
1.1	phrydy package	3
2	Indices and tables	11
2.1	Usage	11
	Python Module Index	13

Contents:

1.1 phrydy package

1.1.1 Module contents

1.1.2 Submodules

phrydy.doc module

`phrydy.doc.fields` = {'acoustid_fingerprint': {'category': 'music_brainz', 'description': 'A multidimensional dictionary documenting all metadata fields.

```
fields = {
    'field': {
        'description': 'Title',
        'category': 'Category',
    },
}
```

`phrydy.doc.get_doc` (*additional_doc=False, field_prefix='\$', field_suffix=':', indent=4*)
Return a formatted string containing documentation about the audio fields.

`phrydy.doc.get_max_field_length` (*fields*)
Get the length of the longest field in the dictionary *fields*.

Parameters *fields* (*dict*) – A dictionary to search for the longest field.

`phrydy.doc.merge_fields` (**fields*)

`phrydy.doc.print_debug` (*media_file, MediaClass, field_generator, color=False*)

`phrydy.doc.print_dict_sorted` (*dictionary, color, align='right'*)

`phrydy.doc.print_section` (*text, color=False*)

`phrydy.doc.fields = metadata description`

A multidimensional dictionary documenting all metadata fields.

```
fields = {
    'field': {
        'description': 'Title',
        'category': 'Category',
    },
}
```

phrydy.mediafile module

This file originates from the file `beets/mediafile.py` of the `beets` project.

Handles low-level interfacing for files' tags. Wraps Mutagen to automatically detect file types and provide a unified interface for a useful subset of music files' tags.

Usage:

```
>>> f = MediaFile('Lucy.mp3')
>>> f.title
u'Lucy in the Sky with Diamonds'
>>> f.artist = 'The Beatles'
>>> f.save()
```

A field will always return a reasonable value of the correct type, even if no tag is present. If no value is available, the value will be false (e.g., zero or the empty string).

Internally `MediaFile` uses `MediaField` descriptors to access the data from the tags. In turn `MediaField` uses a number of `StorageStyle` strategies to handle format specific logic.

exception `phrydy.mediafile.UnreadableFileError` (*path, msg*)

Bases: `Exception`

Mutagen is not able to extract information from the file.

exception `phrydy.mediafile.FileTypeError` (*path, mutagen_type=None*)

Bases: `phrydy.mediafile.UnreadableFileError`

Reading this type of file is not supported.

If passed the *mutagen_type* argument this indicates that the mutagen type is not supported by *Mediafile*.

class `phrydy.mediafile.MediaFile` (*path, id3v23=False*)

Bases: `object`

Represents a multimedia file on disk and provides access to its metadata.

classmethod `_field_sort_name` (*name*)

Get a sort key for a field name that determines the order fields should be written in.

Fields names are kept unchanged, unless they are instances of `DateItemField`, in which case *year*, *month*, and *day* are replaced by *date0*, *date1*, and *date2*, respectively, to make them appear in that order.

acoustid_fingerprint

A descriptor providing access to a particular (abstract) metadata field.

acoustid_id

A descriptor providing access to a particular (abstract) metadata field.

classmethod `add_field` (*name, descriptor*)

Add a field to store custom tags.

Parameters

- **name** – the name of the property the field is accessed through. It must not already exist on this class.
- **descriptor** – an instance of `MediaField`.

album

A descriptor providing access to a particular (abstract) metadata field.

albumartist

A descriptor providing access to a particular (abstract) metadata field.

albumartist_credit

A descriptor providing access to a particular (abstract) metadata field.

albumartist_sort

A descriptor providing access to a particular (abstract) metadata field.

albumdisambig

A descriptor providing access to a particular (abstract) metadata field.

albumstatus

A descriptor providing access to a particular (abstract) metadata field.

albumtype

A descriptor providing access to a particular (abstract) metadata field.

arranger

A descriptor providing access to a particular (abstract) metadata field.

art

A descriptor that provides access to the *raw image data* for the cover image on a file. This is used for backwards compatibility: the full *ImageListField* provides richer *Image* objects.

When there are multiple images we try to pick the most likely to be a front cover.

artist

A descriptor providing access to a particular (abstract) metadata field.

artist_credit

A descriptor providing access to a particular (abstract) metadata field.

artist_sort

A descriptor providing access to a particular (abstract) metadata field.

asin

A descriptor providing access to a particular (abstract) metadata field.

bitdepth

The number of bits per sample in the audio encoding (an int). Only available for certain file formats (zero where unavailable).

bitrate

The number of bits per seconds used in the audio coding (an int). If this is provided explicitly by the compressed file format, this is a precise reflection of the encoding. Otherwise, it is estimated from the on-disk file size. In this case, some imprecision is possible because the file header is incorporated in the file size.

bpm

A descriptor providing access to a particular (abstract) metadata field.

catalognum

A descriptor providing access to a particular (abstract) metadata field.

channels

The number of channels in the audio (an int).

comments

A descriptor providing access to a particular (abstract) metadata field.

comp

A descriptor providing access to a particular (abstract) metadata field.

composer

A descriptor providing access to a particular (abstract) metadata field.

composer_sort

A descriptor providing access to a particular (abstract) metadata field.

country

A descriptor providing access to a particular (abstract) metadata field.

date

Descriptor that handles serializing and deserializing dates

The getter parses value from tags into a `datetime.date` instance and setter serializes such an instance into a string.

For granular access to year, month, and day, use the `*_field` methods to create corresponding *DateItemFields*.

day

Descriptor that gets and sets constituent parts of a *DateField*: the month, day, or year.

delete ()

Remove the current metadata tag from the file. May throw *UnreadableFileError*.

disc

A descriptor providing access to a particular (abstract) metadata field.

disctitle

A descriptor providing access to a particular (abstract) metadata field.

disctotal

A descriptor providing access to a particular (abstract) metadata field.

encoder

A descriptor providing access to a particular (abstract) metadata field.

classmethod fields ()

Get the names of all writable properties that reflect metadata tags (i.e., those that are instances of *MediaField*).

format

A string describing the file format/codecs.

genre

A descriptor providing access to a particular (abstract) metadata field.

genres

Property descriptor that retrieves a list of multiple values from a tag.

Uses `get_list` and `set_list` methods of its *StorageStyle* strategies to do the actual work.

grouping

A descriptor providing access to a particular (abstract) metadata field.

images

Descriptor to access the list of images embedded in tags.

The getter returns a list of *Image* instances obtained from the tags. The setter accepts a list of *Image* instances to be written to the tags.

initial_key

A descriptor providing access to a particular (abstract) metadata field.

label

A descriptor providing access to a particular (abstract) metadata field.

language

A descriptor providing access to a particular (abstract) metadata field.

length

The duration of the audio in seconds (a float).

lyricist

A descriptor providing access to a particular (abstract) metadata field.

lyrics

A descriptor providing access to a particular (abstract) metadata field.

mb_albumartistid

A descriptor providing access to a particular (abstract) metadata field.

mb_albumid

A descriptor providing access to a particular (abstract) metadata field.

mb_artistid

A descriptor providing access to a particular (abstract) metadata field.

mb_releasegroupid

A descriptor providing access to a particular (abstract) metadata field.

mb_releasetrackid

A descriptor providing access to a particular (abstract) metadata field.

mb_trackid

A descriptor providing access to a particular (abstract) metadata field.

mb_workhierarchy_ids

All IDs in the work hierarchy. This field corresponds to the field *work_hierarchy*. The top level work ID appears first. As separator a slash (/) is used.

Example:

```
e208c5f5-5d37-3dfc-ac0b-999f207c9e46 / 5adc213f-700a-4435-9e95-831ed720f348  
/ eafec51f-47c5-3c66-8c36-a524246c85f8
```

mb_workid

The MusicBrainz' Work ID

media

A descriptor providing access to a particular (abstract) metadata field.

month

Descriptor that gets and sets constituent parts of a *DateField*: the month, day, or year.

original_date

Descriptor that handles serializing and deserializing dates

The getter parses value from `tags` into a `datetime.date` instance and setter serializes such an instance into a string.

For granular access to year, month, and day, use the `*_field` methods to create corresponding *DateItemFields*.

original_day

Descriptor that gets and sets constituent parts of a *DateField*: the month, day, or year.

original_month

Descriptor that gets and sets constituent parts of a *DateField*: the month, day, or year.

original_year

Descriptor that gets and sets constituent parts of a *DateField*: the month, day, or year.

r128_album_gain

A descriptor providing access to a particular (abstract) metadata field.

r128_track_gain

A descriptor providing access to a particular (abstract) metadata field.

classmethod readable_fields()

Get all metadata fields: the writable ones from *fields()* and also other audio properties.

releasegroup_types

This field collects all items in the MusicBrainz' API related to `type: type, primary-type` and `secondary-type-list`. Main usage of this field is to determine in a secure manner if the release is a soundtrack.

```
"release-group": {
  "first-release-date": "1994-09-27",
  "secondary-type-list": [
    "Compilation",
    "Soundtrack"
  ],
  "primary-type": "Album",
  "title": "Pulp Fiction: Music From the Motion Picture",
  "type": "Soundtrack",
  "id": "1703cd63-9401-33c0-87c6-50c4ba2e0ba8"
}
```

rg_album_gain

A descriptor providing access to a particular (abstract) metadata field.

rg_album_peak

A descriptor providing access to a particular (abstract) metadata field.

rg_track_gain

A descriptor providing access to a particular (abstract) metadata field.

rg_track_peak

A descriptor providing access to a particular (abstract) metadata field.

samplerate

The audio's sample rate (an int).

save()

Write the object's tags back to the file. May throw *UnreadableFileError*.

script

A descriptor providing access to a particular (abstract) metadata field.

classmethod sorted_fields ()

Get the names of all writable metadata fields, sorted in the order that they should be written.

This is a lexicographic order, except for instances of `DateItemField`, which are sorted in year-month-day order.

title

A descriptor providing access to a particular (abstract) metadata field.

track

A descriptor providing access to a particular (abstract) metadata field.

tracktotal

A descriptor providing access to a particular (abstract) metadata field.

update (dict)

Set all field values from a dictionary.

For any key in *dict* that is also a field to store tags the method retrieves the corresponding value from *dict* and updates the *MediaFile*. If a key has the value *None*, the corresponding property is deleted from the *MediaFile*.

work

The last work in the work hierarchy.

work_hierarchy

The hierarchy of works: The top level work appears first. As separator is this string used: -->

Example:

```
Die Zauberflöte, K. 620 --> Die Zauberflöte, K. 620: Akt I --> Die
Zauberflöte, K. 620: Act I, Scene II. No. 2 Aria "Was hör' ...
```

year

Descriptor that gets and sets constituent parts of a *DateField*: the month, day, or year.

phrydy.utils module**phrydy.utils._fsencoding ()**

Get the system's filesystem encoding. On Windows, this is always UTF-8 (not MBCS).

phrydy.utils.as_string (value)

Convert a value to a Unicode object for matching with a query. None becomes the empty string. Bytestrings are silently decoded.

phrydy.utils.bytestring_path (path)

Given a path, which is either a bytes or a unicode, returns a str path (ensuring that we never deal with Unicode pathnames).

phrydy.utils.displayable_path (path, separator=';')

Attempts to decode a bytestring path to a unicode object for the purpose of displaying it to the user. If the *path* argument is a list or a tuple, the elements are joined with *separator*.

phrydy.utils.syspath (path, prefix=True)

Convert a path for use by the operating system. In particular, paths on Windows must receive a magic prefix and must be converted to Unicode before they are sent to the OS. To disable the magic prefix on Windows, set *prefix* to False—but only do this if you *really* know what you're doing.

- `genindex`
- `modindex`
- `search`

Handles low-level interfacing for files' tags. Wraps Mutagen to automatically detect file types and provide a unified interface for a useful subset of music files' tags.

2.1 Usage

```
>>> from phrydy import MediaFile
>>> f = MediaFile('Lucy.mp3')
>>> f.title
u'Lucy in the Sky with Diamonds'
>>> f.artist = 'The Beatles'
>>> f.save()
```

A field will always return a reasonable value of the correct type, even if no tag is present. If no value is available, the value will be false (e.g., zero or the empty string).

Internally *MediaFile* uses *MediaField* descriptors to access the data from the tags. In turn *MediaField* uses a number of *StorageStyle* strategies to handle format specific logic.

p

phrydy, 3
phrydy.doc, 3
phrydy.mediafile, 4
phrydy.utils, 9

Symbols

`_field_sort_name()` (*phrydy.mediafile.MediaFile* class method), 4
`_fsencoding()` (in module *phrydy.utils*), 9

A

`acoustid_fingerprint` (*phrydy.mediafile.MediaFile* attribute), 4
`acoustid_id` (*phrydy.mediafile.MediaFile* attribute), 4
`add_field()` (*phrydy.mediafile.MediaFile* class method), 4
`album` (*phrydy.mediafile.MediaFile* attribute), 5
`albumartist` (*phrydy.mediafile.MediaFile* attribute), 5
`albumartist_credit` (*phrydy.mediafile.MediaFile* attribute), 5
`albumartist_sort` (*phrydy.mediafile.MediaFile* attribute), 5
`albumdisambig` (*phrydy.mediafile.MediaFile* attribute), 5
`albumstatus` (*phrydy.mediafile.MediaFile* attribute), 5
`albumtype` (*phrydy.mediafile.MediaFile* attribute), 5
`arranger` (*phrydy.mediafile.MediaFile* attribute), 5
`art` (*phrydy.mediafile.MediaFile* attribute), 5
`artist` (*phrydy.mediafile.MediaFile* attribute), 5
`artist_credit` (*phrydy.mediafile.MediaFile* attribute), 5
`artist_sort` (*phrydy.mediafile.MediaFile* attribute), 5
`as_string()` (in module *phrydy.utils*), 9
`asin` (*phrydy.mediafile.MediaFile* attribute), 5

B

`bitdepth` (*phrydy.mediafile.MediaFile* attribute), 5
`bitrate` (*phrydy.mediafile.MediaFile* attribute), 5
`bpm` (*phrydy.mediafile.MediaFile* attribute), 5
`bytestring_path()` (in module *phrydy.utils*), 9

C

`catalognum` (*phrydy.mediafile.MediaFile* attribute), 5
`channels` (*phrydy.mediafile.MediaFile* attribute), 5
`comments` (*phrydy.mediafile.MediaFile* attribute), 6
`comp` (*phrydy.mediafile.MediaFile* attribute), 6
`composer` (*phrydy.mediafile.MediaFile* attribute), 6
`composer_sort` (*phrydy.mediafile.MediaFile* attribute), 6
`country` (*phrydy.mediafile.MediaFile* attribute), 6

D

`date` (*phrydy.mediafile.MediaFile* attribute), 6
`day` (*phrydy.mediafile.MediaFile* attribute), 6
`delete()` (*phrydy.mediafile.MediaFile* method), 6
`disc` (*phrydy.mediafile.MediaFile* attribute), 6
`disctitle` (*phrydy.mediafile.MediaFile* attribute), 6
`disctotal` (*phrydy.mediafile.MediaFile* attribute), 6
`displayable_path()` (in module *phrydy.utils*), 9

E

`encoder` (*phrydy.mediafile.MediaFile* attribute), 6

F

`fields` (in module *phrydy.doc*), 3
`fields()` (*phrydy.mediafile.MediaFile* class method), 6
`FileTypeError`, 4
`format` (*phrydy.mediafile.MediaFile* attribute), 6

G

`genre` (*phrydy.mediafile.MediaFile* attribute), 6
`genres` (*phrydy.mediafile.MediaFile* attribute), 6
`get_doc()` (in module *phrydy.doc*), 3
`get_max_field_length()` (in module *phrydy.doc*), 3
`grouping` (*phrydy.mediafile.MediaFile* attribute), 6

I

`images` (*phrydy.mediafile.MediaFile* attribute), 6

initial_key (*phrydy.mediafile.MediaFile* attribute), 7

L

label (*phrydy.mediafile.MediaFile* attribute), 7
language (*phrydy.mediafile.MediaFile* attribute), 7
length (*phrydy.mediafile.MediaFile* attribute), 7
lyricist (*phrydy.mediafile.MediaFile* attribute), 7
lyrics (*phrydy.mediafile.MediaFile* attribute), 7

M

mb_albumartistid (*phrydy.mediafile.MediaFile* attribute), 7
mb_albumid (*phrydy.mediafile.MediaFile* attribute), 7
mb_artistid (*phrydy.mediafile.MediaFile* attribute), 7
mb_releasegroupid (*phrydy.mediafile.MediaFile* attribute), 7
mb_releasetrackid (*phrydy.mediafile.MediaFile* attribute), 7
mb_trackid (*phrydy.mediafile.MediaFile* attribute), 7
mb_workhierarchy_ids (*phrydy.mediafile.MediaFile* attribute), 7
mb_workid (*phrydy.mediafile.MediaFile* attribute), 7
media (*phrydy.mediafile.MediaFile* attribute), 7
MediaFile (class in *phrydy.mediafile*), 4
merge_fields() (in module *phrydy.doc*), 3
month (*phrydy.mediafile.MediaFile* attribute), 7

O

original_date (*phrydy.mediafile.MediaFile* attribute), 7
original_day (*phrydy.mediafile.MediaFile* attribute), 8
original_month (*phrydy.mediafile.MediaFile* attribute), 8
original_year (*phrydy.mediafile.MediaFile* attribute), 8

P

phrydy (module), 3
phrydy.doc (module), 3
phrydy.mediafile (module), 4
phrydy.utils (module), 9
print_debug() (in module *phrydy.doc*), 3
print_dict_sorted() (in module *phrydy.doc*), 3
print_section() (in module *phrydy.doc*), 3

R

r128_album_gain (*phrydy.mediafile.MediaFile* attribute), 8
r128_track_gain (*phrydy.mediafile.MediaFile* attribute), 8

readable_fields() (*phrydy.mediafile.MediaFile* class method), 8

releasegroup_types (*phrydy.mediafile.MediaFile* attribute), 8

rg_album_gain (*phrydy.mediafile.MediaFile* attribute), 8

rg_album_peak (*phrydy.mediafile.MediaFile* attribute), 8

rg_track_gain (*phrydy.mediafile.MediaFile* attribute), 8

rg_track_peak (*phrydy.mediafile.MediaFile* attribute), 8

S

samplerate (*phrydy.mediafile.MediaFile* attribute), 8
save() (*phrydy.mediafile.MediaFile* method), 8
script (*phrydy.mediafile.MediaFile* attribute), 8
sorted_fields() (*phrydy.mediafile.MediaFile* class method), 8
syspath() (in module *phrydy.utils*), 9

T

title (*phrydy.mediafile.MediaFile* attribute), 9
track (*phrydy.mediafile.MediaFile* attribute), 9
tracktotal (*phrydy.mediafile.MediaFile* attribute), 9

U

UnreadableFileError, 4
update() (*phrydy.mediafile.MediaFile* method), 9

W

work (*phrydy.mediafile.MediaFile* attribute), 9
work_hierarchy (*phrydy.mediafile.MediaFile* attribute), 9

Y

year (*phrydy.mediafile.MediaFile* attribute), 9