
GhettoRecorder

René Horn

Sep 23, 2023

CONTENTS:

1	Documentation - GhettoRecorder	1
1.1	How to run installed package	1
1.2	Overview	1
1.3	Links	2
1.4	GhettoRecorder module	5
1.5	Snapcraft package	5
2	ghettorecorder	7
2.1	ghettorecorder package	7
3	Indices and tables	11
	Python Module Index	13
	Index	15

DOCUMENTATION - GHETTORECORDER

Grab hundreds of radio stations *simultaneously*.

1.1 How to run installed package

GhettoRecorder class module (example in ghetto_procnv).:

```
from ghettorecorder import GhettoRecorder

ghetto_01 = GhettoRecorder(radio, url)
ghetto_01.com_in = mp.Queue(maxsize=1) # eval exec communication for multiprocessing
ghetto_01.audio_out = mp.Queue(maxsize=1) # can also be normal queue.Queue()
```

Commandline option (calls cmd.py).:

```
ghetto_cmd or
python3 -m ghettorecorder.cmd
```

Client Server option (calls __main__.py).:

```
ghetto_url or
python3 -m ghettorecorder
```

1.2 Overview

- Queue communication. Multiprocessor ready.
- GhettoRecorder class has connector attributes for external modules.
- External modul *Blacklisting recorded titles* is already included.
- Optional Browser Frontend on Python multithreading HTTP server.

1.3 Links

- Snap: <https://snapcraft.io/ghettorecorder>
- GitHub: <https://github.com/44xtc44/GhettoRecorder>
- Issues to fix: <https://github.com/44xtc44/GhettoRecorder/issues>
- ReadTheDocs: <https://ghettorecorder.readthedocs.io/> (see module index)

1.3.1 Configuration File

‘Settings.ini’ is the config file for GhettoRecorder. INI files consist of sections to divide different settings.:

```
[STATIONS]
anime_jp = http://streamingv2.shoutcast.com/japanimradio-tokyo

[GLOBAL]
blacklist_enable = True
save_to_dir = f:\54321
```

[STATIONS]
custom radio name and radio connection information (can be pls or m3u playlist)

[GLOBAL]
stores blacklist status and the *custom* parent directory location

1.3.2 Usage

Main Menu

```
menu 'Main'
1 -- Record (local listen option)
2 -- Change parent record path
3 -- Enable/disable blacklists
4 -- Set path to config, settings.ini
5 -- aac file repair
6 -- Exit
```

Record Menu

```
0  >> aacchill          <<
1  >> 80ies_nl          <<
2  >> anime_jp           <<
3  >> blues_uk           <<
4  >> br24               <<
...
Enter to record -->:
```

Write the leading Number (list index) into the input field . Hit 'Enter'.

OR

Write or copy/paste the radio name into the input field. Hit 'Enter'.

Add as many radios as you like.

Hit 'Enter' without input to start grabbing.

Listen to the first selected radio via local streaming <http://localhost:1242/>

Change parent record path Menu

```
option 'Change record parent path'
1 -- New parent path for recorded radios. Write to config.
2 -- Back to Main Menu
Enter your choice: 1

    Write a new path to store files
..settings.ini [GLOBAL] section: {'blacklist_enable': 'True', 'save_to_dir': 'f:\\31'}
Enter a new path, OS syntax (f:\\10 or /home ) -->:
```

The default path is the directory of the module. In most cases you want to store grabbed files somewhere else.

Blacklist Menu

```
Write a new blacklist option to settings.ini file
..settings.ini [GLOBAL] section: {'blacklist_enable': 'True', 'save_to_dir': 'f:\\31'}
1 -- blacklist on (don't write title if already downloaded)
2 -- blacklist off
3 -- Back to Main Menu
Enter your choice: 1

    blacklist is ON: settings.ini file
    Existing titles are not recorded again and again.
    file name is "blacklist.json" in the same folder as "settings.ini"
..settings.ini [GLOBAL] section: {'blacklist_enable': 'True', 'save_to_dir': 'f:\\31'}
Hit Enter to leave -->:
```

Blacklist writing can be switched on/off.

Titles are listed for each of the radios and can be deleted to 'unlist' them.

File name is `blacklist.json` and always in the same folder as `'settings.ini'`.

Set path to config

```
Write Path to settings.ini and blacklist.json file
Enter a new path, OS syntax (f:\10 or /home ) -->: F:\44
created: F:\44
..settings.ini [GLOBAL] section: {'blacklist_enable': 'True'}
Hit Enter to leave -->:
```

You can store your config file `'settings.ini'` somewhere on the file system.

Default place for grabbed files is the mentioned folder.

If a custom save path is written to config, this path is used.

aac file repair

```
Write a path to aac files. Only aac files will be touched.
..settings.ini [GLOBAL] section: {'blacklist_enable': 'True', 'save_to_dir': 'f:\\31'}
Enter a path, OS syntax (f:\10 or /home ) -->:f:\6aac
created: f:\6aac
f:\6aac\aac_repair created
[ COPY(s) in f:\6aac\aac_repair ]
----- 1 file(s) failed -----
f:\6aac\Sergey Sirotin & Golden Light Orchestra - Around The World.aacp
ValueError non-hexadecimal number found in fromhex() arg at position 5438113
----- 97 file(s) repaired -----
f:\6aac\111_Slovo_Original_Mix.aac; cut(bytes): [330]
f:\6aac\351 Lake Shore Drive - You Make My Day.aacp; cut(bytes): [389]
```

The repair option uses a folder name as input.

Repaired files are stored in `'aac_repair'` sub folder.

Cut bytes count is shown at the end of the line.

Repair can fail if the file is corrupted not only at start or end.

Pip Install

```
""" Linux """
$ pip3 install ghettorecorder

""" Windows """
> pip install ghettorecorder
```


Uninstall

Python user

- find the module location
- uninstall and then remove remnants

remove:

```
>$ pip3 show ghettorecorder
>$ pip3 uninstall ghettorecorder
```

Location: ... /python310/site-packages

1.4 GhettoRecorder module

Communication with the GhettoRecorder instance

port	action	description
com_in	commands input	tuple (radio, [str 'eval' or 'exec'], str 'command')
com_out	status, err msg	(radio, [str 'eval' or 'exec'], response)
audio_out	copy of html resp	server can loop through to a browser

Feature attributes to switch on/off

attribute	description
runs_meta	call metadata periodically, create path for rec out; False: recorder is the file
runs_record	disable writing to recorder file at all
recorder_file_write	allow dumping current recorder file
runs_listen	disable write to audio output queue; 3rd party can grab it. (listen blacklist)

1.5 Snapcraft package

The installer creates an icon with the name “GhettoRecorder”. You can use two command line options.:

```
ghettorecorder.url
ghettorecorder.cmd
```

First is Client, Server connection. Second is command line menu.

GHETTORECORDER

2.1 ghettorecorder package

2.1.1 Submodules

2.1.2 ghettorecorder.audio_conf module

2.1.3 ghettorecorder.cmd module

2.1.4 ghettorecorder.ghetto_agents module

2.1.5 ghettorecorder.ghetto_api module

2.1.6 ghettorecorder.ghetto_blacklist module

2.1.7 ghettorecorder.ghetto_container module

2.1.8 ghettorecorder.ghetto_db_worker module

2.1.9 ghettorecorder.ghetto_header_aac module

2.1.10 ghettorecorder.ghetto_header_mp3 module

Output is human-readable. Non-destructive. Use as input for further mp3 stream processing.

- Feed streaming web server with file system sound files. Needs bitrate per second, else browser hangs.

http://www.mp3-tech.org/programmer/frame_header.html

Bytes

1	2	3	4	5	6
AAAAAAAA	AAABBCCD	EEEEFFGH	IIJKLMM	OOOOOOOO	OOOOOOOO

Bit Groups

Group	Number	Count	Description
A	0-11	11	Syncword, all bits 1
B	12-13	2	[00 = MPEG Version 2.5] [01 = reserved] [10 = MPEG Version 2] [11 = MPEG Version 1]
C	14-15	2	Layer [00 = reserviert] [01 = Layer III] [10 = Layer II] [11 = Layer I]
D	16	1	[[[Warning]]], set to 1 if there is no CRC and 0 if there is CRC
E	17-20	4	Bitrate index table, Version and layer read the table
F	21-22	2	Sampling rate frequency index
G	23	1	Padding bit [0 - frame is not padded] [1 - frame is padded with one extra slot]
H	24	1	Private bit. This one is only informative.
I	25-26	2	Channel Mode [00 - Stereo] [01 - Joint stereo] [10 - Dual] [11 - Single (Mono)]
J	27-28	2	Mode extension (Only used in Joint stereo)
K	29	1	Copyright ID bit
L	30	1	Original [0 - Copy of original media] [1 - Original media]
M	31-32	2	Emphasis [00 - none] [01 - 50/15 ms] [10 - reserved] [11 - CCIT J.17]
O	33-48	16	CRC if existing, D 16 set

frame length layer 1: $\text{FrameLengthInBytes} = (12 * \text{BitRate} / \text{SampleRate} + \text{Padding}) * 4$ frame length

layer 2: $\text{FrameLengthInBytes} = 144 * \text{BitRate} / \text{SampleRate} + \text{Padding}$

The next line is C code:

```

//*****
// This reference data is from MPEGAudioInfo app
// Samples per Frame / 8
static const u32 m_dwCoefficients[2][3] =
{
    { // MPEG 1
      12, // Layer1 (must be multiplied with 4, because of slot size)
      144, // Layer2
      144 // Layer3
    },
    { // MPEG 2, 2.5
      12, // Layer1 (must be multiplied with 4, because of slot size)
      144, // Layer2
      72 // Layer3
    }
};

```

```
class ghettorecorder.ghetto_header_mp3.Mp3Header(file)
```

Bases: object

Find bitrate in header. 'br_dict' list with most votes wins.

__init__(self, file)

Param

file: # can be path string

bitrate_get()

Looking for the first bits of mp3 header. Search sync_word. Second byte of word has only 3bit information.

Mask and bit-shift.

Returns

bitrate int, or None

bitrate_get_from_dict()

Calculate the bitrate from dict. We have defective Header and intersections of unknown. Key bitrate with most votes wins.

Returns

bitrate of file like obj, else None

bitrate_one_frame_get(*mp3_object*)

Extract bitrate via bit shift and mask into 'br_dict'. Bit Groups listed in module's docString. and(ing) 11... bits, move resulting bits right and calculate

mp3_object_to_bytes()

Convert path to file like object, if path, else it is bytes object

`ghettorecorder.ghetto_header_mp3.main(path_str=None)`

Example to get the bitrate.

Param

path_str: file path or object

2.1.11 ghettherecorder.ghetto_http_simple module

2.1.12 ghettherecorder.ghetto_ini module

2.1.13 ghettherecorder.ghetto_menu module

2.1.14 ghettherecorder.ghetto_meta module

2.1.15 ghettherecorder.ghetto_net module

2.1.16 ghettherecorder.ghetto_procnv module

2.1.17 ghettherecorder.ghetto_recorder module

2.1.18 ghettherecorder.ghetto_utils module

2.1.19 Module contents

INDICES AND TABLES

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

PYTHON MODULE INDEX

g

`ghettorecorder.ghetto_header_mp3`, [7](#)

INDEX

B

`bitrate_get()` (*ghettorecorder.ghetto_header_mp3.Mp3Header*
method), 8

`bitrate_get_from_dict()` (*ghettorecorder.ghetto_header_mp3.Mp3Header*
method), 9

`bitrate_one_frame_get()` (*ghettorecorder.ghetto_header_mp3.Mp3Header*
method), 9

G

`ghettorecorder.ghetto_header_mp3`
module, 7

M

`main()` (*in module ghettherecorder.ghetto_header_mp3*),
9

module
 `ghettorecorder.ghetto_header_mp3`, 7

`mp3_object_to_bytes()` (*ghettorecorder.ghetto_header_mp3.Mp3Header*
method), 9

`Mp3Header` (class *in ghettherecorder.ghetto_header_mp3*), 8