

---

# **aacRepair**

**René Horn**

**Sep 11, 2023**



# CONTENTS

<b>1</b>	<b>Contents</b>	<b>3</b>
1.1	aacRepair - Full documentation . . . . .	3
1.2	aacrepair . . . . .	7
<b>2</b>	<b>Indices and tables</b>	<b>15</b>
	<b>Python Module Index</b>	<b>17</b>
	<b>Index</b>	<b>19</b>



repairs aac and aacPlus files grabbed from the internet



## CONTENTS

### 1.1 aacRepair - Full documentation

repair aac and aacPlus files grabbed from the internet



#### 1.1.1 Info

AAC files consist of multiple segments, frames. Each frame has a header and a payload. Browser gets stuck if AAC file frame is defective and will not start to play or refuse to play next AAC file. This will stop the entire playlist. File gets trimmed from head to tail, to remove defective frames. Cut off byte count is shown in the summary.

---

**Note:** Self documenting code is (docStrings) short and read by document builder tools. -> *aacrepair package*

---

#### 1.1.2 Command Line

```
$ aacrepair

    Menu "Main"
1 -- Single File aac or aacPlus
2 -- Bulk Repair, Folder
3 -- Exit
Enter your choice: 1
```

Python or sys.path fail; use

```
$ python -m aacrepair.cmd
```

### 1.1.3 aacrepair Module

#### bulk repair

```
from aacrepair import AacRepair

# 'r' before a string tells the Python interpreter to treat backslashes as a literal,
↳ (raw) character
aacRepair = AacRepair(r"F:\propaganda-podcasts")
# setter overrides default export path 'F:\propaganda-podcasts\aac_repair'
aacRepair.set_export_path(r"F:\repaired_foobar")
aacRepair.repair()
```

**Instantiate AacRepair class with two possible arguments, mandatory folder path and optional dictionary.**

1. No dictionary provided. Folder path is used as list to import files into a dictionary AND store repaired files.
2. A dictionary of files is provided. Folder path is used to store repaired files. (best use on web server)

#### Web Server

- Endpoint converts multi-file upload from file storage type to bytestream, uses .read() method
- List of files is written to dictionary {file\_name\_key: file\_byte\_content\_value}
- web server gets not the file path, only file name - needs path to store repaired files
- dictionary {file(n).aac: b'x65x66x67x00x10x00x00x00x04x00' }

code:

```
files = request.files.getlist('fileUploadAcpRepair')
f_dict = {f: open(f, "rb").read() for f in files if f[-5:] == ".aacp" or f[-4:] == ".aac"
↳ ""}
aacRepair = AacRepair("/home/Kitty/aac_files", f_dict)
aacRepair.repair()
```

#### File System

- List of files in folder is written to dictionary {file\_name\_key: file\_byte\_content\_value}

code:

```
aacRepair = AacRepair("/home/Kitty/meow_aac")
aacRepair.set_export_path("/home/Kitty/foo")
aacRepair.repair()
```

#### single object

head and tail are used to cut chunks left or right only

```
aacRepair = AacRepair()
# converts file path to file content, if object is not of type bytes
rep_object = aacRepair.repair_object(aac_path_or_object, head=None, tail=None)
```



### 1.1.4 header\_aac Module

Use as input for further aac stream processing or repair.

```
from aacrepair import header_aac
```

header\_aac module example to show all frames with header.

```
header_aac.pull_frame('/home/foo/bar.aac')
```

header\_aac dictionary output of header\_info(aac\_object, frame\_bytes=None, print\_out=True)

```
SYNC_WORD_BOOL: True
MPEG4_BOOL: True
Layer_BOOL: True
CRC_16_IS_SET_BOOL: False
PROFILE_INT: 1
PROFILE_STR: AAC Main
SAMPLING_FREQUENCY_INT: 3
SAMPLING_FREQUENCY_STR: 48000 Hz
PRIVATE_BIT_BOOL: False
CHANNEL_CONFIG_INT: 2
CHANNEL_CONFIG_STR: 2 channels: front-left, front-right
ORIGINALITY_BOOL: False
HOME_BOOL: False
COPYRIGHT_ID_INT: 0
COPYRIGHT_START_INT: 0
FRAME_LENGTH_INT: 530
BIT_RESERVOIR_INT: 2047
FRAME_NUMBER_INT: 0
CRC_16: {}
IS_LAST_FRAME_BOOL: False
ERROR_STR:
FRAME_BYTES: b''
```

Bytes

1	2	3	4	5	6	7	8	9
AAAAAA	AAAA	BCDE	FFFF	HHI-	MMMM-	MM-	OOOOOO	PP
		FGH	JKLM	MM	MOOOOO		QQ	QQQQQQ

Bit Groups

Group	Num-ber	Count	Description
A	0-12	12	Syncword, all bits 1
B	13	1	MPEG Version set 0 is MPEG-4, set 1 MPEG-2
C	14-15	2	Layer set to 0
D	16	1	[[[ Warning ]]], set to 1 if there is no CRC and 0 if there is CRC
E	17-18	2	Profile, the MPEG-4 Audio Object Type <a href="https://en.wikipedia.org/wiki/MPEG-4_Part_3">https://en.wikipedia.org/wiki/MPEG-4_Part_3</a>
F	19-22	4	MPEG-4 Sampling Frequency <a href="https://wiki.multimedia.cx/index.php/MPEG-4_Audio">https://wiki.multimedia.cx/index.php/MPEG-4_Audio</a>
G	23	1	Private bit set to 0,
H	24-26	3	MPEG-4 Channel Configuration <a href="https://wiki.multimedia.cx/index.php/MPEG-4_Audio">https://wiki.multimedia.cx/index.php/MPEG-4_Audio</a>
I	27	1	Originality, set 1 originality of audio, else 0
J	28	1	Home, set to 1 to signal home usage of the audio, else 0
K	29	1	Copyright ID bit
L	30	1	Copyright ID start
M	31-43	13	Frame length of ADTS frame including headers and CRC check - 1
O	44-54	11	Buffer fullness, states the bit-reservoir per frame
P	55-56	2	Number of AAC ADTS frame minus 1
Q	57-72	16	CRC check

### 1.1.5 pip install

```
""" Linux """
$ pip3 install aacrepair

""" Windows """
> pip install aacrepair
```

### 1.1.6 Uninstall

Python user

- find the module location
- uninstall and then remove remnants

remove:

```
>$ pip3 show aacrepair
>$ pip3 uninstall aacrepair
```

Location: ... /python310/site-packages

### 1.1.7 What's next - contributions welcome

- try multithread a test
- multiprocessing plus multithreading repair, if file count is x (works only on linux)
- android 'Threading in Worker', make it run

## 1.2 aacrepair

### 1.2.1 aacrepair package

#### Submodules

#### aacrepair.audio\_conf module

Translate Bit representation of audio properties into readable strings.

- "PROFILE\_INT": 3
- "PROFILE\_STR": AAC SSR (Scalable Sample Rate)

#### aacrepair.cmd module

Module runs aacrepair on commandline with menu options.

Please check error\_dict message, if instance fails to repair.

#### aacrepair.cmd.bulk\_repair()

Input loop to repair a whole folder of aac files.

##### Methods

prepare\_path\_write\_bulk: create instance with folder and export path argument

#### aacrepair.cmd.file\_repair()

Input loop to repair a single aac file.

##### Methods

repair\_write\_one\_file: read, repair and store renamed file

#### aacrepair.cmd.instance\_repair\_bulk(aac\_path, export\_path=None)

Create instance with option for custom export folder.

##### Params

aac\_path: src directory

##### Params

export\_path: this function calls the setter to change export directory

#### aacrepair.cmd.main()

Call menu\_main to start the module from command line.

#### aacrepair.cmd.menu\_main()

Main menu to choose from.

`aacrepair.cmd.prepare_path_run_write_bulk(aac_path)`

Prepare path arguments to feed the repair instance

**Params**

aac\_path: src directory arg[0] dst directory arg[1]

**Methods**

instance\_repair\_bulk: create instance with default folder or export path

`aacrepair.cmd.repair_write_one_file(aac_path, name_prefix=None, print_out=True)`

Repair a single aac or aacPlus file and writes it with a name prefix.

**Param**

aac\_path: file path

**Param**

name\_prefix: distinguish the file from defective one

**Param**

print\_out: None disables print to screen

**Methods**

header\_dict.header\_info: header bits to property dictionary

**Returns**

AAC frame header for further processing

**Return type**

dict

## **aacrepair.crc module**

Should help to read crc bytes and calculate checksum of a container and content.

Place to calculate checksum of Audio container and/or its content.

`aacrepair.crc.flip_this(bit_str)`

Complement 1's (flip 0 to 1, 1 to 0) string of bits.

**Param**

bit\_str: left to right ordered string of all CRC fields

**Returns**

1's complement

**Return type**

str

`aacrepair.crc.reveal_crc(byte_list, print_out=None)`

Helper to create a function for AAC stream CRC check. Calc all possible bit orders, to detect the CRC-CCITT input function pre return options used; Normal, reflected and/or reversed. Then use either Python CRC check, or a Pypi packet. aacrepair.header\_aac module can return hex output to put it in an online CRC checker or use fastCRC packet. Output must be altered bytes array to slice in the (clean-up before calc) crc sum fields 'ffff', if CRC bit is set. Document 'ETSI TS 102 563' section 5.2, Transport of Advanced Audio Coding (AAC) audio; Technical Specification Perhaps we can create a software error correction ourselves. Document 'header\_firecode', perhaps go from there.

**Params**

byte\_list: CRC SUM as list of 8 bits of type bytearray or bytes [8], [8,8], [8,8,8,8], [8,8,8,8,8,8,8,8]

**Params**

print\_out: disable screen printing

**Returns**

CRC SUM as string, hex and int representation of left right, right left, reversed and flipped bites order

**Return type**

dict

**aacrepair.dev\_aac module****aacrepair.haeder\_mp4 module****aacrepair.header\_aac module**

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

- check next fame is available
- check if profile changed to low quality stream
- compare sample rates to keep the better stream
- switch the stream url endpoint on low quality, if channel config changed

`aacrepair.header_aac.header_index_get(aac_object)`

Scan the file object for an aac frame. Search frame is hex.

**Param**

aac\_object: bytes

**Returns**

INDEX number of the first frame start Byte in the stream

**Return type**

int

`aacrepair.header_aac.header_info(aac_object, frame_bytes=None, print_out=None)`

Caller can name a slice of the object `aac_object[start idx:]` to work. Example how to move the header\_info in `header_aac.read_all_header()`.

**Param**

aac\_object: full object or slice

**Param**

frame\_bytes: dump whole frame to header\_dict['FRAME\_BYTES']

**Param**

print\_out: enable print to screen

**Returns**

frame header properties

**Return type**

dict

`aacrepair.header_aac.main()`

`aacrepair.header_aac.pull_frame(path_str=None)`

Example to get the header and all frames of the object (file content).

**Param**

path\_str: file path or object

`aacrepair.header_aac.read_all_header(aac_object, convert_bytes_hex=None)`

Example function returns all frame content with header until empty.

**Param**

aac\_object: bytes type object

**Param**

convert\_bytes\_hex: for cut-and-paste into checksum calculator

**Returns**

header information and object content

**Return type**

Iterator[dict]

## Module contents

Module repairs aac or aacPlus file objects. Stores on disk or returns in memory.

technical:

AAC file wants a nice head **and** tail.  
AAC header starts **hex** fff(1) **or** (9), mpeg-(4) **or** (2)  
Create a header search frame binary **b'yñ'**, **hex** fff1  
Move the header search frame over the AAC file.  
Search start **is** file[0:2] **bytes**, shift the search frame file[1:3], file[2:4]  
AAC head: remove the first defective frame ...[fff1 file]  
AAC tail: remove the last frame **with** defective payload [file]fff1...

**class** `aacrepair.AacRepair(folder=None, file_dict=None)`

Bases: object

Write repaired aac or aacPlus file objects to disk or memory.

`__init__`

**Params**

folder: use folder for dict if file\_dict is None, else is export\_path

**Params**

file\_dict: {file name or path: content}, folder is export\_path

**Methods**

file\_dict\_from\_folder: read content of aac files folder into a dict for bulk repair

Class Methods

**Methods**

set\_export\_path: setter export\_path

**Methods**

get\_export\_path: getter export\_path

**Methods**

file\_dict\_from\_folder: {file\_path: content,}

**Methods**

make\_dirs: create a folder with subfolders

**Methods**

repair: threaded call of repair\_one\_file with path, content args

**Methods**

convert\_file\_like\_object: return a bytes type object

**Methods**

repair\_object: single file or chunk, dispatcher

**Methods**

repair\_one\_file: head and tail repair

**Methods**

repair\_head: only head

**Methods**

repair\_tail: only tail

**Methods**

log\_writer: log\_list as result

**Methods**

byte\_calc: calc cut bytes

**Methods**

write\_repaired\_file: write object to file

**Methods**

delete\_file\_dict: del file\_dict

**Methods**

all\_files\_touched: if fail generate report

**Methods**

report\_skip\_list: skipped files not in error\_dict; shall reveal module failure

**all\_files\_touched()**

Check for error in module, file failure not written in error\_dict.

**Returns**

False if calc not match, see report\_skip\_list

**byte\_calc(f\_name)**

Return number of cut bytes.

**Param**

f\_name: file path for error, or info dict

**Exception**

write error, set size to 1

**Return type**

int

**Returns**

cut bytes

**Return type**

int

**convert\_file\_like\_object**(*file\_full\_name*)

Read file from file system and return content as bytes type.

**Param**

file\_full\_name: The absolute path to the file

**Exception**

write report into error\_dict

**Return type**

False

**Returns**

object type bytes

**delete\_file\_dict**()

Outsourced to prevent del dict in test mode.

**file\_dict\_from\_folder**()

Create dictionary of files {name: content} for the 'repair()' method, bulk repair.

Class can also use an existing dictionary (prepared by web server multi-file upload)

**get\_export\_path**()

Getter for export\_path dir.

**log\_writer**()

Write available logs to screen and keep it log\_list for later HTML colorized report.

**Return type**

True: ok

**Return type**

False: errors found, or files where skipped due to a program bug

**static make\_dirs**(*path*)

Create folders.

**Param**

path: absolute path that can contain subdirectories

**repair**(*keep\_file\_dict=None*)

Threaded bulk repair from dictionary.

**Params**

keep\_file\_dict: delete the dictionary yourself

**Return type**

True; if keep\_file\_dict manual delete\_file\_dict

**Return type**

False; check skip\_list and error\_dict, then -> delete both

**repair\_head**(*f\_name, chunk*)

Return bytes content left repaired ... [fff1 file]

**Param**

f\_name: file name for the error, info dict

**Param**

chunk: The data to cut.



**Returns**

binary data

**Return type**

bytes

**repair\_object**(*chunk, head=None, tail=None*)

Return the repaired binary file object from binary source (buffer) or file content.

*file\_full\_name* is alias to reuse methods and write to error and info dicts

**Params**

chunk: buffer, queue, or file content

**Params**

head: of chunk repair

**Params**

tail: of chunk repair

**Methods**

convert\_file\_like\_object: get file content from path given

**Methods**

repair\_one\_file: one chunk left and right, call repair\_head and repair\_tail

**Methods**

repair\_head: left side only

**Methods**

repair\_tail: right side only

**Return type**

bytes type file object

**Return type**

False; check *error\_dict* after this method call

**repair\_one\_file**(*file\_full\_name, damaged\_data, skip\_write=None*)

Repair the beginning or end of file (*damaged\_data* is dictionary value).

**Params**

*file\_full\_name*: The absolute path to the file.

**Params**

*damaged\_data*: binary file content

**Params**

*file\_full\_name*: bulk repair, name of export folder

**Params**

*skip\_write*: method should not write a file

**Return type**

bytes: binary data: repaired file content

**Return type**

False: read the error dict

**repair\_tail**(*f\_name, chunk*)

Return bytes content right repaired [file]fff1...

**Param**

*f\_name*: file name for the error, info dict

**Param**

chunk: The data to cut.

**Returns**

binary data

**Return type**

bytes

**report\_skip\_list()**

all\_files\_touched sum calc got no match, write file names of skipped files into skip\_list

**Returns**

list of skipped files

**Return type**

list

**set\_export\_path(*path*)**

Setter for export\_path dir attribute.

**static write\_repaired\_file(*export\_path*, *file\_content*)**

Write repaired file content to disk.

**Param**

export\_path: setter or os.joined default export path

**Param**

file\_content: bytes type content

## INDICES AND TABLES

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



## PYTHON MODULE INDEX

### a

- `aacrepair`, [10](#)
- `aacrepair.audio_conf`, [7](#)
- `aacrepair.cmd`, [7](#)
- `aacrepair.crc`, [8](#)
- `aacrepair.header_aac`, [9](#)



## A

[aacrepair](#)  
     module, 10  
[AacRepair](#) (class in *aacrepair*), 10  
[aacrepair.audio\\_conf](#)  
     module, 7  
[aacrepair.cmd](#)  
     module, 7  
[aacrepair.crc](#)  
     module, 8  
[aacrepair.header\\_aac](#)  
     module, 9  
[all\\_files\\_touched\(\)](#) (*aacrepair.AacRepair* method), 11

## B

[bulk\\_repair\(\)](#) (in module *aacrepair.cmd*), 7  
[byte\\_calc\(\)](#) (*aacrepair.AacRepair* method), 11

## C

[convert\\_file\\_like\\_object\(\)](#) (*aacrepair.AacRepair* method), 11

## D

[delete\\_file\\_dict\(\)](#) (*aacrepair.AacRepair* method), 12

## F

[file\\_dict\\_from\\_folder\(\)](#) (*aacrepair.AacRepair* method), 12  
[file\\_repair\(\)](#) (in module *aacrepair.cmd*), 7  
[flip\\_this\(\)](#) (in module *aacrepair.crc*), 8

## G

[get\\_export\\_path\(\)](#) (*aacrepair.AacRepair* method), 12

## H

[header\\_index\\_get\(\)](#) (in module *aacrepair.header\_aac*), 9  
[header\\_info\(\)](#) (in module *aacrepair.header\_aac*), 9

## I

[instance\\_repair\\_bulk\(\)](#) (in module *aacrepair.cmd*), 7

## L

[log\\_writer\(\)](#) (*aacrepair.AacRepair* method), 12

## M

[main\(\)](#) (in module *aacrepair.cmd*), 7  
[main\(\)](#) (in module *aacrepair.header\_aac*), 9  
[make\\_dirs\(\)](#) (*aacrepair.AacRepair* static method), 12  
[menu\\_main\(\)](#) (in module *aacrepair.cmd*), 7  
[module](#)  
     [aacrepair](#), 10  
     [aacrepair.audio\\_conf](#), 7  
     [aacrepair.cmd](#), 7  
     [aacrepair.crc](#), 8  
     [aacrepair.header\\_aac](#), 9

## P

[prepare\\_path\\_run\\_write\\_bulk\(\)](#) (in module *aacrepair.cmd*), 7  
[pull\\_frame\(\)](#) (in module *aacrepair.header\_aac*), 9

## R

[read\\_all\\_header\(\)](#) (in module *aacrepair.header\_aac*), 10  
[repair\(\)](#) (*aacrepair.AacRepair* method), 12  
[repair\\_head\(\)](#) (*aacrepair.AacRepair* method), 12  
[repair\\_object\(\)](#) (*aacrepair.AacRepair* method), 13  
[repair\\_one\\_file\(\)](#) (*aacrepair.AacRepair* method), 13  
[repair\\_tail\(\)](#) (*aacrepair.AacRepair* method), 13  
[repair\\_write\\_one\\_file\(\)](#) (in module *aacrepair.cmd*), 8  
[report\\_skip\\_list\(\)](#) (*aacrepair.AacRepair* method), 14  
[reveal\\_crc\(\)](#) (in module *aacrepair.crc*), 8

## S

[set\\_export\\_path\(\)](#) (*aacrepair.AacRepair* method), 14

## W

`write_repaired_file()` (*aacrepair.AacRepair* static method), [14](#)