
pyffstream

Gregory Beauregard

Aug 07, 2022

CONTENTS

1	Reference	1
1.1	pyffstream.ffmpeg	1
1.2	pyffstream.encode	7
1.3	pyffstream.cli	12
2	License	17
3	Installation	19
4	CLI Usage	21
4.1	pyffstream	21
	Python Module Index	29
	Index	31

REFERENCE

- *pyffstream.ffmpeg*
- *pyffstream.encode*
- *pyffstream.cli*

1.1 pyffstream.ffmpeg

Helper functions for interacting with ffmpeg.

Functions include mimicking number/duration processing, probing files with ffprobe, and a utility class to deal with filters.

`pyffstream.ffmpeg.ff_bin`

A tuple that is the recommended way to specify the path to ffmpeg utilities. Can be overridden.

Type

FFBin

class `pyffstream.ffmpeg.FFBanner(ffversion, ffconfig, versions)`

ffconfig: `list[str]`

Alias for field number 1

ffversion: `str`

Alias for field number 0

versions: `dict[str, pyffstream.ffmpeg.FFVersion]`

Alias for field number 2

class `pyffstream.ffmpeg.FFBin(ffmpeg='ffmpeg', ffprobe='ffprobe', env=None)`

An instance of a path to an ffmpeg binary.

Provides various helper properties to get information about the capabilities of the ffmpeg binary. Note that just because something was compiled into the binary it doesn't mean it's usable at runtime. This is the case for hardware support (e.g. nvidia) in particular.

ffmpeg

Path to ffmpeg binary.

Type

`str`

ffprobe

Path to ffprobe binary.

Type

`str`

env

Environmental variables to use with ffmpeg.

Type

`dict[str, str]`

Parameters

- **ffmpeg** (`str`) –
- **ffprobe** (`str`) –
- **env** (`dict[str, str] | None`) –

property aencoders: `set[str]`

Set of compiled-in audio encoders.

Return type

`set[str]`

property build_config: `list[str]`

List of build config options.

Return type

`list[str]`

property ffversion: `str`

String representing ffmpeg release version.

Note this might not always be sensible to use for simple comparison, for example in the case of versions compiled from git.

Return type

`str`

property filters: `set[str]`

Set of filters compiled into ffmpeg instance.

property hwaccels: `set[str]`

Set of hwaccels compiled into ffmpeg version.

make_playlist(*pathlist*, *directory*, *idxs*=(0, 0), *deep_probe*=False, *name*='streamplaylist.txt')

Construct fconcat playlist from path list.

Paths created are absolute, and the files are probed in parallel if needed to determine the duration.

Parameters

- **pathlist** (`Iterable[Path]`) – Ordered pathlist to construct fconcat with.
- **directory** (`Path`) – Directory to put the constructed playlist in.
- **idxs** (`Sequence[int]`) – Tuple of (video, audio) stream index.
- **deep_probe** (`bool`) – Whether to probe file deeply.
- **name** (`str`) – Optional; name of playlist file.

Return type`Path`**Returns**

Path to created ficoncat playlist.

probe_json(*entries*, *fileargs*, *streamtype=None*, *deep_probe=False*, *extraargs=None*)

Probes a media file with ffprobe and returns results.

Generic function for probing a media file for information using ffmpeg's *ffprobe* utility and returning its JSON.

Parameters

- **entries** (*str*) – Argument passed to the `-show_entries` flag in ffprobe.
- **fileargs** (*str* | *Iterable[str]* | *None*) – String of the file you want to analyze. If additional args are needed to specify the input, accepts a list of args to pass on.
- **streamtype** (*str* | *None*) – Optional; Argument to pass on to the `-select_streams` flag in ffprobe. Argument not passed if *None*.
- **deep_probe** (*bool*) – Optional; Pass extra arguments to ffprobe in order to probe the file more deeply. This is useful for containers that can't be lightly inspected.
- **extraargs** (*str* | *Iterable[str]* | *None*) – Optional; A list of additional arguments to pass to ffprobe during runtime. Can be used for example to request `-sexagesimal` formatting of duration fields.

Returns

The query failed or returned “unknown” or “N/A”.

deserialized JSON: For raw probetype, the JSON returned after deserialization.

Return type`None`

probe_val(*entry*, *fileargs*, *streamtype=None*, *probetype=ProbeType.STREAM*, *deep_probe=False*, *extraargs=None*)

Wrapper for probe_vals to probe a single val.

Parameters

- **entry** (*str*) –
- **fileargs** (*str* | *Iterable[str]* | *None*) –
- **streamtype** (*str* | *None*) –
- **probetype** (*ProbeType*) –
- **deep_probe** (*bool*) –
- **extraargs** (*str* | *Iterable[str]* | *None*) –

Return type`str` | `None`

probe_vals(*entries*, *fileargs*, *streamtype=None*, *probetype=ProbeType.STREAM*, *deep_probe=False*, *extraargs=None*)

Probes a media file with ffprobe and returns results.

Generic function for probing a media file for information using ffmpeg's *ffprobe* utility. Returns individual values.

Parameters

- **entries** (*Iterable*[*str*]) – List of arguments passed to the `-show_entries` flag in `ffprobe`. If a non-raw streamtype is specified, then the argument may be the type field you want to query, for example the duration.
- **fileargs** (*str* | *Iterable*[*str*] | *None*) – String of the file you want to analyze. If additional args are needed to specify the input, accepts a list of args to pass on.
- **streamtype** (*str* | *None*) – Optional; Argument to pass on to the `-select_streams` flag in `ffprobe`. Argument not passed if *None*.
- **proptype** (*ProbeType*) – Optional; One of `STREAM`, `TAGS`, `DISPOSITION`, `FORMAT`; query file for metadata of selected proptype and streamtype and return the requested entries of the first matching stream.
- **deep_probe** (*bool*) – Optional; Pass extra arguments to `ffprobe` in order to probe the file more deeply. This is useful for containers that can't be lightly inspected.
- **extraargs** (*str* | *Iterable*[*str*] | *None*) – Optional; A list of additional arguments to pass to `ffprobe` during runtime. Can be used for example to request `-sexagesimal` formatting of duration fields.

Returns

None: The query failed or returned “unknown” or “N/A”.

str: The value of the requested query.

Return type

list of values in order of entries iterable containing

Raises

ValueError – Invalid proptype was passed.

property protocols: *FFProtocols*

A *FFProtocols* of compiled in input/output protocols.

property sencoders: *set*[*str*]

Set of compiled-in subtitle encoders.

Return type

set[*str*]

property vencoders: *set*[*str*]

Set of compiled-in video encoders.

Return type

set[*str*]

property version: *dict*[*str*, *pyffstream.ffmpeg.FFVersion*]

Dict of *FFVersions* indexed by component name.

Return type

dict[*str*, *FFVersion*]

class `pyffstream.ffmpeg.FFEncoders`(*vencoders*, *aencoders*, *sencoders*)

aencoders: *set*[*str*]

Alias for field number 1

sencoders: `set[str]`

Alias for field number 2

vencoders: `set[str]`

Alias for field number 0

class `pyffstream.ffmpeg.FFProbeJSON`

class `pyffstream.ffmpeg.FFProtocols(inputs, outputs)`

inputs: `set[str]`

Alias for field number 0

outputs: `set[str]`

Alias for field number 1

class `pyffstream.ffmpeg.FFVersion(*args)`

Holds a ffmpeg component version.

Parameters

args (`str` | `int` | `FFVersion`) –

class `pyffstream.ffmpeg.Filter(filt, *filtopts, src=None, dst=None)`

A single ffmpeg filter.

Collects helper methods for constructing and rendering out ffmpeg filter strings for use with the CLI.

Parameters

- **filt** (`str` | `Filter`) –
- **filtopts** (`str`) –
- **src** (`Sequence[str | int | None]` | `None`) –
- **dst** (`Sequence[str | int | None]` | `None`) –

classmethod `complex_join(filterlist, startkey='v', endkey='b', basekey='c')`

Combine filters in a way compatible with -filter_complex.

Parameters

- **filterlist** (`Sequence[Filter | str]`) –
- **startkey** (`str`) –
- **endkey** (`str`) –
- **basekey** (`str`) –

Return type

`str`

classmethod `full_escape(val)`

Do full escaping needed for complex filter graph in ffmpeg.

<https://ffmpeg.org/ffmpeg-filters.html#Notes-on-filtergraph-escaping>

Parameters

val (`str`) –

Return type

`str`

static `vf_join(filterlist)`

Combine filters in a way compatible with -af/-vf.

Parameters

filterlist (*Sequence*[*str* | *Filter*]) –

Return type

str

class `pyffstream.ffmpeg.ProbeType(value)`

An enumeration.

class `pyffstream.ffmpeg.Progress`

Assists in monitoring the progress output of an ffmpeg encode.

`pyffstream.ffmpeg.duration(timestamp)`

Processes ffmpeg duration string into time in seconds.

<https://ffmpeg.org/ffmpeg-utils.html#Time-duration>

Parameters

timestamp (*str* | *float* | *int*) –

Return type

float

`pyffstream.ffmpeg.format_q_tuple(init_tuple, is_stream)`

Format the entries arg for raw JSON queries to the probefile.

This corresponds to the `-show_entries` flag in `ffprobe` and can be used generically to format arguments to it.

Parameters

- **init_tuple** (*InitTuple* | *None*) – If querying a stream, a 3-tuple of the stream values, stream tags, and stream dispositions to query `ffprobe` for. If querying a format, a list of the format values to query. Can be *None*.
- **is_stream** (*bool*) – A boolean indicating whether or not `init_tuple` is a 3-tuple for a stream or a list for a format.

Return type

str

Returns

A formatted query for `-show_entries` in `ffprobe`. An empty string if `init_tuple` is *None*.

`pyffstream.ffmpeg.num(val)`

Process input into float in a way that mimics ffmpeg.

Method follows ffmpeg's [numerical options](#). All whitespace is stripped, then valid input is a number followed optionally with an SI prefix that may be appended with an *i* modifier that indicates the SI prefix is in powers of 1024 instead of 1000. Finally, the number may end in a *B* indicating it is to be multiplied by 8. The optional ffmpeg utility `ffeval` may be used to validate the output of this function.

Parameters

val (*str*) –

Return type

float

`pyffstream.ffmpeg.single_quote(unescape)`
Return a single-quote escaped string from input.

Parameters

unescape (*object*) –

Return type

str

1.2 pyffstream.encode

Classes and functions for ffmpeg encoding.

class `pyffstream.encode.EncType(value)`

An enumeration.

class `pyffstream.encode.FileOpts(main, subtitle, fpath, sopath, allpaths)`

allpaths: *Sequence*[*Path*]

Alias for field number 4

fpath: *Path*

Alias for field number 2

main: *MutableSequence*[*str*]

Alias for field number 0

sopath: *Path*

Alias for field number 3

subtitle: *MutableSequence*[*str*]

Alias for field number 1

class `pyffstream.encode.FilterList`

Thread-safe class for storing and getting ffmpeg filters.

get(*k*, *default=None*)

An implementation of the get method as in dicts.

Parameters

- **k** (*str*) –
- **default** (*ffmpeg.Filter* | *str* | *None*) –

Return type

str | *None*

if_exists(*key*)

Return 1-tuple of filter at key if it exists, else 0-tuple.

The intent is to use this function starred in-line like `*if_exists(key)` when defining a list to simplify creation of filter lists.

Parameters

key (*str*) –

Return type

tuple[*str*] | *tuple*[]

```
class pyffstream.encode.StaticEncodeVars(tempdir, verbosity=0, api_url="", api_key="",
                                         samplerate='48000', endpoint='127.0.0.1:9998',
                                         target_w='1920', target_h='1080', framerate='24/1',
                                         framerate_multiplier=Fraction(1, 1), scale_w='1920',
                                         scale_h='1080', bound_w='1920', bound_h='1080',
                                         kf_target_sec=5.0, clip_length=None,
                                         vencoder=VEncoder(name='libx264', codec='h264',
                                         flag_function=<function get_libx264_flags>,
                                         presets=['placebo', 'veryslow', 'slower', 'slow', 'medium', 'fast',
                                         'faster', 'veryfast', 'superfast', 'ultrafast'],
                                         default_preset='medium', tenbit=False,
                                         type=<EncType.SOFTWARE: 1>, multipass=True),
                                         aencoder='aac', encode_preset=None, encode_tune=None,
                                         realtime=False, vstandard='h264', astandard='aac',
                                         include_audio=True, protocol='srt', vbitrate='6M',
                                         max_vbitrate='0', rc_mode='cbr', abitrage='256k',
                                         chlayout='stereo', start_delay='30', end_pad=True,
                                         end_delay='600', timestamp=None, suboffset=None,
                                         cleanborders=None, crop_ts='600', crop_len='60',
                                         target_i='-19', target_lra='11.0', target_tp='-1.0',
                                         pix_fmt='yuv420p', subfile_provided=False, text_subs=True,
                                         subfilter_list=<factory>, kf_int='0', min_kf_int='0',
                                         bufsize='0', kf_sec='0', latency_target='0', afilters="",
                                         filter_complex="", input_flags=<factory>,
                                         encode_flags=<factory>, filter_flags=<factory>,
                                         output_flags=<factory>, shader_list=<factory>,
                                         ff_flags=<factory>, srt_passphrase="", srt_latency=5.0,
                                         ff_verbosity_flags=<factory>, ff_deepprobe_flags=<factory>,
                                         placebo_opts=<factory>, sw_filters=<factory>,
                                         vencoder_params=<factory>, copy_audio=False,
                                         copy_video=False, use_timeline=False, hwaccel=True,
                                         subs=False, deep_probe=False, vindex=0, aindex=0,
                                         sindex=None, outfile=None, npass=None, passfile=None,
                                         wait=False, overwrite=False, fifo=False, soxr=False,
                                         zscale=False, slowseek=False, live=False, obs=False,
                                         vulkan=False, vgap=False, trust_vulkan=False,
                                         vulkan_device=-1, decimate_target='24/1', is_playlist=False,
                                         eightbit=False, cropsecond=False, subcropfirst=False,
                                         picsubscale='bicubic', delay_start=False, deinterlace=False,
                                         crop=False, upscale=False, anormalize=False,
                                         normfile=None, dynamicnorm=False, fix_start_time=True,
                                         pyffserver=False, ffprogress=<factory>)
```

Class holding general encoding parameters.

Needs to be passed to an encode session to initialize it.

Parameters

- **tempdir** (*pathlib.Path*) –
- **verbosity** (*int*) –
- **api_url** (*str*) –
- **api_key** (*str*) –
- **samplerate** (*str*) –

- `endpoint(str)` –
- `target_w(str)` –
- `target_h(str)` –
- `framerate(str)` –
- `framerate_multiplier(fractions.Fraction)` –
- `scale_w(str)` –
- `scale_h(str)` –
- `bound_w(str)` –
- `bound_h(str)` –
- `kf_target_sec(float)` –
- `clip_length(str | None)` –
- `vencoder(VEncoder)` –
- `aencoder(str)` –
- `encode_preset(str | None)` –
- `encode_tune(str | None)` –
- `realtime(bool)` –
- `vstandard(str)` –
- `astandard(str)` –
- `include_audio(bool)` –
- `protocol(str)` –
- `vbitrate(str)` –
- `max_vbitrate(str)` –
- `rc_mode(str)` –
- `abitrage(str)` –
- `chlayout(str)` –
- `start_delay(str)` –
- `end_pad(bool)` –
- `end_delay(str)` –
- `timestamp(str | None)` –
- `suboffset(str | None)` –
- `cleanborders(list[int] | None)` –
- `crop_ts(str)` –
- `crop_len(str)` –
- `target_i(str)` –
- `target_lra(str)` –
- `target_tp(str)` –

- `pix_fmt(str)` –
- `subfile_provided(bool)` –
- `text_subs(bool)` –
- `subfilter_list(Sequence[ffmpeg.Filter])` –
- `kf_int(str)` –
- `min_kf_int(str)` –
- `bufsize(str)` –
- `kf_sec(str)` –
- `latency_target(str)` –
- `afilters(str)` –
- `filter_complex(str)` –
- `input_flags(MutableSequence[str])` –
- `encode_flags(MutableSequence[str])` –
- `filter_flags(MutableSequence[str])` –
- `output_flags(MutableSequence[str])` –
- `shader_list(MutableSequence[str])` –
- `ff_flags(MutableSequence[str])` –
- `srt_passphrase(str)` –
- `srt_latency(float)` –
- `ff_verbosity_flags(Sequence[str])` –
- `ff_deepprobe_flags(Sequence[str])` –
- `placebo_opts(Sequence[str])` –
- `sw_filters(Sequence[str])` –
- `vencoder_params(Sequence[str])` –
- `copy_audio(bool)` –
- `copy_video(bool)` –
- `use_timeline(bool)` –
- `hwaccel(bool)` –
- `subs(bool)` –
- `deep_probe(bool)` –
- `vindex(int)` –
- `aindex(int)` –
- `sindex(int | None)` –
- `outfile(pathlib.Path | None)` –
- `npass(int | None)` –
- `passfile(pathlib.Path | None)` –

- `wait (bool)` –
- `overwrite (bool)` –
- `fifo (bool)` –
- `soxr (bool)` –
- `zscale (bool)` –
- `slowseek (bool)` –
- `live (bool)` –
- `obs (bool)` –
- `vulkan (bool)` –
- `vgop (bool)` –
- `trust_vulkan (bool)` –
- `vulkan_device (int)` –
- `decimate_target (str)` –
- `is_playlist (bool)` –
- `eightbit (bool)` –
- `cropsecond (bool)` –
- `subcropfirst (bool)` –
- `picsubscale (str)` –
- `delay_start (bool)` –
- `deinterlace (bool)` –
- `crop (bool)` –
- `upscale (bool)` –
- `anormalize (bool)` –
- `normfile (pathlib.Path | None)` –
- `dynamicnorm (bool)` –
- `fix_start_time (bool)` –
- `pyffserver (bool)` –
- `ffprogress (ffmpeg.Progress[str])` –

classmethod `from_args(args)`

Constructor to make StaticEncodeVars from passed args.

Parameters

`args (Namespace)` –

Return type

`StaticEncodeVars`

class `pyffstream.encode.StatusCode(value)`

An enumeration.

```
class pyffstream.encode.StyleFile(names, lines, insert_index)
```

```
    insert_index: int
```

```
        Alias for field number 2
```

```
    lines: MutableSequence[str]
```

```
        Alias for field number 1
```

```
    names: MutableSequence[str]
```

```
        Alias for field number 0
```

```
class pyffstream.encode.VEncoder(name, codec, flag_function, presets=<factory>, default_preset=None,
                                tenbit=False, type=EncType.SOFTWARE, multipass=False)
```

Parameters

- **name** (*str*) –
- **codec** (*str*) –
- **flag_function** (*Callable*[[*EncodeSession*], *list*[*str*]]) –
- **presets** (*Iterable*[*object*]) –
- **default_preset** (*str* | *None*) –
- **tenbit** (*bool*) –
- **type** (*EncType*) –
- **multipass** (*bool*) –

```
pyffstream.encode.close_futures(futures)
```

```
    Wait on futures in list, then raise any exceptions in them.
```

Parameters

```
    futures (Iterable[Future[Any]]) –
```

Return type

```
    None
```

1.3 pyffstream.cli

CLI frontend for encoding and streaming.

```
class pyffstream.cli.ConfName(file_name, arg_name)
```

```
    arg_name: str
```

```
        Alias for field number 1
```

```
    file_name: str
```

```
        Alias for field number 0
```

```
class pyffstream.cli.DefaultConfig(pyffserver=False, protocol='srt', vbitrate='6M', max_vbitrate='0',
                                   abitrage='256k', aencoder='aac', vencoder='libx264',
                                   endpoint='127.0.0.1:9998', api_url='', api_key='', soxr=False,
                                   preset=<factory>, zscale=False, vulkan=False, trust_vulkan=False,
                                   vgop=False, vulkan_device=-1, hwaccel=True, height=1080,
                                   shader_list=<factory>, kf_target_sec=5.0, srt_latency=5.0,
                                   ffmpeg_bin='ffmpeg', ffprobe_bin='ffprobe', env=<factory>)
```

Holds config file values.

Used to determine the default CLI parameters after processing config from files.

Parameters

- **pyffserver** (*bool*) –
- **protocol** (*str*) –
- **vbitrate** (*str*) –
- **max_vbitrate** (*str*) –
- **abitrage** (*str*) –
- **aencoder** (*str*) –
- **vencoder** (*str*) –
- **endpoint** (*str*) –
- **api_url** (*str*) –
- **api_key** (*str*) –
- **soxr** (*bool*) –
- **preset** (*dict*[*str*, *str* | *None*]) –
- **zscale** (*bool*) –
- **vulkan** (*bool*) –
- **trust_vulkan** (*bool*) –
- **vgop** (*bool*) –
- **vulkan_device** (*int*) –
- **hwaccel** (*bool*) –
- **height** (*int*) –
- **shader_list** (*list*[*str*]) –
- **kf_target_sec** (*float*) –
- **srt_latency** (*float*) –
- **ffmpeg_bin** (*str*) –
- **ffprobe_bin** (*str*) –
- **env** (*dict*[*str*, *str*]) –

`pyffstream.cli.download_win_ffmpeg(dltype='git')`

Download and install ffmpeg for windows in user_data_path.

The current ffmpeg in the location is replaced if already there. User data path is determined from platformdirs.

Parameters

dltype (*str*) –

Return type

bool

`pyffstream.cli.get_parserconfig(reproducible=True)`

Return parser and config used.

Parameters

reproducible (`bool`) –

Return type

`tuple[ArgumentParser, DefaultConfig]`

`pyffstream.cli.get_stream_list(streamtype, q_tuple, myfileargs, deep_probe=False)`

Make and return tuples of (key,val) pairs for each stream.

Parameters

- **streamtype** (`str`) –
- **q_tuple** (`tuple[Collection[str], Collection[str], Collection[str]]`) –
- **myfileargs** (`Sequence[str]`) –
- **deep_probe** (`bool`) –

Return type

`list[list[tuple[str, str]]]`

`pyffstream.cli.main()`

Process config and CLI arguments then send off for processing.

Return type

`None`

`pyffstream.cli.parse_files(args, parser)`

Process input arguments and send them off processing.

Parameters

- **args** (`Namespace`) –
- **parser** (`ArgumentParser`) –

Return type

`None`

`pyffstream.cli.print_info(fopts, *, verbosity=0, deep_probe=False)`

Prints to console formatted information about the input file.

Output is nicely formatted for console usage using rich tables.

Parameters

- **fopts** (`FileOpts`) – The file to print information about.
- **verbosity** (`int`) – Print extra fields based on size.
- **deep_probe** (`bool`) – Whether or not to probe the file deeply.

Return type

`None`

`pyffstream.cli.process_file(fpath, args, stream_flist)`

Format input arguments needed for a file and send to output.

Parameters

- **fpath** (`Path`) –

- **args** (`Namespace`) –
- **stream_list** (`Sequence[Path]`) –

Return type`None``pyffstream.cli.set_console_logger(verbosity, logfile)`

Set loglevel.

Parameters

- **verbosity** (`int`) –
- **logfile** (`pathlib.Path` | `None`) –

Return type`None``pyffstream.cli.setup_pyffserver_stream(fv)`

Communicate with a pyffserver API to set up encode session.

Parameters**fv** (`EncodeSession`) –**Return type**`None``pyffstream.cli.start_stream(fv)`

Start and track the actual encode.

Parameters**fv** (`EncodeSession`) –**Return type**`None``pyffstream.cli.status_wait(fv, futures)`

Wait on remaining background processes while showing status.

Parameters

- **fv** (`EncodeSession`) –
- **futures** (`Iterable[Future[Any]]`) –

Return type`None``pyffstream.cli.stream_file(fopts, args)`

Manage calculating of all stream parameters.

Parameters

- **fopts** (`FileOpts`) –
- **args** (`Namespace`) –

Return type`None``pyffstream.cli.win_set_local_ffmpeg(dltype, env)`

Set the ffmpeg instance to the app-local Windows copy.

If ffmpeg is not already available in `user_data_path`, offer to download it from a public repository.

Parameters

- **dtype** (`str`) –
- **env** (`dict[str, str]`) –

Return type

`None`

LICENSE**MIT License**

Copyright (c) 2021, 2022 Gregory Beauregard

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

A CLI wrapper for ffmpeg to stream files over SRT/RTMP. Also supports the api for a not (yet) open sourced pyffserver endpoint.

INSTALLATION

To install pyffstream download a binary from [Github releases](#), or run this command in your terminal:

```
$ pip install pyffstream
```


CLI USAGE

4.1 pyffstream

CLI frontend for streaming over SRT and RTMP.

```
usage: pyffstream [-h] [--config FILE] [--show-config-dirs] [-v]
                  [--logfile FILE] [-b BITRATE] [-M BITRATE] [-A BITRATE]
                  [--aencoder {aac,libfdk_aac,libopus}] [-o] [-p] [-i]
                  [-t TIMESTAMP] [-I OPT] [--pyffserver | --no-pyffserver]
                  [-y] [--srt-passphrase PASSWORD] [--srt-latency SEC]
                  [--protocol {rtmp,srt}] [-U URL] [-k KEY]
                  [-E DOMAIN[:PORT][PATH]] [-w] [-B]
                  [--hwaccel | --no-hwaccel] [-e] [--subfile FILE] [-s N]
                  [--suboffset TIMESTAMP] [-a N] [--vindex N] [--live]
                  [--fix-start-time | --no-fix-start-time]
                  [--soxr | --no-soxr] [--fifo | --no-fifo]
                  [--zscale | --no-zscale] [--fdk] [-f FILE] [-T LENGTH] [-D]
                  [--copy-audio] [--copy-video] [-c COPY] [-H] [-8]
                  [--h264-nvenc] [-x] [--preset PRESET] [--tune TUNE]
                  [--realtime | --no-realtime] [--pass N] [--passfile FILE]
                  [--vgop | --no-vgop]
                  [--vencoder {h264_nvenc,hevc_nvenc,libaom-av1,librav1e,libsvtav1,
↪ libvpx-vp9,libx264,libx265}]
                  [-u] [--slowseek] [-d]
                  [--framerate-multiplier FRAMERATE_MULTIPLIER] [-C]
                  [--croptime TIMESTAMP]
                  [--cleanborders LEFT RIGHT TOP BOTTOM]
                  [--croplength DURATION] [-V | --vulkan | --no-vulkan]
                  [--trust-vulkan | --no-trust-vulkan] [--vulkan-device NUM]
                  [--sw-filters FILTER] [-P OPT] [--vencoder-params PARAM]
                  [-z] [--subfirst | --no-subfirst] [--picsubscale ALGORITHM]
                  [--nodecimate | --paldecimate | --sixtyfps]
                  [--audio | --no-audio] [-n] [-N FILE | -Q]
                  [--height HEIGHT | -4 | -2 | -7] [-K SEC] [--mono]
                  [--startdelay] [--endpad | --no-endpad] [--tempdir DIR]
                  [--shaders PATH_TO_SHADER]
                  [--system-ffmpeg | --downloaded-ffmpeg] [--redownload]
                  [--dltype {git,stable}] [--write]
                  [FILES ...]
```

4.1.1 optional arguments

- h, --help**
show this help message and exit
- config <file>**
Path to config file
- show-config-dirs**
Print out config search locations
- v, --verbose**
increase verbosity level
- logfile <file>**
path to logfile to send output to
- p, --playlist**
make ffconcat playlist from input files
- i, --print-info**
print information about input file(s) instead of streaming
- t <timestamp>, --timestamp <timestamp>**
timestamp to start stream from
- I <opt>, --preinput-opts <opt>**
pass option ffmpeg before input (specify once/item)
- w, --wait**
wait for keypress before starting stream
- fix-start-time, --no-fix-start-time**
Fix start_time of streams (default: True)
- c <copy>, --copy <copy>**
pass a/v to copy audio/video
- slowseek**
use slow ffmpeg seeking
- startdelay**
delay stream start by 30 seconds
- endpad, --no-endpad**
Pad end of stream with nothing to prevent early stream cutoff. (default: True)
- tempdir <dir>**
directory to use for storing temporary files
- shaders <path_to_shader>**
shader to use with vulkan (specify once for each shader to add) (default: [])
- system-ffmpeg**
use system ffmpeg binaries instead of configured (default is system if unconfigured)

--downloaded-ffmpeg

Use downloaded local Windows ffmpeg instead of configured or system ffmpeg (default is to only use as a fallback)

--redownload

Redownload stored local Windows ffmpeg binaries

--dltype {git,stable}

Type of Windows ffmpeg binary to download (default: git)

--write

write chosen arguments as defaults to config if not already default

4.1.2 input arguments

files

list of input files and directories; if last argument is file already contained in input list, start list from that file

-o, --obs

get input from OBS pipe

-B, --bluray

input directory is bluray

--hwaccel, --no-hwaccel

Attempt to automatically use hw accelerated decoding if available (default: True)

--live

hint that input is live

-D, --deep-probe

pass extra args to probe input file deeper

--nodecimate

don't decimate 30 fps obs input to 24 fps

--paldecimate

decimate 30 fps obs input to 25 fps

--sixtyfps

don't halve 60 fps obs input to 30 fps

4.1.3 video arguments

-b <bitrate>, --vbitrate <bitrate>

encoding video bitrate (ffmpeg num) (default: 6M)

-M <bitrate>, --max-vbitrate <bitrate>

max encoding video bitrate (ffmpeg num) (default: vbitrate)

--vindex <n>

subindex of video stream to use (default: 0)

--zscale, --no-zscale

Use zimg library for scaling instead of ffmpeg's scale (default: False)

--copy-video
copy video stream from input

-H, --hevc-nvenc
encode with NVENC HEVC

-8, --eightbit
encode with 8-bit HEVC (default 10-bit)

--h264-nvenc
encode with NVENC H264

-x, --x264
encode with x264

--preset <preset>
preset to use for encoding

--tune <tune>
tune parameter to use for supported encoders

--realtime, --no-realtime
tune for realtime encoding on encoders that support it (default: False)

--pass {1,2,3}
encoder pass

--passfile <file>
multipass statistics file to use

--vgop, --no-vgop
use variable GOP length and treat keyframe target as max (default: False)

--encoder {h264_nvenc,hevc_nvenc,libaom-av1,librav1e,libsvtav1,libvpx-vp9,libx264,libx265}
video encoder to use (default: libx264)

-u, --upscale
unconditionally scale video to target size

-d, --deinterlace
deinterlace video

--framerate-multiplier <framerate_multiplier>
fraction to multiply framerate by (e.g. because a filter modifies it)

-C, --crop
automatically crop video

--croptime <timestamp>
timestamp to start crop calculation at (default: 600)

--cleanborders <left> <right> <top> <bottom>
Clean up N pixels of border after crop

--croplength <duration>
duration to estimate crop for (default: 60)

-V, --vulkan, --no-vulkan
 use vulkan processing path (default: False)

--trust-vulkan, --no-trust-vulkan
 whether to trust ffmpeg vulkan to do the right thing or use workarounds (default: False)

--vulkan-device <num>
 GPU device listing to use for vulkan HW context

--sw-filters <filter>
 pass software ffmpeg filter to filter chain start (specify once/filter)

-P <opt>, --placebo-opts <opt>
 pass option to vf_libplacebo when using vulkan (specify once/opt)

--vencoder-params <param>
 pass option to params argument of encoders (specify once/param)

--height <height>
 target 16:9 bounding box encode height (default: 1080)

-4, --res2160
 set 4k encoding resolution

-2, --res1440
 set 1440p encoding resolution

-7, --res720
 set 720p encoding resolution

4.1.4 audio arguments

-A <bitrate>, --abitrage <bitrate>
 encoding audio bitrate (ffmpeg num) (default: 256k)

--aencoder {aac,libfdk_aac,libopus}
 audio encoder to use (default: aac)

-a <n>, --aindex <n>
 subindex of audio stream to use (default: 0)

--soxr, --no-soxr
 Use SoX resampler library instead of ffmpeg's avresample (default: False)

--fdk
 Use libfdk_aac encoder

--copy-audio
 copy audio stream from input

--audio, --no-audio
 attempt to include audio in output (default: True) (default: True)

-n, --anormalize
 normalize audio (implied by -N and -Q)

-N <file>, --normfile <file>
path to file to store audio normalization data

-Q, --dynamicnorm
do one-pass audio normalization

--mono
output audio in mono

4.1.5 subtitle arguments

-e, --subs
enable subtitles

--subfile <file>
path to external subtitles

-s <n>, --sindex <n>
subindex of subtitle stream to use

--suboffset <timestamp>
timestamp to offset subtitle timing by

-z, --cropsecond
crop after subtitles are rendered

--subfirst, --no-subfirst
render subtitles and crop video before scaling (default: False)

--picsubscale <algorithm>
algorithm to use for scaling picture subtitles (default: bicubic)

4.1.6 output arguments

--pyffserver, --no-pyffserver
use pyffserver as an API to send to (default: False)

-y, --overwrite
overwrite output file if it already exists

--srt-passphrase <password>
optional passphrase to use for SRT when not streaming to a pyffserver

--srt-latency <sec>
SRT latency (default: 5.0)

--protocol {rtmp,srt}
streaming protocol to use (default: srt)

-U <url>, --api-url <url>
pyffserver API URL to use (default: from config)

-k <key>, --api-key <key>
pyffserver API key to use (default: from config)

-E <domain[:port][path]>, **--endpoint** <domain[:port][path]>
endpoint to stream to without protocol (default: from config)

--fifo, **--no-fifo**
Use FIFO to try to sustain and stabilize the connection. (default: False)

-f <file>, **--outfile** <file>
path to an output file to use instead of streaming

-T <length>, **--cliplength** <length>
clip stream to this length

-K <sec>, **--keyframe-target-sec** <sec>
target keyframe interval in seconds (default: 5.0)

PYTHON MODULE INDEX

p

`pyffstream.cli`, [12](#)
`pyffstream.encode`, [7](#)
`pyffstream.ffmpeg`, [1](#)

Symbols

- 2
 - pyffstream command line option, 25
- 4
 - pyffstream command line option, 25
- 7
 - pyffstream command line option, 25
- 8
 - pyffstream command line option, 24
- A
 - pyffstream command line option, 25
- B
 - pyffstream command line option, 23
- C
 - pyffstream command line option, 24
- D
 - pyffstream command line option, 23
- E
 - pyffstream command line option, 26
- H
 - pyffstream command line option, 24
- I
 - pyffstream command line option, 22
- K
 - pyffstream command line option, 27
- M
 - pyffstream command line option, 23
- N
 - pyffstream command line option, 25
- P
 - pyffstream command line option, 25
- Q
 - pyffstream command line option, 26
- T
 - pyffstream command line option, 27
- U
 - pyffstream command line option, 26
- V
 - pyffstream command line option, 24
- abitrage
 - pyffstream command line option, 25
- aencoder
 - pyffstream command line option, 25
- aindex
 - pyffstream command line option, 25
- anormalize
 - pyffstream command line option, 25
- api-key
 - pyffstream command line option, 26
- api-url
 - pyffstream command line option, 26
- audio
 - pyffstream command line option, 25
- bluray
 - pyffstream command line option, 23
- cleanborders
 - pyffstream command line option, 24
- cliplength
 - pyffstream command line option, 27
- config
 - pyffstream command line option, 22
- copy
 - pyffstream command line option, 22
- copy-audio
 - pyffstream command line option, 25
- copy-video
 - pyffstream command line option, 23
- crop
 - pyffstream command line option, 24
- croplength
 - pyffstream command line option, 24
- cropsecond
 - pyffstream command line option, 26
- croptime
 - pyffstream command line option, 24
- deep-probe
 - pyffstream command line option, 23
- deinterlace
 - pyffstream command line option, 24
- dltype
 - pyffstream command line option, 23
- downloaded-ffmpeg
 - pyffstream command line option, 22
- dynamicnorm

pyffstream command line option, 26	pyffstream command line option, 25
--eightbit	--no-vgop
pyffstream command line option, 24	pyffstream command line option, 24
--endpad	--no-vulkan
pyffstream command line option, 22	pyffstream command line option, 24
--endpoint	--no-zscale
pyffstream command line option, 26	pyffstream command line option, 23
--fdk	--nodecimate
pyffstream command line option, 25	pyffstream command line option, 23
--fifo	--normfile
pyffstream command line option, 27	pyffstream command line option, 25
--fix-start-time	--obs
pyffstream command line option, 22	pyffstream command line option, 23
--framerate-multiplier	--outfile
pyffstream command line option, 24	pyffstream command line option, 27
--h264-nvenc	--overwrite
pyffstream command line option, 24	pyffstream command line option, 26
--height	--paldecimate
pyffstream command line option, 25	pyffstream command line option, 23
--help	--pass
pyffstream command line option, 22	pyffstream command line option, 24
--hevc-nvenc	--passfile
pyffstream command line option, 24	pyffstream command line option, 24
--hwaccel	--picsubscale
pyffstream command line option, 23	pyffstream command line option, 26
--keyframe-target-sec	--placebo-opts
pyffstream command line option, 27	pyffstream command line option, 25
--live	--playlist
pyffstream command line option, 23	pyffstream command line option, 22
--logfile	--preinput-opts
pyffstream command line option, 22	pyffstream command line option, 22
--max-vbtrate	--preset
pyffstream command line option, 23	pyffstream command line option, 24
--mono	--print-info
pyffstream command line option, 26	pyffstream command line option, 22
--no-audio	--protocol
pyffstream command line option, 25	pyffstream command line option, 26
--no-endpad	--pyffserver
pyffstream command line option, 22	pyffstream command line option, 26
--no-fifo	--realtime
pyffstream command line option, 27	pyffstream command line option, 24
--no-fix-start-time	--redownload
pyffstream command line option, 22	pyffstream command line option, 23
--no-hwaccel	--res1440
pyffstream command line option, 23	pyffstream command line option, 25
--no-pyffserver	--res2160
pyffstream command line option, 26	pyffstream command line option, 25
--no-realtime	--res720
pyffstream command line option, 24	pyffstream command line option, 25
--no-soxr	--shaders
pyffstream command line option, 25	pyffstream command line option, 22
--no-subfirst	--show-config-dirs
pyffstream command line option, 26	pyffstream command line option, 22
--no-trust-vulkan	--sindex

pyffstream command line option, 26
 --sixtyfps
 pyffstream command line option, 23
 --slowseek
 pyffstream command line option, 22
 --soxr
 pyffstream command line option, 25
 --srt-latency
 pyffstream command line option, 26
 --srt-passphrase
 pyffstream command line option, 26
 --startdelay
 pyffstream command line option, 22
 --subfile
 pyffstream command line option, 26
 --subfirst
 pyffstream command line option, 26
 --suboffset
 pyffstream command line option, 26
 --subs
 pyffstream command line option, 26
 --sw-filters
 pyffstream command line option, 25
 --system-ffmpeg
 pyffstream command line option, 22
 --tempdir
 pyffstream command line option, 22
 --timestamp
 pyffstream command line option, 22
 --trust-vulkan
 pyffstream command line option, 25
 --tune
 pyffstream command line option, 24
 --upscale
 pyffstream command line option, 24
 --vbitrate
 pyffstream command line option, 23
 --vencoder
 pyffstream command line option, 24
 --vencoder-params
 pyffstream command line option, 25
 --verbose
 pyffstream command line option, 22
 --vgop
 pyffstream command line option, 24
 --vindex
 pyffstream command line option, 23
 --vulkan
 pyffstream command line option, 24
 --vulkan-device
 pyffstream command line option, 25
 --wait
 pyffstream command line option, 22
 --write

pyffstream command line option, 23
 --x264
 pyffstream command line option, 24
 --zscale
 pyffstream command line option, 23
 -a
 pyffstream command line option, 25
 -b
 pyffstream command line option, 23
 -c
 pyffstream command line option, 22
 -d
 pyffstream command line option, 24
 -e
 pyffstream command line option, 26
 -f
 pyffstream command line option, 27
 -h
 pyffstream command line option, 22
 -i
 pyffstream command line option, 22
 -k
 pyffstream command line option, 26
 -n
 pyffstream command line option, 25
 -o
 pyffstream command line option, 23
 -p
 pyffstream command line option, 22
 -s
 pyffstream command line option, 26
 -t
 pyffstream command line option, 22
 -u
 pyffstream command line option, 24
 -v
 pyffstream command line option, 22
 -w
 pyffstream command line option, 22
 -x
 pyffstream command line option, 24
 -y
 pyffstream command line option, 26
 -z
 pyffstream command line option, 26

A

aencoders (*pyffstream.ffmpeg.FFBin* property), 2
 aencoders (*pyffstream.ffmpeg.FFEncoders* attribute), 4
 allpaths (*pyffstream.encode.FileOpts* attribute), 7
 arg_name (*pyffstream.cli.ConfName* attribute), 12

B

build_config (*pyffstream.ffmpeg.FFBin* property), 2

C

`close_futures()` (in module `pyffstream.encode`), 12
`complex_join()` (`pyffstream.ffmpeg.Filter` class method), 5
`ConfName` (class in `pyffstream.cli`), 12

D

`DefaultConfig` (class in `pyffstream.cli`), 12
`download_win_ffmpeg()` (in module `pyffstream.cli`), 13
`duration()` (in module `pyffstream.ffmpeg`), 6

E

`EncType` (class in `pyffstream.encode`), 7
`env` (`pyffstream.ffmpeg.FFBin` attribute), 2

F

`ff_bin` (in module `pyffstream.ffmpeg`), 1
`FFBanner` (class in `pyffstream.ffmpeg`), 1
`FFBin` (class in `pyffstream.ffmpeg`), 1
`ffconfig` (`pyffstream.ffmpeg.FFBanner` attribute), 1
`FFEncoders` (class in `pyffstream.ffmpeg`), 4
`ffmpeg` (`pyffstream.ffmpeg.FFBin` attribute), 1
`ffprobe` (`pyffstream.ffmpeg.FFBin` attribute), 1
`FFProbeJSON` (class in `pyffstream.ffmpeg`), 5
`FFProtocols` (class in `pyffstream.ffmpeg`), 5
`FFVersion` (class in `pyffstream.ffmpeg`), 5
`ffversion` (`pyffstream.ffmpeg.FFBanner` attribute), 1
`ffversion` (`pyffstream.ffmpeg.FFBin` property), 2
`file_name` (`pyffstream.cli.ConfName` attribute), 12
`FileOpts` (class in `pyffstream.encode`), 7
`files`
 `pyffstream` command line option, 23
`Filter` (class in `pyffstream.ffmpeg`), 5
`FilterList` (class in `pyffstream.encode`), 7
`filters` (`pyffstream.ffmpeg.FFBin` property), 2
`format_q_tuple()` (in module `pyffstream.ffmpeg`), 6
`fpath` (`pyffstream.encode.FileOpts` attribute), 7
`from_args()` (`pyffstream.encode.StaticEncodeVars` class method), 11
`full_escape()` (`pyffstream.ffmpeg.Filter` class method), 5

G

`get()` (`pyffstream.encode.FilterList` method), 7
`get_parserconfig()` (in module `pyffstream.cli`), 13
`get_stream_list()` (in module `pyffstream.cli`), 14

H

`hwaccels` (`pyffstream.ffmpeg.FFBin` property), 2

I

`if_exists()` (`pyffstream.encode.FilterList` method), 7
`inputs` (`pyffstream.ffmpeg.FFProtocols` attribute), 5

`insert_index` (`pyffstream.encode.StyleFile` attribute), 12

L

`lines` (`pyffstream.encode.StyleFile` attribute), 12

M

`main` (`pyffstream.encode.FileOpts` attribute), 7
`main()` (in module `pyffstream.cli`), 14
`make_playlist()` (`pyffstream.ffmpeg.FFBin` method), 2
module
 `pyffstream.cli`, 12
 `pyffstream.encode`, 7
 `pyffstream.ffmpeg`, 1

N

`names` (`pyffstream.encode.StyleFile` attribute), 12
`num()` (in module `pyffstream.ffmpeg`), 6

O

`outputs` (`pyffstream.ffmpeg.FFProtocols` attribute), 5

P

`parse_files()` (in module `pyffstream.cli`), 14
`print_info()` (in module `pyffstream.cli`), 14
`probe_json()` (`pyffstream.ffmpeg.FFBin` method), 3
`probe_val()` (`pyffstream.ffmpeg.FFBin` method), 3
`probe_vals()` (`pyffstream.ffmpeg.FFBin` method), 3
`ProbeType` (class in `pyffstream.ffmpeg`), 6
`process_file()` (in module `pyffstream.cli`), 14
`Progress` (class in `pyffstream.ffmpeg`), 6
`protocols` (`pyffstream.ffmpeg.FFBin` property), 4
`pyffstream` command line option
 -2, 25
 -4, 25
 -7, 25
 -8, 24
 -A, 25
 -B, 23
 -C, 24
 -D, 23
 -E, 26
 -H, 24
 -I, 22
 -K, 27
 -M, 23
 -N, 25
 -P, 25
 -Q, 26
 -T, 27
 -U, 26
 -V, 24
 --abitrage, 25

```

--aencoder, 25
--aindex, 25
--anormalize, 25
--api-key, 26
--api-url, 26
--audio, 25
--bluray, 23
--cleanborders, 24
--cliplength, 27
--config, 22
--copy, 22
--copy-audio, 25
--copy-video, 23
--crop, 24
--croplength, 24
--cropsecond, 26
--croptime, 24
--deep-probe, 23
--deinterlace, 24
--dltype, 23
--downloaded-ffmpeg, 22
--dynamicnorm, 26
--eightbit, 24
--endpad, 22
--endpoint, 26
--fdk, 25
--fifo, 27
--fix-start-time, 22
--framerate-multiplier, 24
--h264-nvenc, 24
--height, 25
--help, 22
--hevc-nvenc, 24
--hwaccel, 23
--keyframe-target-sec, 27
--live, 23
--logfile, 22
--max-vbtrate, 23
--mono, 26
--no-audio, 25
--no-endpad, 22
--no-fifo, 27
--no-fix-start-time, 22
--no-hwaccel, 23
--no-pyffserver, 26
--no-realtime, 24
--no-soxr, 25
--no-subfirst, 26
--no-trust-vulkan, 25
--no-vgop, 24
--no-vulkan, 24
--no-zscale, 23
--nodecimate, 23
--normfile, 25
--obs, 23
--outfile, 27
--overwrite, 26
--paldecimate, 23
--pass, 24
--passfile, 24
--picsubscale, 26
--placebo-opts, 25
--playlist, 22
--preinput-opts, 22
--preset, 24
--print-info, 22
--protocol, 26
--pyffserver, 26
--realtime, 24
--redownload, 23
--res1440, 25
--res2160, 25
--res720, 25
--shaders, 22
--show-config-dirs, 22
--sindex, 26
--sixtyfps, 23
--slowseek, 22
--soxr, 25
--srt-latency, 26
--srt-passphrase, 26
--startdelay, 22
--subfile, 26
--subfirst, 26
--suboffset, 26
--subs, 26
--sw-filters, 25
--system-ffmpeg, 22
--tempdir, 22
--timestamp, 22
--trust-vulkan, 25
--tune, 24
--upscale, 24
--vbitrate, 23
--vencoder, 24
--vencoder-params, 25
--verbose, 22
--vgop, 24
--vindex, 23
--vulkan, 24
--vulkan-device, 25
--wait, 22
--write, 23
--x264, 24
--zscale, 23
-a, 25
-b, 23
-c, 22

```

- d, 24
- e, 26
- f, 27
- h, 22
- i, 22
- k, 26
- n, 25
- o, 23
- p, 22
- s, 26
- t, 22
- u, 24
- v, 22
- w, 22
- x, 24
- y, 26
- z, 26

- files, 23

- pyffstream.cli
 - module, 12

- pyffstream.encode
 - module, 7

- pyffstream.ffmpeg
 - module, 1

S

- sencoders (*pyffstream.ffmpeg.FFBin* property), 4

- sencoders (*pyffstream.ffmpeg.FFEncoders* attribute), 4

- set_console_logger() (*in module pyffstream.cli*), 15

- setup_pyffserver_stream() (*in module pyffstream.cli*), 15

- sfilepath (*pyffstream.encode.FileOpts* attribute), 7

- single_quote() (*in module pyffstream.ffmpeg*), 6

- start_stream() (*in module pyffstream.cli*), 15

- StaticEncodeVars (*class in pyffstream.encode*), 7

- status_wait() (*in module pyffstream.cli*), 15

- StatusCode (*class in pyffstream.encode*), 11

- stream_file() (*in module pyffstream.cli*), 15

- StyleFile (*class in pyffstream.encode*), 11

- subtitle (*pyffstream.encode.FileOpts* attribute), 7

V

- VEncoder (*class in pyffstream.encode*), 12

- vencoders (*pyffstream.ffmpeg.FFBin* property), 4

- vencoders (*pyffstream.ffmpeg.FFEncoders* attribute), 5

- version (*pyffstream.ffmpeg.FFBin* property), 4

- versions (*pyffstream.ffmpeg.FFBanner* attribute), 1

- vf_join() (*pyffstream.ffmpeg.Filter* static method), 5

W

- win_set_local_ffmpeg() (*in module pyffstream.cli*), 15