
ffmpeg-cli-wrapper Documentation

Release 0.6

Andrew Brampton

November 13, 2016

1	net.bramp.commons.lang3.math.gson	3
1.1	FractionAdapter	3
1.1.1	Constructors	3
1.1.2	Methods	3
 2	 net.bramp.ffmpeg	 5
2.1	FFcommon	5
2.1.1	Fields	5
2.1.2	Constructors	5
2.1.3	Methods	6
2.2	FFmpeg	6
2.2.1	Fields	6
2.2.2	Constructors	9
2.2.3	Methods	10
2.3	FFmpegExecutor	11
2.3.1	Fields	11
2.3.2	Constructors	11
2.3.3	Methods	11
2.4	FFmpegUtils	12
2.4.1	Fields	12
2.4.2	Constructors	12
2.4.3	Methods	12
2.5	FFprobe	13
2.5.1	Fields	13
2.5.2	Constructors	13
2.5.3	Methods	13
2.6	ProcessFunction	14
2.6.1	Methods	14
2.7	RunProcessFunction	14
2.7.1	Fields	14
2.7.2	Methods	14
 3	 net.bramp.ffmpeg.builder	 15
3.1	FFmpegBuilder	15
3.1.1	Fields	15
3.1.2	Methods	16
3.2	FFmpegBuilder.Strict	19
3.2.1	Enum Constants	19

3.3	FFmpegBuilder.Verboseity	20
3.3.1	Enum Constants	20
3.4	FFmpegOutputBuilder	20
3.4.1	Fields	21
3.4.2	Constructors	24
3.4.3	Methods	24
3.5	MetadataSpecifier	32
3.5.1	Fields	32
3.5.2	Methods	32
3.6	StreamSpecifier	33
3.6.1	Fields	33
3.6.2	Methods	33
3.7	StreamSpecifierType	35
3.7.1	Enum Constants	35
4	net.bramp.ffmpeg.gson	37
4.1	LowercaseEnumTypeAdapterFactory	37
4.1.1	Methods	37
4.2	NamedBitsetAdapter	37
4.2.1	Fields	37
4.2.2	Constructors	38
4.2.3	Methods	38
5	net.bramp.ffmpeg.info	39
5.1	Codec	39
5.1.1	Fields	39
5.1.2	Constructors	40
5.1.3	Methods	40
5.2	Codec.Type	40
5.2.1	Enum Constants	41
5.3	Format	41
5.3.1	Fields	41
5.3.2	Constructors	41
5.3.3	Methods	42
6	net.bramp.ffmpeg.io	43
6.1	CRC32InputStream	43
6.1.1	Fields	43
6.1.2	Constructors	43
6.1.3	Methods	43
6.2	LoggingFilterReader	44
6.2.1	Fields	44
6.2.2	Constructors	45
6.2.3	Methods	45
6.3	ProcessUtils	45
6.3.1	Methods	45
7	net.bramp.ffmpeg.job	47
7.1	FFmpegJob	47
7.1.1	Fields	47
7.1.2	Constructors	47
7.1.3	Methods	48
7.2	FFmpegJob.State	48
7.2.1	Enum Constants	48
7.3	SinglePassFFmpegJob	48

7.3.1	Fields	48
7.3.2	Constructors	48
7.3.3	Methods	49
7.4	TwoPassFFmpegJob	49
7.4.1	Fields	49
7.4.2	Constructors	49
7.4.3	Methods	49
8	net.bramp.ffmpeg.modelmapper	51
8.1	Mapper	51
8.1.1	Methods	51
8.2	Mapper.AudioWrapper	51
8.2.1	Fields	52
8.2.2	Constructors	52
8.3	Mapper.VideoWrapper	52
8.3.1	Fields	52
8.3.2	Constructors	52
8.4	NotDefaultCondition	52
8.4.1	Fields	52
8.4.2	Methods	53
9	net.bramp.ffmpeg.nut	55
9.1	Frame	55
9.1.1	Fields	55
9.1.2	Methods	57
9.2	FrameCode	57
9.2.1	Fields	57
9.2.2	Methods	58
9.3	IndexPacket	58
9.4	InfoPacket	58
9.5	MainHeaderPacket	58
9.5.1	Fields	58
9.5.2	Constructors	59
9.5.3	Methods	59
9.6	NutDataInputStream	59
9.6.1	Fields	60
9.6.2	Constructors	60
9.6.3	Methods	60
9.7	NutReader	62
9.7.1	Fields	62
9.7.2	Constructors	63
9.7.3	Methods	63
9.8	NutReaderListener	64
9.8.1	Methods	64
9.9	Packet	64
9.9.1	Fields	64
9.9.2	Methods	65
9.10	Packet.Startcode	65
9.10.1	Enum Constants	65
9.11	PacketFooter	66
9.11.1	Fields	66
9.11.2	Methods	66
9.12	PacketHeader	66
9.12.1	Fields	66

9.12.2	Methods	67
9.13	RawHandler	67
9.13.1	Methods	67
9.14	Stream	67
9.14.1	Fields	67
9.14.2	Constructors	68
9.15	StreamHeaderPacket	68
9.15.1	Fields	68
9.15.2	Methods	70
9.16	SyncPointPacket	70
10	net.bramp.ffmpeg.options	71
10.1	AudioEncodingOptions	71
10.1.1	Fields	71
10.1.2	Constructors	72
10.2	EncodingOptions	72
10.2.1	Fields	72
10.2.2	Constructors	72
10.2.3	Methods	72
10.3	MainEncodingOptions	73
10.3.1	Fields	73
10.3.2	Constructors	73
10.4	VideoEncodingOptions	73
10.4.1	Fields	73
10.4.2	Constructors	74
11	net.bramp.ffmpeg.probe	75
11.1	FFmpegDisposition	75
11.1.1	Fields	75
11.2	FFmpegError	76
11.2.1	Fields	76
11.3	FFmpegFormat	77
11.3.1	Fields	77
11.4	FFmpegProbeResult	78
11.4.1	Fields	78
11.4.2	Methods	78
11.5	FFmpegStream	79
11.5.1	Fields	79
11.6	FFmpegStream.CodecType	82
11.6.1	Enum Constants	82
12	net.bramp.ffmpeg.progress	83
12.1	AbstractSocketProgressParser	83
12.1.1	Fields	83
12.1.2	Constructors	83
12.1.3	Methods	83
12.2	Progress	84
12.2.1	Fields	84
12.2.2	Constructors	85
12.2.3	Methods	85
12.3	ProgressListener	86
12.3.1	Methods	86
12.4	ProgressParser	86
12.4.1	Methods	86

12.5	StreamProgressParser	87
12.5.1	Fields	87
12.5.2	Constructors	87
12.5.3	Methods	87
12.6	TcpProgressParser	87
12.6.1	Fields	87
12.6.2	Constructors	88
12.6.3	Methods	88
12.7	TcpProgressParserRunnable	88
12.7.1	Fields	88
12.7.2	Constructors	89
12.7.3	Methods	89
12.8	UdpProgressParser	89
12.8.1	Fields	89
12.8.2	Constructors	89
12.8.3	Methods	89
12.9	UdpProgressParserRunnable	90
12.9.1	Fields	90
12.9.2	Constructors	90
12.9.3	Methods	90

13 Indices and tables

91

Contents:

net.bramp.commons.lang3.math.gson

1.1 FractionAdapter

public class **FractionAdapter** extends TypeAdapter<Fraction>
GSON TypeAdapter for Apache Commons Math Fraction Object

Author bramp

1.1.1 Constructors

FractionAdapter

public **FractionAdapter** ()

1.1.2 Methods

read

public Fraction **read** (JsonReader *reader*)

write

public void **write** (JsonWriter *writer*, Fraction *value*)

2.1 FFcommon

abstract class **FFcommon**

Private class to contain common methods for both FFmpeg and FFprobe.

2.1.1 Fields

path

final *String* **path**

Path to the binary (e.g. /usr/bin/ffmpeg)

runFunc

final *ProcessFunction* **runFunc**

Function to run FFmpeg. We define it like this so we can swap it out (during testing)

version

String **version**

Version string

2.1.2 Constructors

FFcommon

public **FFcommon** (*String path*)

FFcommon

protected **FFcommon** (*String path, ProcessFunction runFunction*)

2.1.3 Methods

getPath

```
public String getPath ()
```

path

```
public List<String> path (List<String> args)
```

run

```
public void run (List<String> args)
```

Runs ffmpeg with the supplied args. Blocking until finished.

Parameters

- **args** – The arguments to pass to the binary.

Throws

- **IOException** – If there is a problem executing the binary.

throwOnError

```
protected void throwOnError (Process p)
```

version

```
public synchronized String version ()
```

wrapInReader

```
protected BufferedReader wrapInReader (Process p)
```

2.2 FFmpeg

```
public class FFmpeg extends FFcommon
```

Wrapper around FFmpeg

Author bramp

2.2.1 Fields

AUDIO_DEPTH_DBL

```
public static final String AUDIO_DEPTH_DBL
```

AUDIO_DEPTH_FLT

public static final String **AUDIO_DEPTH_FLT**

AUDIO_DEPTH_S16

public static final String **AUDIO_DEPTH_S16**

AUDIO_DEPTH_S32

public static final String **AUDIO_DEPTH_S32**

AUDIO_DEPTH_U8

public static final String **AUDIO_DEPTH_U8**

AUDIO_FORMAT_DBL

public static final String **AUDIO_FORMAT_DBL**

AUDIO_FORMAT_FLT

public static final String **AUDIO_FORMAT_FLT**

AUDIO_FORMAT_S16

public static final String **AUDIO_FORMAT_S16**

AUDIO_FORMAT_S32

public static final String **AUDIO_FORMAT_S32**

AUDIO_FORMAT_U8

public static final String **AUDIO_FORMAT_U8**

AUDIO_MONO

public static final int **AUDIO_MONO**

AUDIO_SAMPLE_11025

public static final int **AUDIO_SAMPLE_11025**

AUDIO_SAMPLE_12000

public static final int **AUDIO_SAMPLE_12000**

AUDIO_SAMPLE_16000

public static final int **AUDIO_SAMPLE_16000**

AUDIO_SAMPLE_22050

public static final int **AUDIO_SAMPLE_22050**

AUDIO_SAMPLE_32000

public static final int **AUDIO_SAMPLE_32000**

AUDIO_SAMPLE_44100

public static final int **AUDIO_SAMPLE_44100**

AUDIO_SAMPLE_48000

public static final int **AUDIO_SAMPLE_48000**

AUDIO_SAMPLE_8000

public static final int **AUDIO_SAMPLE_8000**

AUDIO_SAMPLE_96000

public static final int **AUDIO_SAMPLE_96000**

AUDIO_STEREO

public static final int **AUDIO_STEREO**

CODECS_REGEX

static final [Pattern](#) **CODECS_REGEX**

DEFAULT_PATH

public static final [String](#) **DEFAULT_PATH**

FFMPEG

public static final String **FFMPEG**

FORMATS_REGEX

static final Pattern **FORMATS_REGEX**

FPS_23_976

public static final Fraction **FPS_23_976**

FPS_24

public static final Fraction **FPS_24**

FPS_29_97

public static final Fraction **FPS_29_97**

FPS_30

public static final Fraction **FPS_30**

codecs

List<Codec> **codecs**
Supported codecs

formats

List<Format> **formats**
Supported formats

2.2.2 Constructors

FFmpeg

public **FFmpeg** ()

FFmpeg

public **FFmpeg** (*ProcessFunction runFunction*)

FFmpeg

public **FFmpeg** (*String path*)

FFmpeg

public **FFmpeg** (*String path, ProcessFunction runFunction*)

2.2.3 Methods

builder

public *FFmpegBuilder* **builder** ()

codecs

public synchronized *List<Codec>* **codecs** ()

createProgressParser

protected *ProgressParser* **createProgressParser** (*ProgressListener listener*)

formats

public synchronized *List<Format>* **formats** ()

getPath

public *String* **getPath** ()

isFFmpeg

public boolean **isFFmpeg** ()

Returns true if the binary we are using is the true ffmpeg. This is to avoid conflict with avconv (from the libav project), that some symlink to ffmpeg.

Throws

- **IOException** – If a I/O error occurs while executing ffmpeg.

Returns true iff this is the official ffmpeg binary.

run

public void **run** (*List<String> args*)

run

public void **run** (*FFmpegBuilder builder, ProgressListener listener*)

2.3 FFmpegExecutor

```
public class FFmpegExecutor
```

2.3.1 Fields

ffmpeg

```
final FFmpeg fffmpeg
```

ffprobe

```
final FFprobe ffprobe
```

2.3.2 Constructors

FFmpegExecutor

```
public FFmpegExecutor ()
```

FFmpegExecutor

```
public FFmpegExecutor (FFmpeg ffmpeg)
```

FFmpegExecutor

```
public FFmpegExecutor (FFmpeg ffmpeg, FFprobe ffprobe)
```

2.3.3 Methods

createJob

```
public FFmpegJob createJob (FFmpegBuilder builder)
```

createJob

```
public FFmpegJob createJob (FFmpegBuilder builder, ProgressListener listener)
```

createTwoPassJob

```
public FFmpegJob createTwoPassJob (FFmpegBuilder builder)
```

Creates a two pass job, which will execute FFmpeg twice to produce a better quality output. More info: <https://trac.ffmpeg.org/wiki/x264EncodingGuide#twopass>

Parameters

- **builder** – The FFmpegBuilder

Returns A new two-pass FFmpegJob

2.4 FFmpegUtils

public final class **FFmpegUtils**
Helper class with commonly used methods

2.4.1 Fields

BITRATE_REGEX

static final `Pattern` **BITRATE_REGEX**

gson

static final `Gson` **gson**

2.4.2 Constructors

FFmpegUtils

FFmpegUtils ()

2.4.3 Methods

getGson

static `Gson` **getGson** ()

millisecondsToString

public static `String` **millisecondsToString** (long *milliseconds*)
Convert milliseconds to “hh:mm:ss.ms” String representation.

Parameters

- **milliseconds** – time duration in milliseconds

Returns time duration in human-readable format

parseBitrate

public static long **parseBitrate** (`String` *bitrate*)
Converts a string representation of bitrate to a long of bits per second

Parameters

- **bitrate** – in the form of 12.3kbits/s

Returns the bitrate in bits per second.

2.5 FFprobe

public class **FFprobe** extends *FFcommon*
Wrapper around FFprobe

Author bramp

2.5.1 Fields

DEFAULT_PATH

static final *String* **DEFAULT_PATH**

FFPROBE

static final *String* **FFPROBE**

LOG

static final *Logger* **LOG**

gson

static final *Gson* **gson**

2.5.2 Constructors

FFprobe

public **FFprobe** ()

FFprobe

public **FFprobe** (*ProcessFunction* runFunction)

FFprobe

public **FFprobe** (*String* path, *ProcessFunction* runFunction)

2.5.3 Methods

isFFprobe

public boolean **isFFprobe** ()

Returns true if the binary we are using is the true ffprobe. This is to avoid conflict with avprobe (from the libav project), that some symlink to ffprobe.

Throws

- **IOException** – If a I/O error occurs while executing ffprobe.

Returns true iff this is the official ffprobe binary.

probe

```
public FFmpegProbeResult probe (String mediaPath)
```

probe

```
public FFmpegProbeResult probe (String mediaPath, String userAgent)
```

run

```
public void run (List<String> args)
```

2.6 ProcessFunction

```
public interface ProcessFunction
```

Runs a process returning a Reader to its stdout

Author bramp

2.6.1 Methods

run

```
Process run (List<String> args)
```

2.7 RunProcessFunction

```
public class RunProcessFunction implements ProcessFunction
```

Simple function that creates a Process with the arguments, and returns a BufferedReader reading stdout

Author bramp

2.7.1 Fields

LOG

```
static final Logger LOG
```

2.7.2 Methods

run

```
public Process run (List<String> args)
```

net.bramp.ffmpeg.builder

3.1 FFmpegBuilder

```
public class FFmpegBuilder  
    Builds a ffmpeg command line  
    Author bramp
```

3.1.1 Fields

LOG

```
static final Logger LOG
```

extra_args

```
final List<String> extra_args
```

format

```
String format
```

inputProbes

```
final Map<String, FFmpegProbeResult> inputProbes
```

inputs

```
final List<String> inputs
```

outputs

```
final List<FFmpegOutputBuilder> outputs
```

override

boolean **override**

pass

int **pass**

pass_directory

String **pass_directory**

pass_prefix

String **pass_prefix**

progress

URI **progress**

read_at_native_frame_rate

boolean **read_at_native_frame_rate**

startOffset

Long **startOffset**

user_agent

String **user_agent**

verbosity

Verbosity **verbosity**

3.1.2 Methods

addExtraArgs

public *FFmpegBuilder* **addExtraArgs** (String... *values*)

Add additional output arguments (for flags which aren't currently supported).

Parameters

- **values** – The extra arguments.

Returns this

addInput

```
public FFmpegBuilder addInput (FFmpegProbeResult result)
```

addInput

```
public FFmpegBuilder addInput (String filename)
```

addOutput

```
public FFmpegOutputBuilder addOutput (String filename)
    Adds new output file.
```

Parameters

- **filename** – output file path

Returns A new *FFmpegOutputBuilder*

addOutput

```
public FFmpegOutputBuilder addOutput (URI uri)
    Adds new output file.
```

Parameters

- **uri** – output file uri typically a stream

Returns A new *FFmpegOutputBuilder*

addOutput

```
public FFmpegBuilder addOutput (FFmpegOutputBuilder output)
    Adds an existing FFmpegOutputBuilder. This is similar to calling the other addOutput methods but instead allows an existing FFmpegOutputBuilder to be used, and reused.
```

```
List<String> args = new FFmpegBuilder()
    .addOutput(new FFmpegOutputBuilder()
        .setFilename("output.flv")
        .setVideoCodec("flv")
    )
    .build();
```

Parameters

- **output** – FFmpegOutputBuilder to add

Returns this

addProgress

```
public FFmpegBuilder addProgress (URI uri)
```

addStdoutOutput

public *FFmpegOutputBuilder* **addStdoutOutput** ()
Create new output (to stdout)

Returns A new *FFmpegOutputBuilder*

build

public List<String> **build** ()

clearInputs

protected void **clearInputs** ()

getOverrideOutputFiles

public boolean **getOverrideOutputFiles** ()

overrideOutputFiles

public *FFmpegBuilder* **overrideOutputFiles** (boolean *override*)

readAtNativeFrameRate

public *FFmpegBuilder* **readAtNativeFrameRate** ()

setFormat

public *FFmpegBuilder* **setFormat** (String *format*)

setInput

public *FFmpegBuilder* **setInput** (*FFmpegProbeResult* *result*)

setInput

public *FFmpegBuilder* **setInput** (String *filename*)

setPass

public *FFmpegBuilder* **setPass** (int *pass*)

setPassDirectory

public *FFmpegBuilder* **setPassDirectory** (String *directory*)

setPassPrefix

```
public FFmpegBuilder setPassPrefix (String prefix)
```

setStartOffset

```
public FFmpegBuilder setStartOffset (long duration, TimeUnit units)
```

setUserAgent

```
public FFmpegBuilder setUserAgent (String userAgent)
```

setVerbosity

```
public FFmpegBuilder setVerbosity (Verbosity verbosity)
```

3.2 FFmpegBuilder.Strict

```
public enum Strict
```

3.2.1 Enum Constants

EXPERIMENTAL

```
public static final FFmpegBuilder.Strict EXPERIMENTAL
```

NORMAL

```
public static final FFmpegBuilder.Strict NORMAL
```

STRICT

```
public static final FFmpegBuilder.Strict STRICT
```

UNOFFICAL

```
public static final FFmpegBuilder.Strict UNOFFICAL
```

VERY

```
public static final FFmpegBuilder.Strict VERY
```

3.3 FFmpegBuilder.Verbose

public enum **Verbose**

Log level options: <https://ffmpeg.org/ffmpeg.html#Generic-options>

3.3.1 Enum Constants

DEBUG

public static final *FFmpegBuilder.Verbose* **DEBUG**

ERROR

public static final *FFmpegBuilder.Verbose* **ERROR**

FATAL

public static final *FFmpegBuilder.Verbose* **FATAL**

INFO

public static final *FFmpegBuilder.Verbose* **INFO**

PANIC

public static final *FFmpegBuilder.Verbose* **PANIC**

QUIET

public static final *FFmpegBuilder.Verbose* **QUIET**

VERBOSE

public static final *FFmpegBuilder.Verbose* **VERBOSE**

WARNING

public static final *FFmpegBuilder.Verbose* **WARNING**

3.4 FFmpegOutputBuilder

public class **FFmpegOutputBuilder**

Builds a representation of a single output/encoding setting

3.4.1 Fields

audio_bit_rate

public long **audio_bit_rate**

audio_bit_stream_filter

public `String` **audio_bit_stream_filter**

audio_channels

public int **audio_channels**

audio_codec

public `String` **audio_codec**

audio_enabled

public boolean **audio_enabled**

audio_quality

public `Integer` **audio_quality**

audio_sample_format

public `String` **audio_sample_format**

audio_sample_rate

public int **audio_sample_rate**

duration

public `Long` **duration**

extra_args

public final `List<String>` **extra_args**

filename

public `String` **filename**

Output filename or uri. Only one may be set

format

public String **format**

meta_tags

public final List<String> **meta_tags**

parent

final *FFmpegBuilder* **parent**

pass_padding_bitrate

public long **pass_padding_bitrate**

startOffset

public Long **startOffset**

strict

public *FFmpegBuilder.Strict* **strict**

subtitle_enabled

public boolean **subtitle_enabled**

targetSize

public long **targetSize**

throwWarnings

public boolean **throwWarnings**

uri

public URI **uri**

video_bit_rate

public long **video_bit_rate**

video_bit_stream_filter

```
public String video_bit_stream_filter
```

video_codec

```
public String video_codec
```

video_copyinkf

```
public boolean video_copyinkf
```

video_enabled

```
public boolean video_enabled
```

video_filter

```
public String video_filter
```

video_filter_complex

```
public String video_filter_complex
```

video_frame_rate

```
public Fraction video_frame_rate
```

video_frames

```
public Integer video_frames
```

video_height

```
public int video_height
```

video_movflags

```
public String video_movflags
```

video_pixel_format

```
public String video_pixel_format
```

video_preset

```
public String video_preset
```

video_quality

```
public Integer video_quality
```

video_size

```
public String video_size
```

video_width

```
public int video_width
```

3.4.2 Constructors

FFmpegOutputBuilder

```
public FFmpegOutputBuilder ()
```

FFmpegOutputBuilder

```
protected FFmpegOutputBuilder (FFmpegBuilder parent, String filename)
```

FFmpegOutputBuilder

```
protected FFmpegOutputBuilder (FFmpegBuilder parent, URI uri)
```

3.4.3 Methods

addExtraArgs

```
public FFmpegOutputBuilder addExtraArgs (String... values)
```

Add additional output arguments (for flags which aren't currently supported).

Parameters

- **values** – The extra arguments

Returns this

addMetaTag

public *FFmpegOutputBuilder* **addMetaTag** (*String key, String value*)

Add metadata on output streams. Which keys are possible depends on the used codec.

Parameters

- **key** – Metadata key, e.g. “comment”
- **value** – Value to set for key

Returns this

addMetaTag

public *FFmpegOutputBuilder* **addMetaTag** (*MetadataSpecifier spec, String key, String value*)

Add metadata on output streams. Which keys are possible depends on the used codec.

```
import static net.bramp.ffmpeg.builder.MetadataSpecifier.*;
import static net.bramp.ffmpeg.builder.StreamSpecifier.*;
import static net.bramp.ffmpeg.builder.StreamSpecifierType.*;

new FFmpegBuilder()
    .addMetaTag("title", "Movie Title") // Annotate whole file
    .addMetaTag(chapter(0), "author", "Bob") // Annotate first chapter
    .addMetaTag(program(0), "comment", "Awesome") // Annotate first program
    .addMetaTag(stream(0), "copyright", "Megacorp") // Annotate first stream
    .addMetaTag(stream(Video), "framerate", "24fps") // Annotate all video streams
    .addMetaTag(stream(Video, 0), "artist", "Joe") // Annotate first video stream
    .addMetaTag(stream(Audio, 0), "language", "eng") // Annotate first audio stream
    .addMetaTag(stream(Subtitle, 0), "language", "fre") // Annotate first subtitle stream
    .addMetaTag(usable(), "year", "2010") // Annotate all streams with a usable configuration
```

```
assertThat(global().spec(), is("g"));
assertThat(chapter(1).spec(), is("c:1"));
assertThat(program(1).spec(), is("p:1"));
assertThat(stream(1).spec(), is("s:1"));
assertThat(stream(id(1)).spec(), is("s:i:1"));
```

Parameters

- **spec** – Metadata specifier, e.g *MetadataSpec.stream(Audio, 0)*
- **key** – Metadata key, e.g. “comment”
- **value** – Value to set for key

Returns this

build

protected *List<String>* **build** (*int pass*)

build

protected *List<String>* **build** (*FFmpegBuilder parent, int pass*)

Builds the arguments

Parameters

- **parent** – The parent *FFmpegBuilder*

- **pass** – The particular pass. For one-pass this value will be zero, for multi-pass, it will be 1 for the first pass, 2 for the second, and so on.

Returns The arguments

buildOptions

public *EncodingOptions* **buildOptions** ()

Returns a representation of this Builder that can be safely serialised. NOTE: This method is horribly out of date, and its use should be rethought.

Returns A new EncodingOptions capturing this Builder's state

checkValidStream

public static *URI* **checkValidStream** (*URI uri*)

Checks if the URI is valid for streaming to

Parameters

- **uri** – The URI to check

Throws

- *IllegalArgumentException* – if the URI is not valid.

Returns The passed in URI if it is valid

disableAudio

public *FFmpegOutputBuilder* **disableAudio** ()

disableSubtitle

public *FFmpegOutputBuilder* **disableSubtitle** ()

disableVideo

public *FFmpegOutputBuilder* **disableVideo** ()

done

public *FFmpegBuilder* **done** ()

Finished with this output

Returns the parent FFmpegBuilder

getFilename

public *String* **getFilename** ()

getUri

public URI **getUri** ()

isValidSize

protected static boolean **isValidSize** (int *widthOrHeight*)

setAudioBitDepth

public *FFmpegOutputBuilder* **setAudioBitDepth** (String *bit_depth*)
Sets the audio bit depth.

Parameters

- **bit_depth** – The sample format, one of the `net.bramp.ffmpeg.FFmpeg#AUDIO_DEPTH_*` constants.

Returns this

See also: `net.bramp.ffmpeg.FFmpeg.AUDIO_DEPTH_U8`, `net.bramp.ffmpeg.FFmpeg.AUDIO_DEPTH_S16`, `net.bramp.ffmpeg.FFmpeg.AUDIO_DEPTH_S32`, `net.bramp.ffmpeg.FFmpeg.AUDIO_DEPTH_FLT`, `net.bramp.ffmpeg.FFmpeg.AUDIO_DEPTH_DBL`

setAudioBitRate

public *FFmpegOutputBuilder* **setAudioBitRate** (long *bit_rate*)
Sets the Audio bit rate

Parameters

- **bit_rate** – Audio bitrate in bits per second.

Returns this

setAudioBitStreamFilter

public *FFmpegOutputBuilder* **setAudioBitStreamFilter** (String *filter*)

setAudioChannels

public *FFmpegOutputBuilder* **setAudioChannels** (int *channels*)
Sets the number of audio channels

Parameters

- **channels** – Number of channels

Returns this

See also: `net.bramp.ffmpeg.FFmpeg.AUDIO_MONO`, `net.bramp.ffmpeg.FFmpeg.AUDIO_STEREO`

setAudioCodec

public *FFmpegOutputBuilder* **setAudioCodec** (String *codec*)

setAudioQuality

public *FFmpegOutputBuilder* **setAudioQuality** (int *quality*)

setAudioSampleFormat

public *FFmpegOutputBuilder* **setAudioSampleFormat** (String *sample_format*)
Sets the audio sample format.

Parameters

- **sample_format** – The sample format, one of the `net.bramp.ffmpeg.FFmpeg#AUDIO_FORMAT_*` constants.

Returns

See also: `net.bramp.ffmpeg.FFmpeg.AUDIO_FORMAT_U8`, `net.bramp.ffmpeg.FFmpeg.AUDIO_FORMAT_S16`, `net.bramp.ffmpeg.FFmpeg.AUDIO_FORMAT_S32`, `net.bramp.ffmpeg.FFmpeg.AUDIO_FORMAT_FLT`, `net.bramp.ffmpeg.FFmpeg.AUDIO_FORMAT_DBL`

setAudioSampleRate

public *FFmpegOutputBuilder* **setAudioSampleRate** (int *sample_rate*)
Sets the Audio sample rate, for example 44_000.

Parameters

- **sample_rate** – Samples measured in Hz

Returns

See also: `net.bramp.ffmpeg.FFmpeg.AUDIO_SAMPLE_8000`, `net.bramp.ffmpeg.FFmpeg.AUDIO_SAMPLE_12000`, `net.bramp.ffmpeg.FFmpeg.AUDIO_SAMPLE_16000`, `net.bramp.ffmpeg.FFmpeg.AUDIO_SAMPLE_22050`, `net.bramp.ffmpeg.FFmpeg.AUDIO_SAMPLE_32000`, `net.bramp.ffmpeg.FFmpeg.AUDIO_SAMPLE_44100`, `net.bramp.ffmpeg.FFmpeg.AUDIO_SAMPLE_48000`, `net.bramp.ffmpeg.FFmpeg.AUDIO_SAMPLE_96000`

setComplexVideoFilter

public *FFmpegOutputBuilder* **setComplexVideoFilter** (String *filter*)

setDuration

public *FFmpegOutputBuilder* **setDuration** (long *duration*, *TimeUnit units*)
Stop writing the output after duration is reached.

Parameters

- **duration** – The duration
- **units** – The units the duration is in

Returns

setFilename

```
public FFmpegOutputBuilder setFilename (String filename)
```

setFormat

```
public FFmpegOutputBuilder setFormat (String format)
```

setFrames

```
public FFmpegOutputBuilder setFrames (int frames)  
    Set the number of video frames to record.
```

Parameters

- **frames** – The number of frames

Returns this

setPassPaddingBitrate

```
public FFmpegOutputBuilder setPassPaddingBitrate (long bitrate)  
    When doing multi-pass we add a little extra padding, to ensure we reach our target
```

Parameters

- **bitrate** – bit rate

Returns this

setStartOffset

```
public FFmpegOutputBuilder setStartOffset (long offset, TimeUnit units)  
    Decodes but discards input until the offset.
```

Parameters

- **offset** – The offset
- **units** – The units the offset is in

Returns this

setStrict

```
public FFmpegOutputBuilder setStrict (FFmpegBuilder.Strict strict)
```

setTargetSize

```
public FFmpegOutputBuilder setTargetSize (long targetSize)  
    Target output file size (in bytes)
```

Parameters

- **targetSize** – The target size in bytes

Returns this

setUri

```
public FFmpegOutputBuilder setUri (URI uri)
```

setVideoBitRate

```
public FFmpegOutputBuilder setVideoBitRate (long bit_rate)
```

setVideoBitStreamFilter

```
public FFmpegOutputBuilder setVideoBitStreamFilter (String filter)
```

setVideoCodec

```
public FFmpegOutputBuilder setVideoCodec (String codec)
```

setVideoCopyInkf

```
public FFmpegOutputBuilder setVideoCopyInkf (boolean copyinkf)
```

setVideoFilter

```
public FFmpegOutputBuilder setVideoFilter (String filter)
```

Sets Video Filter TODO Build a fluent Filter builder

Parameters

- **filter** – The video filter.

Returns this

setVideoFrameRate

```
public FFmpegOutputBuilder setVideoFrameRate (Fraction frame_rate)
```

Sets the video's frame rate

Parameters

- **frame_rate** – Frames per second

Returns this

See also: `net.bramp.ffmpeg.FFmpeg.FPS_30`, `net.bramp.ffmpeg.FFmpeg.FPS_29_97`,
`net.bramp.ffmpeg.FFmpeg.FPS_24`, `net.bramp.ffmpeg.FFmpeg.FPS_23_976`

setVideoFrameRate

```
public FFmpegOutputBuilder setVideoFrameRate (int frames, int per)
```

Set the video frame rate in terms of frames per interval. For example 24fps would be 24/1, however NTSC TV at 23.976fps would be 24000 per 1001.

Parameters

- **frames** – The number of frames within the given seconds

- **per** – The number of seconds

Returns this

setVideoFrameRate

public *FFmpegOutputBuilder* **setVideoFrameRate** (double *frame_rate*)

setVideoHeight

public *FFmpegOutputBuilder* **setVideoHeight** (int *height*)

setVideoMovFlags

public *FFmpegOutputBuilder* **setVideoMovFlags** (*String* *movflags*)

setVideoPixelFormat

public *FFmpegOutputBuilder* **setVideoPixelFormat** (*String* *format*)

setVideoPreset

public *FFmpegOutputBuilder* **setVideoPreset** (*String* *preset*)

setVideoQuality

public *FFmpegOutputBuilder* **setVideoQuality** (int *quality*)

setVideoResolution

public *FFmpegOutputBuilder* **setVideoResolution** (int *width*, int *height*)

setVideoResolution

public *FFmpegOutputBuilder* **setVideoResolution** (*String* *abbreviation*)

Sets video resolution based on an abbreviation, e.g. “ntsc” for 720x480, or “vga” for 640x480

Parameters

- **abbreviation** – The abbreviation size. No validation is done, instead the value is passed as is to ffmpeg.

Returns this

See also: [ffmpeg video size](#)

setVideoWidth

public *FFmpegOutputBuilder* **setVideoWidth** (int *width*)

useOptions

```
public FFmpegOutputBuilder useOptions (EncodingOptions opts)
```

useOptions

```
public FFmpegOutputBuilder useOptions (MainEncodingOptions opts)
```

useOptions

```
public FFmpegOutputBuilder useOptions (AudioEncodingOptions opts)
```

useOptions

```
public FFmpegOutputBuilder useOptions (VideoEncodingOptions opts)
```

3.5 MetadataSpecifier

```
public class MetadataSpecifier
```

```
    Metadata spec, as described in the “map_metadata” section of https://www.ffmpeg.org/ffmpeg-all.html#Main-options
```

3.5.1 Fields

spec

```
final String spec
```

3.5.2 Methods

chapter

```
public static MetadataSpecifier chapter (int index)
```

checkValidKey

```
public static String checkValidKey (String key)
```

global

```
public static MetadataSpecifier global ()
```

program

```
public static MetadataSpecifier program (int index)
```


spec

```
public String spec ()
```

stream

```
public static MetadataSpecifier stream (int index)
```

stream

```
public static MetadataSpecifier stream (StreamSpecifierType type)
```

stream

```
public static MetadataSpecifier stream (StreamSpecifierType stream_type, int stream_index)
```

stream

```
public static MetadataSpecifier stream (StreamSpecifier spec)
```

3.6 StreamSpecifier

```
public class StreamSpecifier  
    https://ffmpeg.org/ffmpeg.html#Stream-specifiers
```

3.6.1 Fields

spec

```
final String spec
```

3.6.2 Methods

id

```
public static StreamSpecifier id (int stream_id)  
    Match the stream by stream id (e.g. PID in MPEG-TS container).
```

Parameters

- **stream_id** – The stream id

Returns A new StreamSpecifier

program

public static *StreamSpecifier* **program** (int *program_id*)
Matches all streams in the program.

Parameters

- **program_id** – The program id

Returns A new StreamSpecifier

program

public static *StreamSpecifier* **program** (int *program_id*, int *stream_index*)
Matches the stream with number *stream_index* in the program with the id *program_id*.

Parameters

- **program_id** – The program id
- **stream_index** – The stream index

Returns A new StreamSpecifier

spec

public String **spec** ()

stream

public static *StreamSpecifier* **stream** (int *index*)
Matches the stream with this index.

Parameters

- **index** – The stream index

Returns A new StreamSpecifier

stream

public static *StreamSpecifier* **stream** (*StreamSpecifierType* *type*)
Matches all streams of this type.

Parameters

- **type** – The stream type

Returns A new StreamSpecifier

stream

public static *StreamSpecifier* **stream** (*StreamSpecifierType* *type*, int *index*)
Matches the stream number *stream_index* of this type.

Parameters

- **type** – The stream type

- **index** – The stream index

Returns A new StreamSpecifier

tag

public static *StreamSpecifier* **tag** (*String key*)

Matches all streams with the given metadata tag.

Parameters

- **key** – The metadata tag

Returns A new StreamSpecifier

tag

public static *StreamSpecifier* **tag** (*String key*, *String value*)

Matches streams with the metadata tag key having the specified value.

Parameters

- **key** – The metadata tag
- **value** – The metadata's value

Returns A new StreamSpecifier

usable

public static *StreamSpecifier* **usable** ()

Matches streams with usable configuration, the codec must be defined and the essential information such as video dimension or audio sample rate must be present.

Returns A new StreamSpecifier

3.7 StreamSpecifierType

public enum **StreamSpecifierType**

3.7.1 Enum Constants

Attachment

public static final *StreamSpecifierType* **Attachment**

Attachment

Audio

public static final *StreamSpecifierType* **Audio**

Audio

Data

public static final *StreamSpecifierType* **Data**
Data

PureVideo

public static final *StreamSpecifierType* **PureVideo**
Video streams which are not attached pictures, video thumbnails or cover arts.

Subtitle

public static final *StreamSpecifierType* **Subtitle**
Subtitles

Video

public static final *StreamSpecifierType* **Video**
Video

net.bramp ffmpeg.gson

4.1 LowercaseEnumTypeAdapterFactory

public class **LowercaseEnumTypeAdapterFactory** implements TypeAdapterFactory
Taken from: `TypeAdapterFactory`

4.1.1 Methods

create

public <T> TypeAdapter<T> **create** (Gson *gson*, TypeToken<T> *type*)

4.2 NamedBitsetAdapter

public class **NamedBitsetAdapter**<T> extends TypeAdapter<T>
Converts a json object which represents a set of booleans. For example:

```
public class Set {  
    public boolean a = true;  
    public boolean b = false;  
    public int c = 1;  
    public int d = 0;  
}
```

is turned into:

```
{  
  "a": true,  
  "b": false,  
  "c": true,  
  "d": false  
}
```

4.2.1 Fields

clazz

final Class<T> **clazz**

4.2.2 Constructors

NamedBitsetAdapter

```
public NamedBitsetAdapter (Class<T> clazz)
```

4.2.3 Methods

read

```
public T read (JsonReader reader)
```

readBoolean

```
protected Optional<Boolean> readBoolean (JsonReader reader)
```

setField

```
protected void setField (T target, String name, boolean value)
```

write

```
public void write (JsonWriter writer, T value)
```

net.bramp.ffmpeg.info

5.1 Codec

```
public class Codec  
    Information about supported Codecs  
    Author bramp
```

5.1.1 Fields

canDecode

```
final boolean canDecode  
    Can I decode with this codec
```

canEncode

```
final boolean canEncode  
    Can I encode with this codec
```

longName

```
final String longName
```

name

```
final String name
```

type

```
final Type type  
    What type of codec is this
```

5.1.2 Constructors

Codec

public **Codec** (*String name*, *String longName*, *String flags*)

5.1.3 Methods

equals

public boolean **equals** (*Object obj*)

getCanDecode

public boolean **getCanDecode** ()

getCanEncode

public boolean **getCanEncode** ()

getLongName

public *String* **getLongName** ()

getName

public *String* **getName** ()

getType

public *Type* **getType** ()

hashCode

public int **hashCode** ()

toString

public *String* **toString** ()

5.2 Codec.Type

enum **Type**

5.2.1 Enum Constants

AUDIO

public static final *Codec.Type* **AUDIO**

SUBTITLE

public static final *Codec.Type* **SUBTITLE**

VIDEO

public static final *Codec.Type* **VIDEO**

5.3 Format

public class **Format**
Information about supported Format
Author bramp

5.3.1 Fields

canDemux

final boolean **canDemux**

canMux

final boolean **canMux**

longName

final *String* **longName**

name

final *String* **name**

5.3.2 Constructors

Format

public **Format** (*String name*, *String longName*, *String flags*)

5.3.3 Methods

equals

public boolean **equals** (*Object obj*)

getCanDemux

public boolean **getCanDemux** ()

getCanMux

public boolean **getCanMux** ()

getLongName

public *String* **getLongName** ()

getName

public *String* **getName** ()

hashCode

public int **hashCode** ()

toString

public *String* **toString** ()

net.bramp.ffmpeg.io

6.1 CRC32InputStream

public class **CRC32InputStream** extends `FilterInputStream`

Calculates the CRC32 for all bytes read through the input stream. Using the `java.util.zip.CRC32` class to calculate the checksum.

6.1.1 Fields

crc

final `CRC32` **crc**

6.1.2 Constructors

CRC32InputStream

public **CRC32InputStream** (`InputStream in`)

6.1.3 Methods

getValue

public long **getValue** ()

mark

public synchronized void **mark** (int *readlimit*)

markSupported

public boolean **markSupported** ()

read

public int **read** ()

read

public int **read** (byte[] *b*)

read

public int **read** (byte[] *b*, int *off*, int *len*)

reset

public synchronized void **reset** ()

resetCrc

public void **resetCrc** ()

skip

public long **skip** (long *n*)

6.2 LoggingFilterReader

public class **LoggingFilterReader** extends [FilterReader](#)
Wraps a Reader, and logs full lines of input as it is read.

Author bramp

6.2.1 Fields

LOG_CHAR

static final char **LOG_CHAR**

buffer

final [StringBuilder](#) **buffer**

logger

final [Logger](#) **logger**

6.2.2 Constructors

LoggingFilterReader

```
public LoggingFilterReader (Reader in, Logger logger)
```

6.2.3 Methods

log

```
protected void log ()
```

read

```
public int read (char[] cbuf, int off, int len)
```

read

```
public int read ()
```

6.3 ProcessUtils

```
public final class ProcessUtils
```

```
    Author bramp
```

6.3.1 Methods

waitForWithTimeout

```
public static int waitForWithTimeout (Process p, long timeout, TimeUnit unit)
```

Waits until a process finishes or a timeout occurs

Parameters

- **p** – process
- **timeout** – timeout in given unit
- **unit** – time unit

Throws

- **TimeoutException** – if a timeout occurs

Returns the process exit value

net.bramp.ffmpeg.job

7.1 FFmpegJob

public abstract class **FFmpegJob** implements `Runnable`

Author bramp

7.1.1 Fields

ffmpeg

final *FFmpeg* **ffmpeg**

listener

final *ProgressListener* **listener**

state

State **state**

7.1.2 Constructors

FFmpegJob

public **FFmpegJob** (*FFmpeg* *ffmpeg*)

FFmpegJob

public **FFmpegJob** (*FFmpeg* *ffmpeg*, *ProgressListener* *listener*)

7.1.3 Methods

getState

```
public State getState ()
```

7.2 FFmpegJob.State

```
public enum State
```

7.2.1 Enum Constants

FAILED

```
public static final FFmpegJob.State FAILED
```

FINISHED

```
public static final FFmpegJob.State FINISHED
```

RUNNING

```
public static final FFmpegJob.State RUNNING
```

WAITING

```
public static final FFmpegJob.State WAITING
```

7.3 SinglePassFFmpegJob

```
public class SinglePassFFmpegJob extends FFmpegJob
```

7.3.1 Fields

builder

```
public final FFmpegBuilder builder
```

7.3.2 Constructors

SinglePassFFmpegJob

```
public SinglePassFFmpegJob (FFmpeg ffmpeg, FFmpegBuilder builder)
```


SinglePassFFmpegJob

```
public SinglePassFFmpegJob (FFmpeg ffmpeg, FFmpegBuilder builder, ProgressListener listener)
```

7.3.3 Methods

run

```
public void run ()
```

7.4 TwoPassFFmpegJob

```
public class TwoPassFFmpegJob extends FFmpegJob
```

7.4.1 Fields

builder

```
final FFmpegBuilder builder
```

passlogPrefix

```
final String passlogPrefix
```

7.4.2 Constructors

TwoPassFFmpegJob

```
public TwoPassFFmpegJob (FFmpeg ffmpeg, FFmpegBuilder builder)
```

TwoPassFFmpegJob

```
public TwoPassFFmpegJob (FFmpeg ffmpeg, FFmpegBuilder builder, ProgressListener listener)
```

7.4.3 Methods

deletePassLog

```
protected void deletePassLog ()
```

run

```
public void run ()
```

net.bramp.ffmpeg.modelmapper

8.1 Mapper

public class **Mapper**
Copies values from one type of object to another
Author bramp

8.1.1 Methods

map

public static void **map** (*MainEncodingOptions* opts, *FFmpegOutputBuilder* dest)

map

public static void **map** (*AudioEncodingOptions* opts, *FFmpegOutputBuilder* dest)

map

public static void **map** (*VideoEncodingOptions* opts, *FFmpegOutputBuilder* dest)

map

public static void **map** (*EncodingOptions* opts, *FFmpegOutputBuilder* dest)

8.2 Mapper.AudioWrapper

static class **AudioWrapper**
Simple wrapper object, to inject the word “audio” in the property name

8.2.1 Fields

audio

public final *AudioEncodingOptions* **audio**

8.2.2 Constructors

AudioWrapper

AudioWrapper (*AudioEncodingOptions* audio)

8.3 Mapper.VideoWrapper

static class **VideoWrapper**

Simple wrapper object, to inject the word “video” in the property name

8.3.1 Fields

video

public final *VideoEncodingOptions* **video**

8.3.2 Constructors

VideoWrapper

VideoWrapper (*VideoEncodingOptions* video)

8.4 NotDefaultCondition

public class **NotDefaultCondition**<S, D> implements Condition<S, D>

Only maps properties which are not their type’s default value.

Author bramp

Parameters

- <S> – source type
- <D> – destination type

8.4.1 Fields

notDefault

public static final *NotDefaultCondition* **notDefault**

8.4.2 Methods

applies

public boolean **applies** (MappingContext<S, D> *context*)

9.1 Frame

```
public class Frame  
    A video or audio frame
```

9.1.1 Fields

FLAG_CHECKSUM

```
static final long FLAG_CHECKSUM
```

FLAG_CODED

```
static final long FLAG_CODED
```

FLAG_CODED_PTS

```
static final long FLAG_CODED_PTS
```

FLAG_EOR

```
static final long FLAG_EOR
```

FLAG_HEADER_IDX

```
static final long FLAG_HEADER_IDX
```

FLAG_INVALID

```
static final long FLAG_INVALID
```

FLAG_KEY

static final long **FLAG_KEY**

FLAG_MATCH_TIME

static final long **FLAG_MATCH_TIME**

FLAG_RESERVED

static final long **FLAG_RESERVED**

FLAG_SIZE_MSB

static final long **FLAG_SIZE_MSB**

FLAG_SM_DATA

static final long **FLAG_SM_DATA**

FLAG_STREAM_ID

static final long **FLAG_STREAM_ID**

data

byte[] **data**

flags

long **flags**

metaData

Map<String, Object> **metaData**

pts

long **pts**

sideData

Map<String, Object> **sideData**

stream

Stream **stream**

9.1.2 Methods

read

public void **read** (*NutReader nut, NutDataInputStream in, int code*)

readMetaData

protected Map<String, Object> **readMetaData** (*NutDataInputStream in*)

toString

public String **toString** ()

9.2 FrameCode

public class **FrameCode**

9.2.1 Fields

dataSizeLsb

int **dataSizeLsb**

dataSizeMul

int **dataSizeMul**

flags

long **flags**

headerIdx

int **headerIdx**

matchTimeDelta

long **matchTimeDelta**

ptsDelta

long **ptsDelta**

reservedCount

int **reservedCount**

streamId

int **streamId**

9.2.2 Methods

toString

public *String* **toString** ()

9.3 IndexPacket

public class **IndexPacket**

9.4 InfoPacket

public class **InfoPacket**

9.5 MainHeaderPacket

class **MainHeaderPacket** extends *Packet*

9.5.1 Fields

BROADCAST_MODE

public static final int **BROADCAST_MODE**

elision

final *List*<*byte*[]> **elision**

flags

long **flags**

frameCodes

```
final List<FrameCode> frameCodes
```

maxDistance

```
long maxDistance
```

minorVersion

```
long minorVersion
```

streamCount

```
int streamCount
```

timeBase

```
Fraction[] timeBase
```

version

```
long version
```

9.5.2 Constructors

MainHeaderPacket

```
public MainHeaderPacket ()
```

9.5.3 Methods

readBody

```
protected void readBody (NutDataInputStream in)
```

toString

```
public String toString ()
```

9.6 NutDataInputStream

```
public class NutDataInputStream implements DataInput  
    A DataInputStream that implements a couple of custom FFmpeg Nut datatypes.
```

9.6.1 Fields

count

final CountingInputStream **count**

crc

final *CRC32InputStream* **crc**

endCrcRange

long **endCrcRange**

in

final DataInputStream **in**

startCrcRange

long **startCrcRange**

9.6.2 Constructors

NutDataInputStream

public **NutDataInputStream** (InputStream *in*)

9.6.3 Methods

getCRC

public long **getCRC** ()

offset

public long **offset** ()

readBoolean

public boolean **readBoolean** ()

readByte

public byte **readByte** ()

readChar

```
public char readChar ()
```

readDouble

```
public double readDouble ()
```

readFloat

```
public float readFloat ()
```

readFully

```
public void readFully (byte[] b)
```

readFully

```
public void readFully (byte[] b, int off, int len)
```

readInt

```
public int readInt ()
```

readLine

```
public String readLine ()
```

readLong

```
public long readLong ()
```

readShort

```
public short readShort ()
```

readSignedVarInt

```
public long readSignedVarInt ()
```

readStartCode

```
public long readStartCode ()
```

readUTF

```
public String readUTF ()
```

readUnsignedByte

```
public int readUnsignedByte ()
```

readUnsignedShort

```
public int readUnsignedShort ()
```

readVarArray

```
public byte[] readVarArray ()
```

readVarInt

```
public int readVarInt ()
```

readVarLong

```
public long readVarLong ()
```

resetCRC

```
public void resetCRC ()
```

skipBytes

```
public int skipBytes (int n)
```

9.7 NutReader

```
public class NutReader
```

Demuxer for the FFmpeg Nut file format. Lots of things not implemented, startcode searching, crc checks, etc

See also: <https://www.ffmpeg.org/~michael/nut.txt>,
<https://github.com/FFmpeg/FFmpeg/blob/master/libavformat/nutdec.c>

9.7.1 Fields

HEADER

```
static final byte[] HEADER
```

header

```
public MainHeaderPacket header
```

in

```
final NutDataInputStream in
```

listener

```
final NutReaderListener listener
```

streams

```
public final List<Stream> streams
```

9.7.2 Constructors

NutReader

```
public NutReader (InputStream in, NutReaderListener listener)
```

9.7.3 Methods

isKnownStartcode

```
public static boolean isKnownStartcode (long startcode)
```

read

```
public void read ()
```

Demux the inputstream

Throws

- **IOException** – If a I/O error occurs

readFileId

```
protected void readFileId ()
```

Read the magic at the beginning of the file.

Throws

- **IOException** – If a I/O error occurs

readReservedHeaders

protected long **readReservedHeaders** ()

Read headers we don't know how to parse yet, returning the next startcode.

Throws

- **IOException** – If a I/O error occurs

Returns The next startcode

9.8 NutReaderListener

public interface **NutReaderListener**

9.8.1 Methods

frame

void **frame** (*Frame frame*)

Executes when a new frame is found.

Parameters

- **frame** – A single Frame

stream

void **stream** (*Stream stream*)

Executes when a new stream is found.

Parameters

- **stream** – The stream

9.9 Packet

public class **Packet**

9.9.1 Fields

LOG

static final Logger **LOG**

footer

public final *PacketFooter* **footer**

header

public final *PacketHeader* **header**

9.9.2 Methods**read**

public void **read** (*NutDataInputStream in*, long *startcode*)

readBody

protected void **readBody** (*NutDataInputStream in*)

seekToPacketFooter

public void **seekToPacketFooter** (*NutDataInputStream in*)

toString

public String **toString** ()

9.10 Packet.Startcode

public enum **Startcode**

9.10.1 Enum Constants**INDEX**

public static final *Packet.Startcode* **INDEX**

INFO

public static final *Packet.Startcode* **INFO**

MAIN

public static final *Packet.Startcode* **MAIN**

STREAM

public static final *Packet.Startcode* **STREAM**

SYNCPOINT

public static final *Packet.Startcode* **SYNCPOINT**

9.11 PacketFooter

class **PacketFooter**

9.11.1 Fields

checksum

int **checksum**

9.11.2 Methods

read

public void **read** (*NutDataInputStream in*)

toString

public *String* **toString** ()

9.12 PacketHeader

class **PacketHeader**

9.12.1 Fields

checksum

int **checksum**

end

long **end**

forwardPtr

long **forwardPtr**

startcode

long **startcode**

9.12.2 Methods

read

public void **read** (*NutDataInputStream in*, long *startcode*)

toString

public *String* **toString** ()

9.13 RawHandler

public class **RawHandler**

9.13.1 Methods

streamToAudioFormat

public static *AudioFormat* **streamToAudioFormat** (*StreamHeaderPacket header*)

Parses a FourCC into a AudioEncoding based on the following rules: “ALAW” = A-LAW “ULAW” = MU-LAW
 P[type][interleaving][bits] = little-endian PCM [bits][interleaving][type]P = big-endian PCM Where: [type] is
 S for signed integer, U for unsigned integer, F for IEEE float [interleaving] is D for default, P is for planar.
 [bits] is 8/16/24/32

Parameters

- **header** – The stream’s header.

Returns The AudioFormat matching this header.

toAudioInputStream

public static *AudioInputStream* **toAudioInputStream** (*Frame frame*)

toBufferedImage

public static *BufferedImage* **toBufferedImage** (*Frame frame*)

9.14 Stream

public class **Stream**

9.14.1 Fields

header

final *StreamHeaderPacket* **header**

last_pts

long **last_pts**

timeBase

final Fraction **timeBase**

9.14.2 Constructors

Stream

public **Stream**(*MainHeaderPacket* header, *StreamHeaderPacket* streamHeader)

9.15 StreamHeaderPacket

class **StreamHeaderPacket** extends *Packet*

9.15.1 Fields

AUDIO

public static final int **AUDIO**

SUBTITLE

public static final int **SUBTITLE**

USER_DATA

public static final int **USER_DATA**

VIDEO

public static final int **VIDEO**

channels

int **channels**

codecSpecificData

byte[] **codecSpecificData**

colorspaceType

long **colorspaceType**

decodeDelay

long **decodeDelay**

flags

long **flags**

fourcc

byte[] **fourcc**

height

int **height**

id

int **id**

maxPtsDistance

int **maxPtsDistance**

msbPtsShift

int **msbPtsShift**

sampleHeight

int **sampleHeight**

sampleRate

Fraction **sampleRate**

sampleWidth

int **sampleWidth**

timeBaseId

int **timeBaseId**

type

long **type**

width

int **width**

9.15.2 Methods

readBody

protected void **readBody** (*NutDataInputStream in*)

toString

public String **toString** ()

9.16 SyncPointPacket

public class **SyncPointPacket**

net.bramp.ffmpeg.options

10.1 AudioEncodingOptions

```
public class AudioEncodingOptions  
    Encoding options for audio
```

```
    Author bramp
```

10.1.1 Fields

bit_rate

```
public final long bit_rate
```

channels

```
public final int channels
```

codec

```
public final String codec
```

enabled

```
public final boolean enabled
```

quality

```
public final Integer quality
```

sample_format

```
public final String sample_format
```

sample_rate

public final int **sample_rate**

10.1.2 Constructors

AudioEncodingOptions

public **AudioEncodingOptions** (boolean *enabled*, String *codec*, int *channels*, int *sample_rate*, String *sample_format*, long *bit_rate*, Integer *quality*)

10.2 EncodingOptions

public class **EncodingOptions**

Author bramp

10.2.1 Fields

audio

public final *AudioEncodingOptions* **audio**

main

public final *MainEncodingOptions* **main**

video

public final *VideoEncodingOptions* **video**

10.2.2 Constructors

EncodingOptions

public **EncodingOptions** (*MainEncodingOptions* *main*, *AudioEncodingOptions* *audio*, *VideoEncodingOptions* *video*)

10.2.3 Methods

getAudio

public *AudioEncodingOptions* **getAudio** ()

getMain

public *MainEncodingOptions* **getMain** ()

getVideo

```
public VideoEncodingOptions getVideo ()
```

10.3 MainEncodingOptions

```
public class MainEncodingOptions
```

```
    Author bramp
```

10.3.1 Fields

duration

```
public final Long duration
```

format

```
public final String format
```

startOffset

```
public final Long startOffset
```

10.3.2 Constructors

MainEncodingOptions

```
public MainEncodingOptions (String format, Long startOffset, Long duration)
```

10.4 VideoEncodingOptions

```
public class VideoEncodingOptions
```

```
    Encoding options for video
```

```
    Author bramp
```

10.4.1 Fields

bit_rate

```
public final long bit_rate
```

codec

```
public final String codec
```

enabled

public final boolean **enabled**

filter

public final String **filter**

frame_rate

public final Fraction **frame_rate**

frames

public final Integer **frames**

height

public final int **height**

preset

public final String **preset**

width

public final int **width**

10.4.2 Constructors

VideoEncodingOptions

public **VideoEncodingOptions** (boolean *enabled*, String *codec*, Fraction *frame_rate*, int *width*, int *height*, long *bit_rate*, Integer *frames*, String *filter*, String *preset*)

net.bramp.ffmpeg.probe

11.1 FFmpegDisposition

```
public class FFmpegDisposition  
    Represents the AV_DISPOSITION_* fields
```

11.1.1 Fields

_default

```
public boolean _default
```

attached_pic

```
public boolean attached_pic
```

captions

```
public boolean captions
```

clean_effects

```
public boolean clean_effects
```

comment

```
public boolean comment
```

descriptions

```
public boolean descriptions
```

dub

public boolean **dub**

forced

public boolean **forced**

hearing_impaired

public boolean **hearing_impaired**

karaoke

public boolean **karaoke**

lyrics

public boolean **lyrics**

metadata

public boolean **metadata**

original

public boolean **original**

visual_impaired

public boolean **visual_impaired**

11.2 FFmpegError

public class **FFmpegError**

11.2.1 Fields

code

public int **code**

string

public `String` **string**

11.3 FFmpegFormat

```
public class FFmpegFormat
```

11.3.1 Fields

bit_rate

```
public long bit_rate  
    Bitrate
```

duration

```
public double duration  
    Duration in seconds
```

filename

```
public String filename
```

format_long_name

```
public String format_long_name
```

format_name

```
public String format_name
```

nb_programs

```
public int nb_programs
```

nb_streams

```
public int nb_streams
```

probe_score

```
public int probe_score
```

size

```
public long size  
    File size in bytes
```

start_time

public double **start_time**

tags

public Map<String, String> **tags**

11.4 FFmpegProbeResult

public class **FFmpegProbeResult**
 TODO Make this immutable

11.4.1 Fields

error

public *FFmpegError* **error**

format

public *FFmpegFormat* **format**

streams

public List<*FFmpegStream*> **streams**

11.4.2 Methods

getError

public *FFmpegError* **getError** ()

getFormat

public *FFmpegFormat* **getFormat** ()

getStreams

public List<*FFmpegStream*> **getStreams** ()

hasError

public boolean **hasError** ()

11.5 FFmpegStream

```
public class FFmpegStream
```

11.5.1 Fields

avg_frame_rate

```
public Fraction avg_frame_rate
```

bit_rate

```
public long bit_rate
```

bits_per_raw_sample

```
public int bits_per_raw_sample
```

bits_per_sample

```
public int bits_per_sample
```

channel_layout

```
public String channel_layout
```

channels

```
public int channels
```

chroma_location

```
public String chroma_location
```

codec_long_name

```
public String codec_long_name
```

codec_name

```
public String codec_name
```

codec_tag

```
public String codec_tag
```

codec_tag_string

public String **codec_tag_string**

codec_time_base

public Fraction **codec_time_base**

codec_type

public *CodecType* **codec_type**

display_aspect_ratio

public String **display_aspect_ratio**

disposition

public *FFmpegDisposition* **disposition**

duration

public double **duration**

duration_ts

public long **duration_ts**

has_b_frames

public int **has_b_frames**

index

public int **index**

is_avc

public String **is_avc**

level

public int **level**

max_bit_rate

public long **max_bit_rate**

nal_length_size

public String **nal_length_size**

nb_frames

public long **nb_frames**

pix_fmt

public String **pix_fmt**

profile

public String **profile**

r_frame_rate

public Fraction **r_frame_rate**

refs

public int **refs**

sample_aspect_ratio

public String **sample_aspect_ratio**

sample_fmt

public String **sample_fmt**

sample_rate

public int **sample_rate**

start_pts

public int **start_pts**

start_time

public double **start_time**

tags

public Map<String, String> **tags**

time_base

public Fraction **time_base**

width

public int **width**

11.6 FFmpegStream.CodecType

public enum **CodecType**

11.6.1 Enum Constants

AUDIO

public static final *FFmpegStream.CodecType* **AUDIO**

VIDEO

public static final *FFmpegStream.CodecType* **VIDEO**

net.bramp.ffmpeg.progress

12.1 AbstractSocketProgressParser

public abstract class **AbstractSocketProgressParser** implements *ProgressParser*

12.1.1 Fields

parser

final *StreamProgressParser* **parser**

thread

Thread **thread**

12.1.2 Constructors

AbstractSocketProgressParser

public **AbstractSocketProgressParser** (*ProgressListener* listener)

12.1.3 Methods

close

public void **close** ()

createUri

static URI **createUri** (*String* scheme, *InetAddress* address, int port)

Creates a URL to parse to FFmpeg based on the scheme, address and port. TODO Move this method to somewhere better.

Parameters

- **scheme** –

- `address` –

- `port` –

Throws

- `URISyntaxException` –

getRunnable

protected abstract `Runnable` **getRunnable** (`CountDownLatch startSignal`)

getThreadName

protected abstract `String` **getThreadName** ()

start

public synchronized void **start** ()

Throws

- `IllegalThreadStateException` – if the parser was already started.

stop

public void **stop** ()

12.2 Progress

public class **Progress**
 TODO Change to be immutable

12.2.1 Fields

bitrate

public long **bitrate**

drop_frames

public long **drop_frames**

dup_frames

public long **dup_frames**

fps

public Fraction **fps**

frame

public long **frame**

out_time_ms

public long **out_time_ms**

progress

public String **progress**

speed

public float **speed**

total_size

public long **total_size**

12.2.2 Constructors

Progress

public **Progress** ()

Progress

public **Progress** (long *frame*, float *fps*, long *bitrate*, long *total_size*, long *out_time_ms*, long *dup_frames*, long *drop_frames*, float *speed*, String *progress*)

12.2.3 Methods

equals

public boolean **equals** (Object *o*)

hashCode

public int **hashCode** ()

isEnd

public boolean **isEnd** ()

parseLine

protected boolean **parseLine** (*String line*)
Parses values from the line, into this object.

Parameters

- **line** – A single line of output from ffmpeg

Returns true if the record is finished

toString

public *String* **toString** ()

12.3 ProgressListener

public interface **ProgressListener**

12.3.1 Methods

progress

void **progress** (*Progress progress*)

12.4 ProgressParser

public interface **ProgressParser** extends *Closeable*
Parses the FFmpeg progress fields

12.4.1 Methods

getUri

URI **getUri** ()
The URL to parse to FFmpeg to communicate with this parser
Returns The URI to communicate with FFmpeg.

start

void **start** ()

stop

```
void stop ()
```

12.5 StreamProgressParser

```
public class StreamProgressParser
```

12.5.1 Fields

listener

```
final ProgressListener listener
```

12.5.2 Constructors

StreamProgressParser

```
public StreamProgressParser (ProgressListener listener)
```

12.5.3 Methods

processReader

```
public void processReader (Reader reader)
```

processStream

```
public void processStream (InputStream stream)
```

12.6 TcpProgressParser

```
public class TcpProgressParser extends AbstractSocketProgressParser
```

12.6.1 Fields

address

```
final URI address
```

server

```
final ServerSocket server
```

12.6.2 Constructors

TcpProgressParser

```
public TcpProgressParser (ProgressListener listener)
```

TcpProgressParser

```
public TcpProgressParser (ProgressListener listener, int port, InetAddress addr)
```

12.6.3 Methods

getRunnable

```
protected Runnable getRunnable (CountDownLatch startSignal)
```

getThreadName

```
protected String getThreadName ()
```

getUri

```
public URI getUri ()
```

stop

```
public synchronized void stop ()
```

12.7 TcpProgressParserRunnable

```
class TcpProgressParserRunnable implements Runnable
```

12.7.1 Fields

parser

```
final StreamProgressParser parser
```

server

```
final ServerSocket server
```

startSignal

```
final CountDownLatch startSignal
```


12.7.2 Constructors

TcpProgressParserRunnable

```
public TcpProgressParserRunnable (StreamProgressParser parser, ServerSocket server, CountdownLatch startSignal)
```

12.7.3 Methods

run

```
public void run ()
```

12.8 UdpProgressParser

```
public class UdpProgressParser extends AbstractSocketProgressParser
```

12.8.1 Fields

address

```
final URI address
```

socket

```
final DatagramSocket socket
```

12.8.2 Constructors

UdpProgressParser

```
public UdpProgressParser (ProgressListener listener)
```

UdpProgressParser

```
public UdpProgressParser (ProgressListener listener, int port, InetAddress addr)
```

12.8.3 Methods

getRunnable

```
protected Runnable getRunnable (CountDownLatch startSignal)
```

getThreadName

```
protected String getThreadName ()
```

getUri

```
public URI getUri ()
```

stop

```
public synchronized void stop ()
```

12.9 UdpProgressParserRunnable

class **UdpProgressParserRunnable** implements `Runnable`

12.9.1 Fields

MAX_PACKET_SIZE

```
static final int MAX_PACKET_SIZE
```

parser

```
final StreamProgressParser parser
```

socket

```
final DatagramSocket socket
```

startSignal

```
final CountdownLatch startSignal
```

12.9.2 Constructors

UdpProgressParserRunnable

```
public UdpProgressParserRunnable (StreamProgressParser parser, DatagramSocket socket, CountdownLatch startSignal)
```

12.9.3 Methods

run

```
public void run ()
```

Indices and tables

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

Symbols

`_default` (Java field), 75

A

AbstractSocketProgressParser (Java class), 83
 AbstractSocketProgressParser(ProgressListener) (Java constructor), 83
 addExtraArgs(String) (Java method), 16, 24
 addInput(FFmpegProbeResult) (Java method), 17
 addInput(String) (Java method), 17
 addMetaTag(MetadataSpecifier, String, String) (Java method), 25
 addMetaTag(String, String) (Java method), 25
 addOutput(FFmpegOutputBuilder) (Java method), 17
 addOutput(String) (Java method), 17
 addOutput(URI) (Java method), 17
 addProgress(URI) (Java method), 17
 address (Java field), 87, 89
 addStdoutOutput() (Java method), 18
 applies(MappingContext) (Java method), 53
 attached_pic (Java field), 75
 Attachment (Java field), 35
 AUDIO (Java field), 41, 68, 82
 Audio (Java field), 35
 audio (Java field), 52, 72
 audio_bit_rate (Java field), 21
 audio_bit_stream_filter (Java field), 21
 audio_channels (Java field), 21
 audio_codec (Java field), 21
 AUDIO_DEPTH_DBL (Java field), 6
 AUDIO_DEPTH_FLT (Java field), 7
 AUDIO_DEPTH_S16 (Java field), 7
 AUDIO_DEPTH_S32 (Java field), 7
 AUDIO_DEPTH_U8 (Java field), 7
 audio_enabled (Java field), 21
 AUDIO_FORMAT_DBL (Java field), 7
 AUDIO_FORMAT_FLT (Java field), 7
 AUDIO_FORMAT_S16 (Java field), 7
 AUDIO_FORMAT_S32 (Java field), 7
 AUDIO_FORMAT_U8 (Java field), 7

AUDIO_MONO (Java field), 7
 audio_quality (Java field), 21
 AUDIO_SAMPLE_11025 (Java field), 7
 AUDIO_SAMPLE_12000 (Java field), 8
 AUDIO_SAMPLE_16000 (Java field), 8
 AUDIO_SAMPLE_22050 (Java field), 8
 AUDIO_SAMPLE_32000 (Java field), 8
 AUDIO_SAMPLE_44100 (Java field), 8
 AUDIO_SAMPLE_48000 (Java field), 8
 AUDIO_SAMPLE_8000 (Java field), 8
 AUDIO_SAMPLE_96000 (Java field), 8
 audio_sample_format (Java field), 21
 audio_sample_rate (Java field), 21
 AUDIO_STEREO (Java field), 8
 AudioEncodingOptions (Java class), 71
 AudioEncodingOptions(boolean, String, int, int, String, long, Integer) (Java constructor), 72
 AudioWrapper (Java class), 51
 AudioWrapper(AudioEncodingOptions) (Java constructor), 52
 avg_frame_rate (Java field), 79

B

bit_rate (Java field), 71, 73, 77, 79
 bitrate (Java field), 84
 BITRATE_REGEX (Java field), 12
 bits_per_raw_sample (Java field), 79
 bits_per_sample (Java field), 79
 BROADCAST_MODE (Java field), 58
 buffer (Java field), 44
 build() (Java method), 18
 build(FFmpegBuilder, int) (Java method), 25
 build(int) (Java method), 25
 builder (Java field), 48, 49
 builder() (Java method), 10
 buildOptions() (Java method), 26

C

canDecode (Java field), 39
 canDemux (Java field), 41

canEncode (Java field), 39
 canMux (Java field), 41
 captions (Java field), 75
 channel_layout (Java field), 79
 channels (Java field), 68, 71, 79
 chapter(int) (Java method), 32
 checksum (Java field), 66
 checkValidKey(String) (Java method), 32
 checkValidStream(URI) (Java method), 26
 chroma_location (Java field), 79
 clazz (Java field), 37
 clean_effects (Java field), 75
 clearInputs() (Java method), 18
 close() (Java method), 83
 code (Java field), 76
 Codec (Java class), 39
 codec (Java field), 71, 73
 Codec(String, String, String) (Java constructor), 40
 codec_long_name (Java field), 79
 codec_name (Java field), 79
 codec_tag (Java field), 79
 codec_tag_string (Java field), 80
 codec_time_base (Java field), 80
 codec_type (Java field), 80
 codecs (Java field), 9
 codecs() (Java method), 10
 CODECS_REGEX (Java field), 8
 codecSpecificData (Java field), 68
 CodecType (Java enum), 82
 colorspaceType (Java field), 69
 comment (Java field), 75
 count (Java field), 60
 crc (Java field), 43, 60
 CRC32InputStream (Java class), 43
 CRC32InputStream(InputStream) (Java constructor), 43
 create(Gson, TypeToken) (Java method), 37
 createJob(FFmpegBuilder) (Java method), 11
 createJob(FFmpegBuilder, ProgressListener) (Java method), 11
 createProgressParser(ProgressListener) (Java method), 10
 createTwoPassJob(FFmpegBuilder) (Java method), 11
 createUri(String, InetAddress, int) (Java method), 83

D

Data (Java field), 36
 data (Java field), 56
 dataSizeLsb (Java field), 57
 dataSizeMul (Java field), 57
 DEBUG (Java field), 20
 decodeDelay (Java field), 69
 DEFAULT_PATH (Java field), 8, 13
 deletePassLog() (Java method), 49
 descriptions (Java field), 75
 disableAudio() (Java method), 26

disableSubtitle() (Java method), 26
 disableVideo() (Java method), 26
 display_aspect_ratio (Java field), 80
 disposition (Java field), 80
 done() (Java method), 26
 drop_frames (Java field), 84
 dub (Java field), 76
 dup_frames (Java field), 84
 duration (Java field), 21, 73, 77, 80
 duration_ts (Java field), 80

E

elision (Java field), 58
 enabled (Java field), 71, 74
 EncodingOptions (Java class), 72
 EncodingOptions(MainEncodingOptions, AudioEncodingOptions, VideoEncodingOptions) (Java constructor), 72
 end (Java field), 66
 endCrcRange (Java field), 60
 equals(Object) (Java method), 40, 42, 85
 ERROR (Java field), 20
 error (Java field), 78
 EXPERIMENTAL (Java field), 19
 extra_args (Java field), 15, 21

F

FAILED (Java field), 48
 FATAL (Java field), 20
 FFcommon (Java class), 5
 FFcommon(String) (Java constructor), 5
 FFcommon(String, ProcessFunction) (Java constructor), 5
 FFmpeg (Java class), 6
 FFMPEG (Java field), 9
 ffmpeg (Java field), 11, 47
 FFmpeg() (Java constructor), 9
 FFmpeg(ProcessFunction) (Java constructor), 9
 FFmpeg(String) (Java constructor), 9
 FFmpeg(String, ProcessFunction) (Java constructor), 10
 FFmpegBuilder (Java class), 15
 FFmpegDisposition (Java class), 75
 FFmpegError (Java class), 76
 FFmpegExecutor (Java class), 11
 FFmpegExecutor() (Java constructor), 11
 FFmpegExecutor(FFmpeg) (Java constructor), 11
 FFmpegExecutor(FFmpeg, FFprobe) (Java constructor), 11
 FFmpegFormat (Java class), 77
 FFmpegJob (Java class), 47
 FFmpegJob(FFmpeg) (Java constructor), 47
 FFmpegJob(FFmpeg, ProgressListener) (Java constructor), 47
 FFmpegOutputBuilder (Java class), 20

FFmpegOutputBuilder() (Java constructor), 24
 FFmpegOutputBuilder(FFmpegBuilder, String) (Java constructor), 24
 FFmpegOutputBuilder(FFmpegBuilder, URI) (Java constructor), 24
 FFmpegProbeResult (Java class), 78
 FFmpegStream (Java class), 79
 FFmpegUtils (Java class), 12
 FFmpegUtils() (Java constructor), 12
 FFprobe (Java class), 13
 FFPROBE (Java field), 13
 ffprobe (Java field), 11
 FFprobe() (Java constructor), 13
 FFprobe(ProcessFunction) (Java constructor), 13
 FFprobe(String, ProcessFunction) (Java constructor), 13
 filename (Java field), 21, 77
 filter (Java field), 74
 FINISHED (Java field), 48
 FLAG_CHECKSUM (Java field), 55
 FLAG_CODED (Java field), 55
 FLAG_CODED_PTS (Java field), 55
 FLAG_EOR (Java field), 55
 FLAG_HEADER_IDX (Java field), 55
 FLAG_INVALID (Java field), 55
 FLAG_KEY (Java field), 56
 FLAG_MATCH_TIME (Java field), 56
 FLAG_RESERVED (Java field), 56
 FLAG_SIZE_MSB (Java field), 56
 FLAG_SM_DATA (Java field), 56
 FLAG_STREAM_ID (Java field), 56
 flags (Java field), 56–58, 69
 footer (Java field), 64
 forced (Java field), 76
 Format (Java class), 41
 format (Java field), 15, 22, 73, 78
 Format(String, String, String) (Java constructor), 41
 format_long_name (Java field), 77
 format_name (Java field), 77
 formats (Java field), 9
 formats() (Java method), 10
 FORMATS_REGEX (Java field), 9
 forwardPtr (Java field), 66
 fourcc (Java field), 69
 fps (Java field), 85
 FPS_23_976 (Java field), 9
 FPS_24 (Java field), 9
 FPS_29_97 (Java field), 9
 FPS_30 (Java field), 9
 FractionAdapter (Java class), 3
 FractionAdapter() (Java constructor), 3
 Frame (Java class), 55
 frame (Java field), 85
 frame(Frame) (Java method), 64
 frame_rate (Java field), 74

FrameCode (Java class), 57
 frameCodes (Java field), 59
 frames (Java field), 74

G

getAudio() (Java method), 72
 getCanDecode() (Java method), 40
 getCanDemux() (Java method), 42
 getCanEncode() (Java method), 40
 getCanMux() (Java method), 42
 getCRC() (Java method), 60
 getError() (Java method), 78
 getFilename() (Java method), 26
 getFormat() (Java method), 78
 getGson() (Java method), 12
 getLongName() (Java method), 40, 42
 getMain() (Java method), 72
 getName() (Java method), 40, 42
 getOverrideOutputFiles() (Java method), 18
 getPath() (Java method), 6, 10
 getRunnable(CountDownLatch) (Java method), 84, 88, 89
 getState() (Java method), 48
 getStreams() (Java method), 78
 getThreadName() (Java method), 84, 88, 89
 getType() (Java method), 40
 getUri() (Java method), 27, 86, 88, 90
 getValue() (Java method), 43
 getVideo() (Java method), 73
 global() (Java method), 32
 gson (Java field), 12, 13

H

has_b_frames (Java field), 80
 hasError() (Java method), 78
 hashCode() (Java method), 40, 42, 85
 HEADER (Java field), 62
 header (Java field), 63, 65, 67
 headerIdx (Java field), 57
 hearing_impaired (Java field), 76
 height (Java field), 69, 74

I

id (Java field), 69
 id(int) (Java method), 33
 in (Java field), 60, 63
 INDEX (Java field), 65
 index (Java field), 80
 IndexPacket (Java class), 58
 INFO (Java field), 20, 65
 InfoPacket (Java class), 58
 inputProbes (Java field), 15
 inputs (Java field), 15
 is_avc (Java field), 80

isEnd() (Java method), 86
 isFFmpeg() (Java method), 10
 isFFprobe() (Java method), 13
 isKnownStartcode(long) (Java method), 63
 isValidSize(int) (Java method), 27

K

karaoke (Java field), 76

L

last_pts (Java field), 68
 level (Java field), 80
 listener (Java field), 47, 63, 87
 LOG (Java field), 13–15, 64
 log() (Java method), 45
 LOG_CHAR (Java field), 44
 logger (Java field), 44
 LoggingFilterReader (Java class), 44
 LoggingFilterReader(Reader, Logger) (Java constructor), 45
 longName (Java field), 39, 41
 LowercaseEnumTypeAdapterFactory (Java class), 37
 lyrics (Java field), 76

M

MAIN (Java field), 65
 main (Java field), 72
 MainEncodingOptions (Java class), 73
 MainEncodingOptions(String, Long, Long) (Java constructor), 73
 MainHeaderPacket (Java class), 58
 MainHeaderPacket() (Java constructor), 59
 map(AudioEncodingOptions, FFmpegOutputBuilder) (Java method), 51
 map(EncodingOptions, FFmpegOutputBuilder) (Java method), 51
 map(MainEncodingOptions, FFmpegOutputBuilder) (Java method), 51
 map(VideoEncodingOptions, FFmpegOutputBuilder) (Java method), 51
 Mapper (Java class), 51
 mark(int) (Java method), 43
 markSupported() (Java method), 43
 matchTimeDelta (Java field), 57
 max_bit_rate (Java field), 81
 MAX_PACKET_SIZE (Java field), 90
 maxDistance (Java field), 59
 maxPtsDistance (Java field), 69
 meta_tags (Java field), 22
 metaData (Java field), 56
 metadata (Java field), 76
 MetadataSpecifier (Java class), 32
 millisecondsToString(long) (Java method), 12
 minorVersion (Java field), 59

msbPtsShift (Java field), 69

N

nal_length_size (Java field), 81
 name (Java field), 39, 41
 NamedBitsetAdapter (Java class), 37
 NamedBitsetAdapter(Class) (Java constructor), 38
 nb_frames (Java field), 81
 nb_programs (Java field), 77
 nb_streams (Java field), 77
 net.bramp.commons.lang3.math.gson (package), 3
 net.bramp.ffmpeg (package), 5
 net.bramp.ffmpeg.builder (package), 15
 net.bramp.ffmpeg.gson (package), 37
 net.bramp.ffmpeg.info (package), 39
 net.bramp.ffmpeg.io (package), 43
 net.bramp.ffmpeg.job (package), 47
 net.bramp.ffmpeg.modelmapper (package), 51
 net.bramp.ffmpeg.nut (package), 55
 net.bramp.ffmpeg.options (package), 71
 net.bramp.ffmpeg.probe (package), 75
 net.bramp.ffmpeg.progress (package), 83
 NORMAL (Java field), 19
 notDefault (Java field), 52
 NotDefaultCondition (Java class), 52
 NutDataInputStream (Java class), 59
 NutDataInputStream(InputStream) (Java constructor), 60
 NutReader (Java class), 62
 NutReader(InputStream, NutReaderListener) (Java constructor), 63
 NutReaderListener (Java interface), 64

O

offset() (Java method), 60
 original (Java field), 76
 out_time_ms (Java field), 85
 outputs (Java field), 15
 override (Java field), 16
 overrideOutputFiles(boolean) (Java method), 18

P

Packet (Java class), 64
 PacketFooter (Java class), 66
 PacketHeader (Java class), 66
 PANIC (Java field), 20
 parent (Java field), 22
 parseBitrate(String) (Java method), 12
 parseLine(String) (Java method), 86
 parser (Java field), 83, 88, 90
 pass (Java field), 16
 pass_directory (Java field), 16
 pass_padding_bitrate (Java field), 22
 pass_prefix (Java field), 16
 passlogPrefix (Java field), 49

path (Java field), 5
 path(List) (Java method), 6
 pix_fmt (Java field), 81
 preset (Java field), 74
 probe(String) (Java method), 14
 probe(String, String) (Java method), 14
 probe_score (Java field), 77
 ProcessFunction (Java interface), 14
 processReader(Reader) (Java method), 87
 processStream(InputStream) (Java method), 87
 ProcessUtils (Java class), 45
 profile (Java field), 81
 program(int) (Java method), 32, 34
 program(int, int) (Java method), 34
 Progress (Java class), 84
 progress (Java field), 16, 85
 Progress() (Java constructor), 85
 Progress(long, float, long, long, long, long, long, float, String) (Java constructor), 85
 progress(Progress) (Java method), 86
 ProgressListener (Java interface), 86
 ProgressParser (Java interface), 86
 pts (Java field), 56
 ptsDelta (Java field), 58
 PureVideo (Java field), 36

Q

quality (Java field), 71
 QUIET (Java field), 20

R

r_frame_rate (Java field), 81
 RawHandler (Java class), 67
 read() (Java method), 44, 45, 63
 read(byte[]) (Java method), 44
 read(byte[], int, int) (Java method), 44
 read(char[], int, int) (Java method), 45
 read(JsonReader) (Java method), 3, 38
 read(NutDataInputStream) (Java method), 66
 read(NutDataInputStream, long) (Java method), 65, 67
 read(NutReader, NutDataInputStream, int) (Java method), 57
 read_at_native_frame_rate (Java field), 16
 readAtNativeFrameRate() (Java method), 18
 readBody(NutDataInputStream) (Java method), 59, 65, 70
 readBoolean() (Java method), 60
 readBoolean(JsonReader) (Java method), 38
 readByte() (Java method), 60
 readChar() (Java method), 61
 readDouble() (Java method), 61
 readFileId() (Java method), 63
 readFloat() (Java method), 61
 readFully(byte[]) (Java method), 61

readFully(byte[], int, int) (Java method), 61
 readInt() (Java method), 61
 readLine() (Java method), 61
 readLong() (Java method), 61
 readMetaData(NutDataInputStream) (Java method), 57
 readReservedHeaders() (Java method), 64
 readShort() (Java method), 61
 readSignedVarInt() (Java method), 61
 readStartCode() (Java method), 61
 readUnsignedByte() (Java method), 62
 readUnsignedShort() (Java method), 62
 readUTF() (Java method), 62
 readVarArray() (Java method), 62
 readVarInt() (Java method), 62
 readVarLong() (Java method), 62
 refs (Java field), 81
 reservedCount (Java field), 58
 reset() (Java method), 44
 resetCRC() (Java method), 62
 resetCrc() (Java method), 44
 run() (Java method), 49, 89, 90
 run(FFmpegBuilder, ProgressListener) (Java method), 10
 run(List) (Java method), 6, 10, 14
 runFunc (Java field), 5
 RUNNING (Java field), 48
 RunProcessFunction (Java class), 14

S

sample_aspect_ratio (Java field), 81
 sample_fmt (Java field), 81
 sample_format (Java field), 71
 sample_rate (Java field), 72, 81
 sampleHeight (Java field), 69
 sampleRate (Java field), 69
 sampleWidth (Java field), 69
 seekToPacketFooter(NutDataInputStream) (Java method), 65
 server (Java field), 87, 88
 setAudioBitDepth(String) (Java method), 27
 setAudioBitRate(long) (Java method), 27
 setAudioBitStreamFilter(String) (Java method), 27
 setAudioChannels(int) (Java method), 27
 setAudioCodec(String) (Java method), 27
 setAudioQuality(int) (Java method), 28
 setAudioSampleFormat(String) (Java method), 28
 setAudioSampleRate(int) (Java method), 28
 setComplexVideoFilter(String) (Java method), 28
 setDuration(long, TimeUnit) (Java method), 28
 setField(T, String, boolean) (Java method), 38
 setFilename(String) (Java method), 29
 setFormat(String) (Java method), 18, 29
 setFrames(int) (Java method), 29
 setInput(FFmpegProbeResult) (Java method), 18
 setInput(String) (Java method), 18

- setPass(int) (Java method), 18
 - setPassDirectory(String) (Java method), 18
 - setPassPaddingBitrate(long) (Java method), 29
 - setPassPrefix(String) (Java method), 19
 - setStartOffset(long, TimeUnit) (Java method), 19, 29
 - setStrict(FFmpegBuilder.Strict) (Java method), 29
 - setTargetSize(long) (Java method), 29
 - setUri(URI) (Java method), 30
 - setUserAgent(String) (Java method), 19
 - setVerbosity(Verbosity) (Java method), 19
 - setVideoBitRate(long) (Java method), 30
 - setVideoBitStreamFilter(String) (Java method), 30
 - setVideoCodec(String) (Java method), 30
 - setVideoCopyInkf(boolean) (Java method), 30
 - setVideoFilter(String) (Java method), 30
 - setVideoFrameRate(double) (Java method), 31
 - setVideoFrameRate(Fraction) (Java method), 30
 - setVideoFrameRate(int, int) (Java method), 30
 - setVideoHeight(int) (Java method), 31
 - setVideoMovFlags(String) (Java method), 31
 - setVideoPixelFormat(String) (Java method), 31
 - setVideoPreset(String) (Java method), 31
 - setVideoQuality(int) (Java method), 31
 - setVideoResolution(int, int) (Java method), 31
 - setVideoResolution(String) (Java method), 31
 - setVideoWidth(int) (Java method), 31
 - sideData (Java field), 56
 - SinglePassFFmpegJob (Java class), 48
 - SinglePassFFmpegJob(FFmpeg, FFmpegBuilder) (Java constructor), 48
 - SinglePassFFmpegJob(FFmpeg, FFmpegBuilder, ProgressListener) (Java constructor), 49
 - size (Java field), 77
 - skip(long) (Java method), 44
 - skipBytes(int) (Java method), 62
 - socket (Java field), 89, 90
 - spec (Java field), 32, 33
 - spec() (Java method), 33, 34
 - speed (Java field), 85
 - start() (Java method), 84, 86
 - start_pts (Java field), 81
 - start_time (Java field), 78, 82
 - Startcode (Java enum), 65
 - startcode (Java field), 66
 - startCrcRange (Java field), 60
 - startOffset (Java field), 16, 22, 73
 - startSignal (Java field), 88, 90
 - State (Java enum), 48
 - state (Java field), 47
 - stop() (Java method), 84, 87, 88, 90
 - Stream (Java class), 67
 - STREAM (Java field), 65
 - stream (Java field), 57
 - stream(int) (Java method), 33, 34
 - Stream(MainHeaderPacket, StreamHeaderPacket) (Java constructor), 68
 - stream(Stream) (Java method), 64
 - stream(StreamSpecifier) (Java method), 33
 - stream(StreamSpecifierType) (Java method), 33, 34
 - stream(StreamSpecifierType, int) (Java method), 33, 34
 - streamCount (Java field), 59
 - StreamHeaderPacket (Java class), 68
 - streamId (Java field), 58
 - StreamProgressParser (Java class), 87
 - StreamProgressParser(ProgressListener) (Java constructor), 87
 - streams (Java field), 63, 78
 - StreamSpecifier (Java class), 33
 - StreamSpecifierType (Java enum), 35
 - streamToAudioFormat(StreamHeaderPacket) (Java method), 67
 - Strict (Java enum), 19
 - STRICT (Java field), 19
 - strict (Java field), 22
 - string (Java field), 76
 - SUBTITLE (Java field), 41, 68
 - Subtitle (Java field), 36
 - subtitle_enabled (Java field), 22
 - SYNCPOINT (Java field), 66
 - SyncPointPacket (Java class), 70
- ## T
- tag(String) (Java method), 35
 - tag(String, String) (Java method), 35
 - tags (Java field), 78, 82
 - targetSize (Java field), 22
 - TcpProgressParser (Java class), 87
 - TcpProgressParser(ProgressListener) (Java constructor), 88
 - TcpProgressParser(ProgressListener, int, InetAddress) (Java constructor), 88
 - TcpProgressParserRunnable (Java class), 88
 - TcpProgressParserRunnable(StreamProgressParser, ServerSocket, CountdownLatch) (Java constructor), 89
 - thread (Java field), 83
 - throwOnError(Process) (Java method), 6
 - throwWarnings (Java field), 22
 - time_base (Java field), 82
 - timeBase (Java field), 59, 68
 - timeBaseId (Java field), 70
 - toAudioInputStream(Frame) (Java method), 67
 - toBufferedImage(Frame) (Java method), 67
 - toString() (Java method), 40, 42, 57–59, 65–67, 70, 86
 - total_size (Java field), 85
 - TwoPassFFmpegJob (Java class), 49
 - TwoPassFFmpegJob(FFmpeg, FFmpegBuilder) (Java constructor), 49

TwoPassFFmpegJob(FFmpeg, FFmpegBuilder, ProgressListener) (Java constructor), 49
 Type (Java enum), 40
 type (Java field), 39, 70

U

UdpProgressParser (Java class), 89
 UdpProgressParser(ProgressListener) (Java constructor), 89
 UdpProgressParser(ProgressListener, int, InetAddress) (Java constructor), 89
 UdpProgressParserRunnable (Java class), 90
 UdpProgressParserRunnable(StreamProgressParser, DatagramSocket, CountdownLatch) (Java constructor), 90
 UNOFFICAL (Java field), 19
 uri (Java field), 22
 usable() (Java method), 35
 useOptions(AudioEncodingOptions) (Java method), 32
 useOptions(EncodingOptions) (Java method), 32
 useOptions(MainEncodingOptions) (Java method), 32
 useOptions(VideoEncodingOptions) (Java method), 32
 user_agent (Java field), 16
 USER_DATA (Java field), 68

V

VERBOSE (Java field), 20
 Verbosity (Java enum), 20
 verbosity (Java field), 16
 version (Java field), 5, 59
 version() (Java method), 6
 VERY (Java field), 19
 VIDEO (Java field), 41, 68, 82
 Video (Java field), 36
 video (Java field), 52, 72
 video_bit_rate (Java field), 22
 video_bit_stream_filter (Java field), 23
 video_codec (Java field), 23
 video_copyinkf (Java field), 23
 video_enabled (Java field), 23
 video_filter (Java field), 23
 video_filter_complex (Java field), 23
 video_frame_rate (Java field), 23
 video_frames (Java field), 23
 video_height (Java field), 23
 video_movflags (Java field), 23
 video_pixel_format (Java field), 23
 video_preset (Java field), 24
 video_quality (Java field), 24
 video_size (Java field), 24
 video_width (Java field), 24
 VideoEncodingOptions (Java class), 73

VideoEncodingOptions(boolean, String, Fraction, int, int, long, Integer, String, String) (Java constructor), 74

VideoWrapper (Java class), 52

VideoWrapper(VideoEncodingOptions) (Java constructor), 52

visual_impaired (Java field), 76

W

waitForWithTimeout(Process, long, TimeUnit) (Java method), 45

WAITING (Java field), 48

WARNING (Java field), 20

width (Java field), 70, 74, 82

wrapInReader(Process) (Java method), 6

write(JsonWriter, Fraction) (Java method), 3

write(JsonWriter, T) (Java method), 38