NAME

sox_ng – Sound eXchange_ng, another Swiss Army knife of audio manipulation

SYNOPSIS

sox_ng [*global-options*] [*format-options*] *infile1*
[[*format-options*] *infile2*] ... [*format-options*] *outfile*
[*effect* [*effect-options*]] ...

play_ng [*global-options*] [*format-options*] *infile1*
[[*format-options*] *infile2*] ... [*format-options*]
[*effect* [*effect-options*]] ...

rec_ng [*global-options*] [*format-options*] *outfile*
[*effect* [*effect-options*]] ...

DESCRIPTION

Introduction

SoX reads and writes audio files in most popular formats and can optionally apply effects to them. It can combine multiple input sources, synthesize audio and, on many systems, act as a general purpose audio player or as a multitrack audio recorder. It can also split the input into multiple output files.

All SoX functionality is available using just the **sox_ng** command. To simplify playing and recording audio, if SoX is invoked as **play_ng**, the output file is automatically set to be the default sound device and, if invoked as **rec_ng**, the default sound device is used as an input source. Additionally, the **soxi_ng** command provides a convenient way to query audio file header information.

The heart of SoX is a library called **libsox_ng**. Those interested in extending SoX or using it in other programs should refer to the **libsox_ng** manual page.

SoX is a command-line audio processing tool particularly suited to making quick, simple edits and to batch processing. If you need an interactive, graphical audio editor, use **audacity**(1).

* * *

The overall SoX processing chain can be summarized as follows:

Input(s) → Combiner → Effects → Output(s)

On the SoX command line, the positions of the Output(s) and the Effects are swapped w.r.t. the logical flow just shown and, while options pertaining to files are placed before their respective file names, the opposite is true for effects. To show how this works in practice, here is a selection of examples of how SoX might be used. The simple

```
sox_ng recital.au recital.wav
```

translates an audio file in Sun AU format to a Microsoft WAV file, while

```
sox_ng recital.au -b 16 recital.wav channels 1 rate 16k fade 3 norm
```

performs the same format translation but also applies four effects (down-mix to one channel, sample rate change, fade in and normalize) and stores the result at a bit-depth of 16.

```
sox_ng -r 16k -e signed -b 8 -c 1 voice-memo.raw voice-memo.wav
```

converts 'raw' (a.k.a. 'headerless') audio to a self-describing file format,

```
sox_ng slow.aiff fixed.aiff speed 1.027
```

adjusts audio speed,

```
sox_ng short.wav long.wav longer.wav
```

concatenates two audio files and

```
sox_ng -m music.mp3 voice.wav mixed.flac
```

mixes together two audio files.

```
play_ng "The Moonbeams/Greatest/*.ogg" bass +3
```

plays a collection of audio files applying a bass boosting effect,

```
play_ng -n -c1 synth sin %-12 sin %-9 sin %-5 sin %-2 fade h 0.1 1 0.1
```

plays a synthesized 'A minor seventh' chord with a pipe organ sound,

```
rec_ng -c 2 radio.aiff trim 0 30:00
```

records half an hour of stereo audio and

```
play_ng -q take1.aiff & rec -M take1.aiff take1-dub.aiff
```

(with a POSIX shell and where supported by hardware) records a new track in a multitrack recording. Finally,

```
rec_ng -r 44100 -b 16 -e signed-integer -p \
   silence 1 0.50 0.1% 1 10:00 0.1% | \
   sox_ng -p song.ogg silence 1 0.50 0.1% 1 2.0 0.1% : \
   newfile : restart
```

records a stream of audio such as an LP or cassette and splits it into multiple audio files at points where there are two seconds of silence. Also, it does not start recording until it detects some sound and stops after it sees ten minutes of silence.

The above is just an overview of SoX's capabilities. Detailed explanations of how to use all SoX parameters, file formats and effects can be found below in this manual, in **soxformat_ng**(7) and in **soxi_ng**(1).

File Format Types

SoX can work with 'self-describing' and 'raw' audio files. 'Self-describing' formats (e.g. WAV, FLAC, MP3) have a header that completely describes the signal and encoding attributes of the audio data that follows. 'raw' or 'headerless' formats do not contain this information, so the audio characteristics of these must be described on the SoX command line, except for a few which can be inferred from the filename extension such as **.gsm**, always **−c1 −r8000 −e gsm**, and named raw formats like **.f32**, .s16 and .ul which give the encoding but not the sample rate or number of channels.

The following four characteristics are used to describe the format of audio data:

sample rate
: The sample rate in samples per second ('Hertz' or 'Hz'). Digital telephony traditionally uses a sample rate of 8000Hz (8kHz) though 16 and even 32kHz are becoming more common. Audio Compact Discs use 44100Hz (44·1kHz), Digital Audio Tape and many computer systems use 48kHz and professional audio systems often use 96kHz.

sample size
: The number of bits used to store each sample. Today, 16-bit is commonly used, 8-bit was popular in the early days of computer audio and 24-bit is used in the professional audio arena.

data encoding
: The way in which each audio sample is represented (or 'encoded'). Some encodings have variants with different byte-orderings or bit-orderings, some compress the audio data so that it takes up less disk space or transmission bandwidth than uncompressed formats. Commonly-used encoding types include floating point, μ-law, ADPCM, signed-integer PCM, MP3 and FLAC.

channels
: The number of audio channels contained in the file. One ('mono') and two ('stereo') are widely used and 'surround sound' audio typically contains six or more channels.

The term 'bit rate' is a measure of the amount of storage occupied by an encoded audio signal per unit of time. It can depend on all of the above and is typically denoted as a number of kilobits per second (kbps). An A-law telephony signal has a bit rate of 64 kbps, MP3-encoded stereo music typically has a bit rate of

128−196 kbps and FLAC-encoded stereo music typically has a bit rate of 550−760 kbps.

Most self-describing formats also allow textual 'comments' to be embedded in the file that can be used to describe the audio in some way, e.g. for music, the title, the author, etc.

One important use of audio file comments is to convey 'Replay Gain' information. SoX can apply Replay Gain automatically for formats that contain comments but does not generate it. By default, SoX copies comments from the first input file to output files that support comments, so output files may contain Replay Gain information which is incorrect. This can be fixed, when converting input files with **−−replay−gain** enabled, by removing all comments using **−−comment ""** or removing just the REPLAYGAIN comment with

```
soxi_ng -a in.au | grep -v REPLAYGAIN > comments
sox_ng --replay-gain=track in.au --comment-file comments out.au
```

Determining and setting the File Format

SoX uses several mechanisms to determine or set the format of an audio file. Depending on the circumstances, individual characteristics may be determined or set using different mechanisms.

To determine the format of an input file, SoX uses, in order of precedence and as given or available:

1. Command-line format options,
2. The contents of the file header,
3. The filename extension.

To set the output file format, SoX uses, in order of precedence and as given or available:

1. Command-line format options,
2. The filename extension,
3. The input file format characteristics or the closest that is supported by the output file type.

For all files, SoX exits with an error if the file type cannot be determined. Command-line format options may need to be added or changed to resolve the problem.

Playing & Recording Audio

The **play_ng** and **rec_ng** commands are provided so that basic playing and recording is as simple as

```
play_ng existing-file.wav
```

and

```
rec_ng new-file.wav
```

These two commands are functionally equivalent to

```
sox_ng existing-file.wav -d
```

and

```
sox_ng -d new-file.wav
```

Further options and effects (as described below) can be added to the commands in either form.

* * *

When playing a file with a sample rate that is not supported by the audio output device, SoX automatically invokes the **rate** effect to perform the necessary sample rate conversion. For compatibility with old hardware, the default **rate** quality level is set to 'low'. This can be changed by explicitly specifying the **rate** effect with a different quality level, e.g.

```
play_ng ... rate -m
```

or by using the **−−play−rate−arg** option (see below).

* * *

On some systems, SoX allows the audio playback volume to be adjusted while using **play_ng**. Where supported, this is achieved by tapping the '**v**' & '**V**' keys during playback. If there is a **softvol** effect in the chain, these keys will adjust that instead of the hardware mixer.

To help with setting a suitable recording level, SoX includes a peak level meter which can be invoked (before making the actual recording) as follows:

```
rec_ng -n
```

The recording level should be adjusted (using the system-provided mixer program, not SoX) so that the meter is, at most, occasionally full scale and never 'in the red' (an exclamation mark is shown). See the **−S** (**−−show−progress**) option below.

Accuracy

Many file formats that compress audio discard some of the audio signal information. Converting to such a format and back again will not produce an exact copy of the original audio. This is the case for many formats used in telephony (e.g. A-law, GSM) where low signal bandwidth is more important than high audio fidelity and for formats used in portable music players (e.g. MP3, Ogg Vorbis) where adequate fidelity can be retained even with the large compression ratios that are needed to make portable players practical.

Formats that discard audio signal information are called 'lossy'. Formats that do not are called 'lossless'. The term 'quality' is used as a measure of how closely the original audio signal is reproduced when using a lossy format.

Audio file conversion with SoX is lossless when it can be, i.e. when not using lossy compression, when not reducing the sampling rate or number of channels and when the number of bits used in the destination format is not less than in the source format. For example, converting from an 8-bit PCM format to a 16-bit PCM format is lossless but converting from an 8-bit PCM format to (8-bit) A-law isn't.

SoX converts all audio files to an internal, uncompressed 32-bit format before performing any audio processing. This means that manipulating a file that is stored in a lossy format can cause further losses in audio fidelity. E.g. with

```
sox_ng long.mp3 short.mp3 trim 10
```

SoX first decompresses the input MP3 file, then applies the **trim** effect and finally creates the output MP3 file by recompressing the audio with a possible reduction in fidelity above that which occurred when the input file was created. Hence, if what is ultimately desired is lossily compressed audio, it is best to perform all audio processing using lossless file formats and then convert to the lossy format only at the final stage. Applying multiple effects with a single SoX invocation will, in general, produce more accurate results than those produced using multiple SoX invocations.

Dithering

Dithering is a technique used to maximize the dynamic range of audio stored at a particular bit-depth. Any distortion introduced by quantization is decorrelated by adding a small amount of white noise to the signal. In most cases, SoX can determine whether the selected processing requires dither and will add it during output formatting if appropriate.

By default, SoX automatically adds TPDF dither when the output bit-depth is less than 24 and any of the following are true:

- bit-depth reduction has been specified explicitly using a command-line option
- the output file format supports only bit-depths lower than that of the input file format
- an effect has increased the effective bit-depth within the internal processing chain

For example, adjusting the volume with **vol 0.25** requires two additional bits in which to losslessly store its results (since 0·25 decimal equals 0·01 binary) so, if the input file bit-depth is 16, SoX's internal representation will use 18 bits after processing this volume change. In order to store the output at the same depth as the input, dithering is used to remove the additional bits.

Use the **−V** option to see what processing SoX has automatically added. The **−D** (**−−no−dither**) option may be given to override automatic dithering. To invoke dithering manually (e.g. to select a noise-shaping curve) use the **dither** effect.

Clipping

Clipping is distortion that occurs when an audio signal level (or 'volume') exceeds the range of the chosen representation. In most cases, clipping is undesirable and so should be corrected by adjusting the level prior to the point in the processing chain at which it occurs.

In SoX, clipping can happen when using the **vol** or **gain** effects to increase the audio volume. Clipping can also occur with many other effects, when converting one format to another and even when simply playing the audio.

Playing an audio file often involves resampling and processing by analog components and that can introduce a DC offset or amplification, all of which can produce distortion if the audio signal level was initially too close to the clipping point.

For these reasons, it is usual to make sure that an audio file's signal level has some 'headroom', i.e. it does not exceed a particular level below the maximum possible level of the given representation. Some standards bodies recommend as much as 9dB headroom, but in most cases, 3dB (≈ 70% linear) is enough. Note that this wisdom seems to have been lost in modern music production; in fact, many CDs, MP3s, etc. are now mastered at levels *above* 0dBFS and the audio is clipped as delivered.

SoX's **stat** and **stats** effects can assist in determining the signal level of an audio file. The **gain** or **vol** effect can be used to prevent clipping, e.g.

```
sox_ng dull.wav bright.wav gain −6 treble +6
```

guarantees that the treble boost will not clip.

If clipping occurs at any point during processing, SoX displays a warning message to that effect.

See the global **−G** (**−−guard**) option and the **gain** and **norm** effects.

Input File Combining

SoX's input combiner can be configured with the **−−combine** global option to combine multiple files using one of the following methods: **concatenate**, **sequence**, **mix**, **mix-power**, **merge** and **multiply**, with shorthands **−m** for **−−combine mix**, **−M** for **merge** and **−T** for **multiply**.

The default is **sequence** for **play_ng**, and **concatenate** for **sox_ng**.

For methods other than **sequence**, multiple input files must have the same sampling rate. If necessary, separate SoX invocations can be used to make sampling rate adjustments prior to combining them. and, with **concatenate**, the input files must also have the same number of channels.

The **sequence** combining method is similar to **concatenate** in that the audio from each input file is sent serially to the output file but here, the output file may be closed and reopened at the transition between input files. This may be just what is needed when sending different types of audio to an output device but is not generally useful when the output is a normal file.

With the **mix** or **mix-power** combining methods, the number of channels in each input file need not be the same but SoX issues a warning if they are not and some channels in the output file will not contain audio from every input file. A mixed audio file cannot be unmixed without reference to the original input files.

If the **merge** combining method is selected the number of channels in each input file need not be the same and a merged audio file comprises all channels from all the input files and unmerging is possible using multiple invocations of SoX with the **remix** effect. For example, two mono files could be merged to form one stereo file and the first and second mono files would become the left and right channels of the stereo file.

The **multiply** combining method multiplies the sample values of corresponding channels treated as numbers in the interval −1 to +1. If the number of channels in the input files is not the same, the missing channels will contain silence.

When combining input files, SoX applies any specified effects (including, for example, the **vol** volume adjustment effect) after the audio has been combined. However, it is often useful to be able to set the volume of the inputs individually (i.e. 'balance' them) before combining takes place. For all combining methods, input file volume adjustments can be made manually using the **−v** option, which can be given for one or more input files. If it is given for only some of the input files, the others receive no volume adjustment. In some circumstances, automatic volume adjustments may be applied. The global **−V** option can be used to show the input file volume adjustments that have been selected manually or automatically.

Some special considerations need to made when mixing input files:

Unlike the other methods, **mix** combining can cause clipping in the combiner if no balancing is performed. In this case, if manual volume adjustments are not given, SoX tries to ensure that clipping does not occur by automatically adjusting the volume (amplitude) of each input signal by a factor of $1/n$, where n is the number of input files. If this results in audio that is too quiet or otherwise unbalanced, the input file volumes can be set manually with **−v**. Using the **norm** effect on the mix is another alternative.

If mixed audio seems loud enough at some points but too quiet in others, dynamic range compression can be applied to correct this—see the **compand** effect.

With the **mix-power** combine method, the mixed volume is approximately equal to that of one of the input signals. This is achieved by balancing using a factor of $1/\sqrt{n}$ instead of $1/n$. Note that this balancing factor does not guarantee that clipping will not occur but the number of clips will usually be low and the resulting distortion is usually imperceptible.

Output Files

SoX's default behaviour is to take one or more input files and write them to a single output file.

This behaviour can be changed by placing the **newfile** pseudo-effect within the effects list and SoX will enter multiple output mode.

In multiple output mode, a new file is created when the effects prior to the **newfile** indicate that they are done. The effects chain listed after **newfile** is then started up and its output is saved to the new file.

In multiple output mode, a unique number is appended automatically to all filenames and, if the filename has an extension, the number is inserted before the extension. This behaviour can be customized by placing **%n** in the filename where the number should be substituted. An optional number can be placed after the **%** to indicate a minimum width for the number with leading zeroes. **%n** defaults to two digits or, if no **%n** is included, to three digits before the filename extension.

Multiple output mode is not very useful unless an effect that stops the effects chain is specified before **newfile**. If the end of the file is reached before the effects chain stops, no new file is created as it would be empty.

The following is an example of splitting the first 60 seconds of an input file into two 30 second files and ignoring the rest.

```
sox_ng song.wav ringtone%1n.wav trim 0 30 : newfile : trim 0 30
```

Stopping SoX

Usually, SoX completes its processing and exits automatically once it has read all audio data from the input files.

It can also be terminated earlier by sending it an interrupt signal, usually by pressing the keyboard interrupt key which is normally Ctrl-C. This is required It is needed when using SoX to make a recording.

When SoX is playing multiple files, Ctrl-C behaves slightly differently: pressing it once skips to the next file; pressing it twice in quick succession causes SoX to exit. However, when playing multiple files decoded with **ffmpeg**, this will kill all the pending file reads, resulting in a second or so of each – what each process had already written to its pipe.

Another way to stop processing early is to use an effect that has a time period or sample count; the **trim** effect is an example of this. Once all effects chains have stopped, SoX stops.

FILENAMES

Filenames can be simple file names, relative or absolute path names, URLs (for input files only) or special filenames. URL support requires one of the **wget**, **wget2** or **curl** programs to be installed.

Giving SoX an input or output filename that is the same as the name of a SoX effect does not work since SoX will treat it as an effect specification. You can work around this by calling the file **./chorus** on Unix or **.\chorus** on MS/DOS but it is not usually a problem since most audio filenames have a filename extension after a dot, which effect names do not.

Using the same file name as an input and an output is unlikely to work as intended because it is likely to truncate the file before reading all of it.

Special Filenames

The following filenames may be used in certain circumstances in place of a normal filename:

– SoX can be used in simple pipeline operations by using the special filename '–' which, if used as an input filename, causes SoX to read audio data from the 'standard input' (stdin) and, if used as the output filename, will cause SoX to send audio data to the 'standard output' (stdout). When using this option for the output file, and sometimes when using it for an input file, the file type (see **–t** below) must also be given.

"|*program* [*options*] ...**"**

An initial 'pipe' character specifies that the given command's standard output (stdout) should be used as an input file. Unlike the special filename **–**, this can be used for several inputs to one SoX command. For example, if a program `genw` generates a mono WAV signal on its standard output, the following command makes a stereo file from two generated signals:

```
sox_ng -M "|genw --imd -" "|genw --thd -" out.wav
```

For headerless (raw) audio and some other formats, **–t** needs to be given before the input command.

"*wildcard-filename***"**

Specifies that filename 'globbing' (wildcard matching) should be performed by SoX instead of by the shell if the *wildcard-filename* contains the characters *****, **?** or characters ranges such as **[A-Z]**. This allows a single set of file options to be applied to a group of files. For example, if the current directory contains three files, `file1.vox`, `file2.vox` and `file3.vox`,

```
play_ng --rate 6k *.vox
```

is expanded by the shell (in most environments) to

```
play_ng --rate 6k file1.vox file2.vox file3.vox
```

which only treats the first 'vox' file as having a sample rate of 6k. With

```
play_ng --rate 6k "*.vox"
```

the given sample rate option is applied to all the files.

If you do not want SoX to glob the filenames, you can use the option **––no–glob** before each filename that should not be globbed, which is necessary if you need to process files whose names contain wildcard characters.

–p, **––sox–pipe**

This can be used in place of an output filename to specify that its output will be used as the input to another SoX command. For example, in the command:

```
play_ng "|sox_ng -n -p synth 2" "|sox_ng -n -p synth 2 tremolo 10"
```

play_ng thinks it's playing two files in succession that come from pipes, but in fact they both come from other invocations of SoX, each with different effects.

–p is in fact an alias for '**–t sox –**'.

−d, **−−default−device**
This can be used in place of an input or output filename to specify that the default audio device (if one has been built into SoX) is to be used. This is akin to invoking **rec_ng** or **play_ng** as described above.

−n, **−−null**
This can be used in place of an input or output filename to specify that a 'null file' is to be used. Here, 'null file' refers to a SoX-specific mechanism and is not related to any operating system mechanism with some special name.

Using a null file as an input is equivalent to using a normal audio file that contains an infinite amount of silence and, as such, is not generally useful unless used with an effect that specifies a finite time length such as **trim** or **synth**.

Using a null file as an output amounts to discarding the audio and is mainly useful with effects that produce information about the audio instead of affecting it such as **noiseprof**, **stat** and **spectrogram**.

The sampling rate associated with a null file is by default 48kHz but, as with a normal file, this can be overridden using format options such as **−r** (see below).

Supported File and Audio Device Types

See **soxformat_ng**(7) for a list and description of the supported file formats and audio device drivers.

OPTIONS

Global Options

These options can be specified on the command line at any point before the first effect name.

The **SOX_OPTS** environment variable can be used to provide alternative default values for SoX's global options. See **ENVIRONMENT** (below).

−A *freq*
Set the frequency of note A4 to *freq* instead of 440Hz.

−−buffer *bytes*, **−−input−buffer** *bytes*
Set the size in bytes of the buffers used for processing audio (default 8192). **−−buffer** applies to input, effects and output processing; **−−input−buffer** applies only to input processing, for which it overrides **−−buffer** if both are given.

Large values for **−−buffer** may cause SoX to be become slow to respond to requests to terminate or to skip to the next input file.

−−clobber
Don't prompt before overwriting an existing file that has the same name as an output file. This is the default behaviour; to override it, use **−−no−clobber**.

−−combine concatenate|merge|mix|mix−power|multiply|sequence
Select the input file combining method. See **Input File Combining** above for a description of them.

−D, **−−no−dither**
Disable automatic dither—see **Dithering** above. This may be useful to ensure that SoX produces the same output in successive runs or if a file has been converted from 16 to 24 bit with the intention of doing some processing on it, but in fact no processing is needed after all and the original 16 bit file has been lost, in which case no dither is needed when converting the file back to 16 bits. See the **stats** effect for how to determine the actual bit-depth of the audio within a file.

−−effects−file *filename*
Read a file to obtain all effects and their arguments. The file is parsed as if the values were specified on the command line. A new line can be used in place of the special **:** marker to separate effect chains. For convenience, such markers at the end of the file are normally ignored; if you want to specify an empty last effects chain, use an explicit **:** by itself on the last line of the file. This

option causes any effects specified on the command line to be discarded.

–G, **––guard**
Automatically invoke the **gain** effect to guard against clipping. E.g.

```
sox_ng -G in.au -b 16 out.au rate 44100 dither -s
```

is shorthand for

```
sox_ng in.au -b 16 out.au gain -h rate 44100 gain -rh dither -s
```

See **–V, ––norm,** and the **gain** effect.

–h, **––help**
Show SoX's version number and usage information.

––help–effect *name*
Show usage information for the specified effect. The name **all** can be used to show it for all available effects.

––help–format *name*
Show information about the specified file format. The name **all** can be used to show information for all supported formats.

––i, **––info**
If given as the first parameter to **sox_ng**, behave as **soxi_ng**.

––interactive
Enable control of SoX by keyboard strokes, making

v and **V**
Lower and raise the master volume control, or that of **softvol** if it is in the effects chain.

< and **>** Seek back and forth in the audio file by 30 seconds.

n Skip to the next track.

q or **Esc**
Quit.

R Restart the effects chain (like **:**)

All the above can be overridden by a **––keymap** option.

Interactive mode is enabled by default when playing, disabled otherwise. To disable it when playing, redirect stdin from somewhere other than a terminal, e.g. **play ... < file**

–k|**––keymap** *key*:*effect*.*field*[+|−|*|/|=]*value*
Pressing the specified key will change an effect's parameter, adding or subtracting *value* for linear changes, multiplying or dividing by *value* for logarithmic control or setting the parameter to that value. For example:

```
play_ng -V -k D:dolbyb.gain+2 -k d:dolbyb.gain-2 in.wav dolbyb
```

lets you hear the results of Dolby B removal and adjust the Threshold Gain in steps of 2dB to find the best setting.

––keymap turns **––interactive** mode on and **–V** makes it display the new value each time it is changed.

When you have the same effect more than once, you can control just with something like **effect2.field** to affect only the second occurrence of *effect* in the effects chain and for effects with multiple stages, like **echo.delay**, the stage can be selected with something like **effect.field1**.

The manual entries for individual effects say which parameters can be adjusted in this way, **––help–effect tells you which of an effect's parameters can be keymapped and**

```
sox_ng --help-effect all | grep Keymap
```

gives a complete list; note that hyphenated parameters like **gain-out** have a keymap of **gain_out**.

−m|−M
Equivalent to **−−combine mix** and **−−combine merge** respectively.

−−magic
If SoX has been built with the optional 'libmagic' library, this option enables its use in helping to detect audio file types.

−−multi−threaded|−−single−threaded
By default, SoX is 'single threaded' but if the **−−multi−threaded** option is given, SoX processes audio channels for most multichannel effects in parallel on hyperthreading and multicore processors.

A larger buffer size than the default may be needed to benefit more from multithreaded processing (e.g. 131072; see **−−buffer** above) and setting `OMP_WAIT_POLICY=PASSIVE` in the environment before launching SoX avoids a defect in OpenMP that can make it inredibly slow if several invocations of SoX are running simultaneously.

−−no−clobber
Prompt before overwriting an existing file with the same name as that given for the output file.

Unintentionally overwriting a file is easier than you might think, for example, if you accidentally enter

```
sox_ng file1 file2 effect1 effect2 ...
```

when what you really meant was

```
play_ng file1 file2 effect1 effect2 ...
```

then, without this option, file2 will be overwritten. Hence, using this option is recommended and can be set in the **SOX_OPTS** environment variable (see **ENVIRONMENT** below).

−−norm[**=***dB-level*]
Automatically invoke the **gain** effect to guard against clipping and to normalize the audio. E.g.

```
sox_ng --norm in.au -b 16 out.au rate 44100 dither -s
```

is shorthand for

```
sox_ng in.au -b 16 out.au gain -h rate 44100 gain -nh dither -s
```

Optionally, the audio can be normalized to a given level, usually below 0 dBFS:

```
sox_ng --norm=-3 in.au out.au
```

See **−V**, **−G** and the **gain** effect.

−−play−rate−arg *arg*
Selects a quality option to be used when the **rate** effect is invoked automatically when playing audio. This option is typically set via the **SOX_OPTS** environment variable (see **ENVIRONMENT** below) and its default value, when playing, is **−l** (low quality but fast). See the **rate** effect for other alternatives.

−−plot gnuplot|octave|off
If not set to **off** (the default if **−−plot** is not given), run in a mode that can be used in conjunction with the **gnuplot** program or the GNU Octave program to assist with the selection and configuration of many of the transfer function-based effects. For the first given effect that supports the selected plotting program, SoX outputs commands to plot the effect's transfer function and stops without actually processing any audio. E.g.

```
sox_ng --plot octave input-file -n highpass 1320 > highpass.plt
octave highpass.plt
```

−q, **−−no−show−progress**
Run in quiet mode when SoX wouldn't otherwise do so. This is the opposite of the **−S** option. To suppress error and warning messages, see **−V** below.

−R Run in 'repeatable' mode. When this option is given, SoX embeds a time stamp in the output file if its format supports comments and will seed pseudo random number generators, as used by **dither**, with that number, ensuring that successive SoX invocations with the same inputs and the same parameters yield the same output.

−−replay−gain track|album|off
Select whether or not to apply replay gain adjustment to input files. The default is **off** for **sox_ng** and **rec_ng**, **album** for **play_ng** when (at least) the first two input files are tagged with the same Artist and Album names and **track** for **play_ng** otherwise.

−S, **−−show−progress**
Display input file format/header information, processing progress as a percentage of the input file(s), elapsed time, remaining time (if known, in brackets) and the number of samples written to the output file. Also shown is a peak level meter, and an indication of whether clipping has occurred. The peak level meter shows up to two channels and is calibrated for digital audio as follows:

dB FSD	*Display*	*dB FSD*	Display
−25	−	−11	====
−23	=	−9	====−
−21	=−	−7	=====
−19	==	−5	=====−
−17	==−	−3	======
−15	===	−1	=====!
−13	===−		

A three-second peak-held value of the headroom in dBs is shown to the right of the meter if the headroom is less than 6dB.

This option is enabled by default when using SoX to play or record audio but can be disabled with **−q**.

−T Equivalent to **−−combine multiply**

−−temp *directory*
Specify that any temporary files should be created in the given directory. This can be useful if there are permission or free space problems with the default location. In this case, using '**−−temp .**' (to use the current directory) is often a good solution.

−−version
Show SoX's version number and exit.

−V[*level*]
Set verbosity. This is particularly useful for seeing how any automatic effects have been invoked by SoX.

SoX displays messages on the console (stderr) according to the following verbosity levels:

0 No messages are shown at all; use the exit status to determine if an error has occurred.

1 Only error messages are shown. These are generated if SoX cannot complete the requested commands.

2 Warning messages are also shown. These are generated if SoX can complete the requested commands but not exactly according to the requested command parameters, or if clipping occurs. This is the default.

3 Descriptions of SoX's processing phases are also shown, to see exactly how SoX is processing your audio.

4 to 6 Messages to help with debugging SoX are also shown.

Each occurrence of the **−V** option increases the verbosity level by 1. Alternatively, the verbosity level can be set to an absolute number by specifying it immediately after the **−V**, e.g. **−V0** shuts it up.

Input File Options

These options apply to the first input filename that follows them on the command line.

−−ignore−length

Override the audio length given in an audio file's header. If this option is given, SoX keeps reading audio until it reaches the end of the input file.

−v, **−−volume** *factor*

Intended for use when combining multiple input files, this option adjusts the volume of the file that follows it on the command line by a factor of *factor*. This allows it to be 'balanced' w.r.t. the other input files. This is a linear (amplitude) adjustment, so a number less than 1 decreases the volume and a number greater than 1 increases it. If a negative number is given then, in addition to the volume adjustment, the audio signal will be inverted.

See the **norm**, **vol** and **gain** effects, **Input File Balancing** above and the **Special Filenames**' section on *wildcard-filenames*.

Input & Output File Format Options

These options apply to the input or output file whose name they immediately precede on the command line and are used mainly when working with headerless file formats or when specifying a format for the output file that is different from that of the input file.

−b *bits*, **−−bits** *bits*

Set the number of bits (a.k.a. bit-depth or word length) in each encoded sample. It is not applicable to complex encodings such as MP3 or GSM and not necessary with encodings that have a fixed number of bits such as A-law, μ-law and ADPCM.

For an input file, the most common use for this option is to inform SoX of the number of bits per sample in a 'raw' ('headerless') audio file. For example

```
sox_ng −r 16k −e signed −b 8 input.raw output.wav
```

converts a particular 'raw' file to a self-describing 'WAV' file.

For an output file, this option can be used to set the output encoding size. By default, the output encoding size is be set to the input encoding size, provided it is supported by the output file type. For example:

```
sox_ng input.cdda −b 24 output.wav
```

converts raw CD digital audio (16-bit, signed-integer) to a 24-bit (signed-integer) 'WAV' file.

−c *CHANNELS*, **−−channels** *CHANNELS*

Sets the number of audio channels in the audio file. This can be any number greater than zero.

For an input file, the most common use for this option is to inform SoX of the number of channels in a 'raw' (headerless) audio file. Occasionally, it may be useful to use this option with a headered file to override the (presumably incorrect) value in the header but this is only supported with certain file types. For example:

```
sox_ng −r 48k −e float −b 32 −c 2 input.raw output.wav
```

converts a 'raw' file to a self-describing 'WAV' file.

```
play_ng −c 1 music.wav
```

interprets the file data as belonging to a single channel regardless of what is indicated in the file header and if the file does in fact have two channels, it is played at half speed.

For an output file, this option provides a shorthand for specifying that the **channels** effect should be invoked in order to change (if necessary) the number of channels in the audio signal to the number given. For example, the following two commands are equivalent:

```
sox_ng input.wav -c 1 output.wav bass -b 24
sox_ng input.wav      output.wav bass -b 24 channels 1
```

though the second form is more flexible as it allows effects to be ordered arbitrarily.

−e *ENCODING*, **−−encoding** *ENCODING*
Set the audio encoding type, sometimes needed with file types that support more than one encoding scheme such as raw, WAV or AU but not with MP3 or FLAC. The available encoding types are as follows:

signed−integer
PCM data stored as signed ('two's complement') integers. Commonly used with a 16 or 24 −bit encoding size. A value of 0 represents minimum signal power.

unsigned−integer
PCM data stored as unsigned integers. Commonly used with an 8-bit encoding size. A value of 0 represents maximum signal power.

floating−point
PCM data stored as IEEE 753 single precision (32-bit) or double precision (64-bit) floating point ('real') numbers. A value of 0 represents minimum signal power.

a−law International telephony standard for logarithmic encoding to 8 bits per sample. It has a precision equivalent to roughly 13-bit PCM and is sometimes encoded with reversed bit-ordering (see the **−X** option).

u−law/mu−law
The North American telephony standard for logarithmic encoding to 8 bits per sample, a.k.a. μ-law has a precision equivalent to roughly 14-bit PCM and is sometimes encoded with reversed bit-ordering (see the **−X** option).

oki−adpcm
OKI (a.k.a. VOX, Dialogic, or Intel) 4-bit ADPCM has a precision equivalent to roughly 12-bit PCM. ADPCM is a form of audio compression that makes a good compromise between audio quality and encoding/decoding speed.

ima−adpcm
IMA (a.k.a. DVI) 4-bit ADPCM has a precision equivalent to roughly 13-bit PCM.

ms-adpcm
Microsoft 4-bit ADPCM has a precision equivalent to roughly 14-bit PCM.

gsm−full−rate
GSM is currently used for the majority of the world's digital wireless telephone calls. It utilizes several audio formats with different bit rates and associated speech quality. SoX has support for GSM's original 13kbps 'Full Rate' audio format.

Encoding names can be abbreviated where this would not be ambiguous; e.g. **unsigned−integer** can be given as **un**, but not **u** (ambiguous with **u-law**).

For an input file, the most common use for this option is to inform SoX of the encoding of a 'raw' ('headerless') audio file (see the examples in **−b** and **−c** above).

For an output file, this option can be used (perhaps with **−b**) to set the output encoding. For example:

```
sox_ng input.cdda -e float output1.wav
```

```
sox_ng input.cdda -b 64 -e float output2.wav
```

converts a raw CD digital audio (16-bit signed integer) to floating point 'WAV' files of single and double precision respectively.

If this option is not given, the output encoding will be the same as the input encoding, provided it is supported by the output file type.

--no-glob
Specifies that filename 'globbing' (wildcard matching) should not be performed by SoX on the following filename. For example, if the current directory contains the two files 'five-seconds.wav' and 'five*.wav', then

```
play_ng --no-glob "five*.wav"
```

can be used to play just the single file 'five*.wav'.

-r, --rate *rate*[**k**]
Gives the sample rate in Hz (or kHz if followed by **k**) of the file.

For an input file, the most common use for this option is to inform SoX of the sample rate of a 'raw' ('headerless') audio file (see the examples in **-b** and **-c** above). Occasionally it may be useful to use this option with a 'headered' file to override the value in the header, though this is only supported with certain file types. For example, if audio was recorded with a sample rate of 48k from a source that played back a little too slowly, say 1·5%,

```
sox_ng -r 48720 input.wav output.wav
```

would correct the speed by changing only the file header (but see the **speed** effect for the more usual solution to this problem).

For an output file, this option provides a shorthand for specifying that the **rate** effect should be invoked in order to change (if necessary) the sample rate of the audio signal to the given value. For example, the following two commands are equivalent:

```
sox_ng input.wav -r 48k output.wav bass -b 24
sox_ng input.wav        output.wav bass -b 24 rate 48k
```

though the second form is more flexible as it allows **rate** options to be given and allows the effects to be ordered arbitrarily.

-t, **--type** *FILE-TYPE*
Give the type of an audio file. For both input and output files, this option is commonly used to inform SoX of the type a 'headerless' audio file where the actual/desired type cannot be determined from the filename extension. For example:

```
another-command | sox_ng -t mp3 - output.wav
sox_ng input.wav -t raw output.bin
```

It can also be used to override the type implied by an input filename extension but, if overriding with a type that has a header, SoX exit with an error message if such a header is not actually present.

There are also pseudo filetypes that tell SoX to use a specified format module that handles more than one type of audio file such as **-t sndfile** and **-t ffmpeg** or when a type of file can be handled by several different format modules, such as WAV files containing MP3-encoded data.

Furthermore, the file type can be used to select a particular audio device driver for recording and playing.

See **soxformat_ng**(7) for a list of supported file types.

-L, **--endian little**
-B, **--endian big**

−x, **−−endian swap**
These options specify whether the byte order of the audio data is, respectively, 'little endian', 'big endian' or the opposite to that of the system on which SoX is being used. Endianness applies only to data encoded as floating point or as signed or unsigned integers of 16 or more bits. It is often necessary to specify one of these options for headerless files and sometimes necessary for (otherwise) self-describing files. A given endian-setting option may be ignored for an input file whose header contains a specific endianness identifier or for an output file that is actually an audio device.

N.B. Unlike other format characteristics, the endianness (byte, nibble, & bit ordering) of the input file is not automatically used for the output file. For example, when the following is run on a little-endian system:

```
sox_ng −B audio.s16 trimmed.s16 trim 2
```

`trimmed.s16` will be created as little-endian;

```
sox_ng −B audio.s16 −B trimmed.s16 trim 2
```

must be used to preserve big-endianness in the output file.

The **−V** option can be used to check the selected orderings.

−N, **−−reverse−nibbles**
Specifies that the nibble ordering of the samples (i.e. the 2 halves of a byte) should be reversed, which is sometimes useful with ADPCM-based formats.

See the N.B. in the section on **−x** above.

−X, **−−reverse−bits**
Specifies that the bit ordering of the samples should be reversed, which is sometimes useful with a few (mostly headerless) formats.

See the N.B. in the section on **−x** above.

Output File Format Options
These options only apply to output files and may only precede an output filename on the command line.

−−add−comment *TEXT*
Append a comment to the output file header (where applicable).

−−comment *TEXT*
Specify the comment text to store in the output file header (where applicable).

SoX provides a default comment 'Processed by SoX' if this option (or **−−comment−file**) is not given. To specify that no comment should be stored in the output file, use **−−comment "" .**

−−comment−file *FILENAME*
Specify a file containing the comment text to store in the output file header (where applicable).

−C, **−−compression** *FACTOR*
Set the compression factor for variably-compressed output file formats. If this option is not given then a default compression factor applies. The compression factor is interpreted differently for different compressed file formats; for details, see the description of the file formats that use this option in **soxformat_ng**(7)

EFFECTS

In addition to converting, playing and recording audio files, SoX can be used to invoke a number of audio effects. Multiple effects may be applied by specifying them one after the other at the end of the SoX command line, forming an 'effects chain'. For a list of the supports effects, see **soxeffect_ng**(7)

ENVIRONMENT

SoX reacts to certain environment variables. To set them on Unix with most shells, use, for example:

```
AUDIODRIVER=alsa
```

```
export AUDIODRIVER
play_ng ...
```

to select the lowest-level driver on Linux, which may work better than the default of pulseaudio, or with Unix csh:

```
setenv AUDIODRIVER alsa
```

or, on Microsoft Windows:

```
set AUDIODRIVER=waveaudio
```

MS-Windows GUI: via Control Panel : System : Advanced : Environment Variables

Mac OS X GUI: Refer to Apple's Technical Q&A QA1067 document.

AUDIODRIVER

On some systems, SoX may have more than one type of audio driver, e.g. **alsa** and **oss** or **SUNAU** and **ao** and they can have more than one audio device (a.k.a. 'sound card'). If more than one audio driver has been built into SoX and the default selected by SoX when recording or playing is not the one that is wanted, the **AUDIODRIVER** environment variable can be used to override it; available drivers are listed near the end of the help text given by **sox** with no arguments.

If it is unset, SoX tries to use, in order, **coreaudio**, **pulseaudio**, **alsa**, **waveaudio**, **sndio**, **oss**, **sunau** and **ao**. For further details on these, see their entries in **soxformat_ng**(7).

AUDIODEV

Override the default audio device, e.g.

```
AUDIODEV=/dev/dsp2
export AUDIODEV
play_ng ...
sox_ng ... -t oss
```

or

```
AUDIODEV=hw:soundwave,1,2
export AUDIODEV
play_ng ...
sox_ng ... -t alsa
```

If AUDIODEV is unset and the audio driver is **oss**, SoX also responds to the standard environment variable **OSS_AUDIODEV**.

LADSPA_PATH

A colon-separated list of directories in which to search for LADSPA plugins. The default depends on how SoX was built but on Unix it defaults to `/usr/lib/ladspa`, on MacOS/X to `/Library/Audio/Plug-Ins/LADSPA`. Windows doesn't have a "usual place" for LADSPA plugins, but Ardour puts them in `C:\Program Files\Ardour6\lib\ardour6\ladspa` or similar.

LD_LIBRARY_PATH

When searching for the dynamic libraries in which most effects and format handlers may be stored, according to how your SoX was built, look in this colon-separated list of directories before the default location.

MIXERDEV

When playing a file, use the specified mixer device for the 'v' and 'V' volume control keys.

SOX_OPTS

Provide alternative default values for SoX's global options. For example:

```
SOX_OPTS="--buffer 20000 --play-rate-arg -hs --temp /mnt/temp"
export SOX_OPTS
```

Note that setting **SOX_OPTS** can create unwanted changes in the behaviour of scripts or other programs that invoke SoX. **SOX_OPTS** might best be used for things that reflect the environment in which SoX is being run and enabling options such as **−−no−clobber** by default might be handled better using a shell alias since that will not affect SoX's operation in scripts or when it is used by other programs.

One way to ensure that scripts and programs cannot be affected by **SOX_OPTS** is to clear **SOX_OPTS** at the start of the script, but this loses the benefit of **SOX_OPTS** carrying system-wide defaults.

TEMP and **TMP**

On Windows, tmpfile() is broken — it creates the file in the root directory of the current drive instead of in a valid temporary directory — but if **TEMP** or **TMP** are set, it creates them in the directory indicated, otherwise in the current directory. To force use of tmpfile(), use **--temp** .

Alternatively, and on Unix, use **--temp** (see **Global Options** above).

EXIT STATUS

SoX exits 0 when there is no error, 1 if there is a problem with the command-line parameters or 2 if an error occurs during audio processing.

BUGS

Please report any bugs found in this version of SoX to the mailing list <sox-ng@groups.io>.

CITATION

To cite SoX in publications please use:

```
Lance Norskog, Chris Bagwell et al. (2015).
SoX: Sound eXchange, the Swiss Army knife of audio manipulation.
URL http://sox.sourceforge.net
```

A BibTeX entry for SoX users is

```
@manual{SoX2015,
  title = "SoX: Sound eXchange, the Swiss Army knife of audio manipulation",
  author = "Norskog, Lance and Bagwell, Chris and others",
  edition = "14.4.2",
  year = 2015,
  url = "http://sox.sourceforge.net",
}
```

SEE ALSO

soxi_ng(1), **soxeffect_ng**(7), **soxformat_ng**(7), **libsox_ng**(3), **audacity**(1), **ecasound**(1), **ffmpeg**(1), **gnuplot**(1), **octave**(1), **sndfile-convert**(1).
The **sox_ng** web site at `https://codeberg.org/sox_ng/sox_ng`
SoX scripting examples under `scripts/` in the SoX source code.

LICENSE

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License version 2 as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

AUTHORS

Lance Norskog, Chris Bagwell and many others listed in the AUTHORS file that is distributed with the source code.