

## NAME

SoX – Sound eXchange, the Swiss Army knife of audio manipulation

## SYNOPSIS

```
sox [global-options] [format-options] infile1  
  [[format-options] infile2] ... [format-options] outfile  
  [effect [effect-options]] ...
```

```
play [global-options] [format-options] infile1  
  [[format-options] infile2] ... [format-options]  
  [effect [effect-options]] ...
```

```
rec [global-options] [format-options] outfile  
  [effect [effect-options]] ...
```

## DESCRIPTION

SoX reads and writes audio files in most popular formats and can optionally apply effects to them; it can combine multiple input sources, synthesise audio, and, on many systems, act as a general purpose audio player or a multi-track audio recorder.

The entire SoX functionality is available using just the ‘sox’ command, however, to simplify playing and recording audio, if SoX is invoked as ‘play’, the output file is automatically set to be the default sound device and if invoked as ‘rec’, the default sound device is used as an input source.

The heart of SoX is a library called libSoX. Those interested in extending SoX or using it in other programs should refer to the libSoX manual page: **libsox(3)**.

The overall SoX processing chain can be summarised as follows:

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

To show how this works in practise, here are some examples of how SoX might be used. The simple:

```
sox recital.au recital.wav
```

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

```
sox recital.au -r 12000 -1 -c 1 recital.wav vol 0.7 dither
```

performs the same format translation, but also changes the audio sampling rate & sample size, down-mixes to mono, and applies the **vol** and **dither** effects.

```
sox -r 8000 -u -1 -c 1 voice-memo.raw voice-memo.wav
```

adds a header to a raw audio file,

```
sox slow.aiff fixed.aiff speed 1.027 rabbit -c0
```

adjusts audio speed using the most accurate **rabbit** algorithm,

```
sox short.au long.au longer.au
```

concatenates two audio files, and

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

mixes together two audio files.

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

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

```
play -n -c1 synth sin %-12 sin %-9 sin %-5 sin %-2 fade q 0.1 1 0.1
```

plays a synthesised ‘A minor seventh’ chord with a pipe-organ sound,

```
rec -c 2 test.aiff trim 0 10
```

records 10 seconds of stereo audio, and

```
rec -M take1.aiff take1-dub.aiff
```

records a new track in a multi-track recording.

Further examples are included throughout this manual; more-detailed examples can be found in **sox-exam(7)**.

## File Formats

There are two types of audio file format that SoX can work with. The first is ‘self-describing’; these formats include a header that completely describes the characteristics of the audio data that follows. The second type is ‘headerless’ (or ‘raw data’); here, the audio data characteristics must be described using the SoX command line.

The following four characteristics are sufficient to describe the format of audio data such that it can be processed with SoX:

sample rate

The sample rate in samples per second (‘Hertz’ or ‘Hz’). For example, digital telephony traditionally uses a sample rate of 8000 Hz (8 kHz); audio Compact Discs use 44100 Hz (44.1 kHz).

sample size

The number of bits used to store each sample. Most popular are 8-bit (one byte) and 16-bit (two bytes). (Since many now-common sound formats were invented when most computers used a 16-bit word, two bytes is often called a ‘word’, but since current personal computers overwhelmingly have 32-bit or 64-bit words, this usage is confusing, and is not used in the SoX documentation.)

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, i.e. the stored audio data takes up less space (i.e. disk-space or transmission band-width) than the other format parameters and the number of samples would imply. Commonly-used encoding types include floating-point,  $\mu$ -law, ADPCM, signed linear, and FLAC.

channels

The number of audio channels contained in the file. One (‘mono’) and two (‘stereo’) are widely used.

The term ‘bit-rate’ is sometimes used as an overall measure of an audio format and may incorporate elements of all of the above.

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 supports applying Replay Gain information, but not generating it. Note that by default, SoX copies input file comments to output files that support comments, so output files may contain Replay Gain information if some was present in the input file. In this case, if anything other than a simple format conversion was performed then the output file Replay Gain information is likely to be incorrect and so should be recalculated using a tool that supports this (not SoX).

## Determining & Setting The File Format

There are several mechanisms available for SoX to use to determine or set the format characteristics 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 will use, 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 will use, 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 to them that is supported by the output file type.

For all files, SoX will exit 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.

### Accuracy

Many file formats that compress audio discard some of the audio signal information whilst doing so; converting to such a format then converting 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 many formats used in portable music players (e.g. MP3, 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’, and formats that do not, ‘lossless’. The term ‘quality’ is used as a measure of how closely the original audio signal can be 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. E.g. 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.

**N.B.** SoX converts all audio files to an internal uncompressed 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 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 highly recommended to perform all audio processing using lossless file formats and then convert to the lossy format at the final stage.

**N.B.** Applying multiple effects with a single SoX invocation will, in general, produce more accurate results than those produced using multiple SoX invocations; hence this is also recommended.

### Clipping

Clipping is distortion that occurs when an audio signal level (or ‘volume’) exceeds the range of the chosen representation. It is nearly always undesirable and so should usually be corrected by adjusting the volume prior to the point at which clipping occurs.

In SoX, clipping could occur, as you might expect, when using the **vol** effect to increase the audio volume, but could 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 re-sampling, and processing by analogue components that can introduce a small DC offset and/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 does not exceed around 70% of the maximum (linear) range available, as this will avoid the majority of clipping problems. SoX’s **stat** effect can assist in determining the signal level in an audio file; the **vol** effect can be used to prevent clipping, e.g.

```
sox dull.au bright.au vol -6 dB treble +6
```

guarantees that the treble boost will not clip.

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

### Input File Combining

SoX's input combiner can combine multiple files using one of four different methods: 'concatenate', 'sequence', 'mix', or 'merge'. The default method is 'sequence' for **play**, and 'concatenate' for **rec** and **sox**.

For all 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.

If the 'concatenate' combining method is selected (usually, this will be by default) then the input files must also have the same number of channels. The audio from each input will be concatenated in the order given to form the output file.

The 'sequence' combining method is selected automatically for **play**. It is similar to 'concatenate' in that the audio from each input file is sent serially to the output file, however here the output file may be closed and reopened at the corresponding transition between input files—this may be just what is needed when sending audio to an output device, but is not generally useful when the output file is a normal file.

If the 'mix' combining method is selected (with **-m**) then two or more input files must be given and will be mixed together to form the output file. The number of channels in each input file need not be the same, however, SoX will issue 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 un-mixed.

If the 'merge' combining method is selected (with **-M**), then two or more input files must be given and will be merged together to form the output file. The number of channels in each input file need not be the same. A merged audio file comprises all of the channels from all of the input files; un-merging is possible using multiple invocations of SoX with the **mixer** effect. For example, two mono files could be merged to form one stereo file; the first and second mono files would become the left and right channels of the stereo file.

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 (i.e. 'balance') the inputs individually, before combining takes place.

For all combining methods, input file volume adjustments can be made manually using the **-v** option (below) which can be given for one or more input files; if it is given for only some of the input files then the others receive no volume adjustment. In some circumstances, automatic volume adjustments may be applied (see below).

The **-V** option (below) can be used to show the input file volume adjustments that have been selected (either manually or automatically).

There are some special considerations that need to be made when mixing input files:

Unlike the other methods, 'mix' combining has the potential to cause clipping in the combiner if no balancing is performed. So here, if manual volume adjustments are not given, to ensure that clipping does not occur, SoX will automatically adjust 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 then the input file volumes should be set manually as described above.

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

### Stopping SoX

Usually SoX will complete its processing and exit automatically, however if desired, it can be terminated by pressing the keyboard interrupt key (usually Ctrl-C). This is a natural requirement in some circumstances, e.g. when using SoX to make a recording. Note that when using SoX to play multiple files, Ctrl-C behaves slightly differently: pressing it once causes SoX to skip to the next file; pressing it twice in quick succession

causes SoX to exit.

## FILENAMES

Filenames can be simple file names, absolute or relative path names, or URLs (input files only). Note that URL support requires that **wget(1)** is available.

Note: Giving SoX an input or output filename that is the same as a SoX effect-name will not work since SoX will treat it as an effect specification. The only work-around to this is to avoid such filenames; however, this is generally not difficult since most audio filenames have a filename 'extension', whilst effect-names do not.

The following 'special' filenames may be used in certain circumstances in place of a normal filename on the command line:

– SoX can be used in pipeline operations by using the special filename '-' which, if used in place of an input filename, will cause SoX will read audio data from 'standard input' (stdin), and which, if used in place of the output filename, will cause SoX will send audio data to 'standard output' (stdout). Note that when using this option, the file-type (see **-t** below) must also be given.

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

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

The sampling rate associated with a null file is by default 44.1 kHz, but, as with a normal file, this can be overridden if desired using command-line format options (see below).

One other use of **-n** is to use it in conjunction with **-V** to display information from the audio file header without having to read any further into the file, e.g.

```
sox -V *.wav -n
```

will display header information for each 'WAV' file in the current directory.

**-e** This is an alias of **-n** and is retained for backwards compatibility only.

## OPTIONS

### Global Options

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

**-h, --help**

Show version number and usage information.

**--help-effect=NAME**

Show usage information on the specified effect. The name **all** can be used to show usage on all effects.

**--interactive**

Prompt before overwriting an existing file with the same name as that given for the output file.

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

```
sox file1 file2 effect1 effect2 ...
```

when what you really meant was

```
play file1 file2 effect1 effect2 ...
```

then, without this option, file2 will be overwritten. Hence, using this option is strongly

recommended; a ‘shell’ alias, script, or batch file may be an appropriate way of permanently enabling it.

**--buffer BYTES**

Set the size in bytes of the buffers used for reading and writing sound data (default 8192).

**-m|-M|--combine concatenate|merge|mix|sequence**

Select the input file combining method; **-m** selects ‘mix’, **-M** selects ‘merge’,

See **Input File Combining** above for a description of the different combining methods.

**--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 will output commands to plot the effect’s transfer function, and then exit without actually processing any audio. E.g.

```
sox --plot octave input-file -n highpass 1320 > plot.m
octave plot.m
```

**-q, --no-show-progress**

Run in quiet mode when SoX wouldn’t otherwise do so; this is the opposite of the **-S** option.

**--replay-gain track|album|off**

Select whether or not to apply replay-gain adjustment to input files. The default is **track** for **play** and **off** otherwise.

**-S, --show-progress**

Display input file format/header information and input file(s) processing progress in terms of elapsed/remaining time and percentage complete. This option is enabled by default when using SoX to play or record audio.

**--version**

Show version number and exit.

**-V[level]**

Set verbosity. SoX prints messages to the console (stderr) according to the following verbosity levels:

- 0 No messages are printed at all; use the exit status to determine if an error has occurred.
- 1 Only error messages are printed. These are generated if SoX cannot complete the requested commands.
- 2 Warning messages are also printed. These are generated if SoX can complete the requested commands, but not exactly according to the requested command parameters, or if clipping occurs.
- 3 Descriptions of SoX’s processing phases are also printed. Useful for seeing exactly how SoX is mangling your audio.
- 4 and above Messages to help with debugging SoX are also printed.

By default, the verbosity level is set to 2. 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** sets it to 0.

## Input File Options

These options apply only to input files and may precede only input filenames on the command line.

**-v, --volume** *FACTOR*

Adjust volume by a factor of *FACTOR*. This is a linear (amplitude) adjustment, so a number less than 1 decreases the volume; 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 also the **stat** effect for information on how to find the maximum volume of an audio file; this can be used to help select suitable values for this option.

See also **Input File Balancing** above.

### 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 to that of the input file.

**-c, --channels** *CHANNELS*

The number of audio channels in the audio file. This may be 1, 2, or 4; for mono, stereo, or quad audio. To cause the output file to have a different number of channels than the input file, include this option with the output file options. If the input and output file have a different number of channels then the **mixer** effect must be used. If the **mixer** effect is not specified on the command line it will be invoked internally with default parameters.

**--comment** *TEXT*

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

SoX will provide a default comment 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).

**-r, --rate** *RATE*

Gives the sample rate in Hz of the file. To cause the output file to have a different sample rate than the input file, include this option with the output file format options.

If the input and output files have different rates then a sample rate change effect must be run. Since SoX has multiple rate changing effects, the user can specify which to use as an effect. If no rate change effect is specified then a default one will be chosen.

**-t, --type** *file-type*

Gives the type of the audio file. This is useful when the file extension is non-standard or when the type can not be determined by looking at the header of the file.

The **-t** option 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 will exit with an appropriate error message if such a header is not actually present.

See **soxformat(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 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 normal format characteristics, the endianness (byte, nibble, & bit ordering) of the input file is not automatically used for the output file; so, for example, when the following is run on a little-endian system:

```
sox -B audio.uw trimmed.uw trim 2
```

trimmed.uw will be created as little-endian;

```
sox -B audio.uw -B trimmed.uw 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 (i.e. the 2 halves of a byte) of the samples should be reversed; sometimes useful with ADPCM-based formats.

**N.B.** See also N.B. in section on **-x** above.

**-X, --reverse-bits**

Specifies that the bit ordering of the samples should be reversed; sometimes useful with a few (mostly headerless) formats.

**N.B.** See also N.B. in section on **-x** above.

**-s/-u/-U/-A/-a/-i/-g/-f**

The audio data encoding is signed linear (2's complement), unsigned linear,  $\mu$ -law (logarithmic), A-law (logarithmic), ADPCM, IMA-ADPCM, GSM, or floating-point.

$\mu$ -law (or mu-law) and A-law are the U.S. and international standards for logarithmic telephone audio compression. When uncompressed  $\mu$ -law has roughly the precision of 14-bit PCM audio and A-law has roughly the precision of 13-bit PCM audio.

A-law and  $\mu$ -law are sometimes encoded using reversed bit-ordering (i.e. MSB becomes LSB). If you need this support then you can use the **-X** option or the pseudo file types of `.la` and `.lu` to inform SoX of the encoding. See supported file types for more information.

ADPCM is a form of audio compression that has a good compromise between good audio quality and fast encoding/decoding time. It is used for telephone audio compression and places where full fidelity is not as important. When uncompressed it has roughly the precision of 16-bit PCM audio. Popular versions of ADPCM include G.726, MS ADPCM, and IMA ADPCM. The **-a** flag has different meanings in different file handlers. In `.wav` files it represents MS ADPCM files, in all others it means G.726 ADPCM. IMA ADPCM is a specific form of ADPCM compression, slightly simpler and slightly lower fidelity than Microsoft's flavor of ADPCM. IMA ADPCM is also called DVI ADPCM.

GSM is currently used for the vast majority of the world's digital wireless telephone calls. It utilises several audio formats with different bit-rates and associated speech quality. SoX has support for GSM's original 13kbps 'Full Rate' audio format. It is usually CPU intensive to work with GSM audio.

**-1/-2/-3/-4/-8**

The sample datum size is 1, 2, 3, 4, or 8 bytes; i.e. 8, 16, 24, 32, or 64 bits.

**Output File Format Options**

These options apply only to the output file and may precede only the output filename on the command line.

**-C, --compression FACTOR**

The compression factor for variably compressing output file formats. If this option is not given, then a default compression factor will apply. The compression factor is interpreted differently for different compressing file formats. See the description of the file formats that use this option in `soxformat(7)` for more information.

**DIAGNOSTICS**

Exit status is 0 for no error, 1 if there is a problem with the command-line parameters, or 2 if an error occurs during file processing.

## **BUGS**

Please report any bugs found in this version of SoX to the mailing list ([sox-users@lists.sourceforge.net](mailto:sox-users@lists.sourceforge.net)).

## **SEE ALSO**

**soxexam(7)**, **soxformat(7)**, **soxeffect(7)**, **gnuplot(1)**, **octave(1)**, **wget(1)**, **libsox(3)**

The SoX web site at <http://sox.sourceforge.net>

## **LICENSE**

Copyright 1991 Lance Norskog and Sundry Contributors. Copyright 1998–2007 by Chris Bagwell and SoX Contributors.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2, or (at your option) any later version.

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**

Chris Bagwell ([cbagwell@users.sourceforge.net](mailto:cbagwell@users.sourceforge.net)). Other authors and contributors are listed in the AUTHORS file that is distributed with the source code.