

## NAME

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

## DESCRIPTION

File types that can be determined by a filename extension are listed with their names preceded by a dot.

File types that require an external library, such as ffmpeg or libsndfile, are marked e.g. '**(ffmpeg)**'. File types that can be handled by an external library via its pseudo file type (currently libsndfile or ffmpeg) are marked e.g. '**(also with -t sndfile)**'. This might be useful if you have a file that doesn't work with SoX's default format readers and writers, and there's an external reader or writer for that format.

### **.raw (also with -t sndfile)**

Raw (headerless) audio files. The sample rate, sample size, and data encoding must be given using command-line format options; the number of channels defaults to 1.

### **.ub, .sb, .uw, .sw, .ul, .al, .lu, .la, .sl (also with -t sndfile)**

These filename extensions serve as shorthand for identifying the format of headerless audio files. Thus, **ub**, **sb**, **uw**, **sw**, **ul**, **al**, **lu**, **la** and **sl** indicate a file with a single audio channel, sample rate of 8000 Hz, and samples encoded as 'unsigned byte', 'signed byte', 'unsigned word', 'signed word', ' $\mu$ -law' (byte), 'A-law' (byte), inverse bit order ' $\mu$ -law', inverse bit order 'A-law', or 'signed long' respectively. Command-line format options can also be given to modify the selected format if it does not provide an exact match for a particular file.

Headerless audio files on a SPARC computer are likely to be of format **ul**; on a Mac, they're likely to be **ub** but with a sample rate of 11025 or 22050 Hz.

### **.8svx (also with -t sndfile)**

Amiga 8SVX musical instrument description format.

### **.aiff, .aif (also with -t sndfile)**

AIFF files used on Apple IIc/IIs and SGI. Note: the AIFF format supports only one SSND chunk. It does not support multiple audio chunks, or the 8SVX musical instrument description format. AIFF files are multimedia archives and can have multiple audio and picture chunks. You may need a separate archiver to work with them.

### **.aifc, .aifc (also with -t sndfile)**

AIFF-C (not compressed, linear), defined in DAVIC 1.4 Part 9 Annex B. This format is referred from ARIB STD-B24, which is specified for Japanese data broadcasting. Any private chunks are not supported.

Note: The input file is currently processed as .aiff.

### **alsa** ALSA device driver. This is a pseudo-file type and can be optionally compiled into SoX. Run

```
sox -h
```

to see if you have support for this file type. When this driver is used it allows you to open up a ALSA device and configure it to use the same data format as passed in to SoX. It works for both playing and recording audio files. When playing audio files it attempts to set up the ALSA driver to use the same format as the input file. It is suggested to always override the output values to use the highest quality format your ALSA system can handle. Example:

```
sox infile -t alsa default
```

### **.amr-nb**

Adaptive Multi Rate—Narrow Band speech codec; a lossy format used in 3rd generation mobile telephony and defined in 3GPP TS 26.071 et al.

AMR-NB audio has a fixed sampling rate of 8 kHz and supports encoding to the following bit-rates (as selected by the **-C** option): 0 = 4.75 kbit/s, 1 = 5.15 kbit/s, 2 = 5.9 kbit/s, 3 = 6.7 kbit/s, 4 = 7.4 kbit/s, 5 = 7.95 kbit/s, 6 = 10.2 kbit/s, 7 = 12.2 kbit/s.

This format in SoX is optional and requires access to external libraries. To see if there is support for this format, enter

```
sox -h
```

and look for it under the list: *SUPPORTED FILE FORMATS*.

#### **.amr-wb**

Adaptive Multi Rate—Wide Band speech codec; a lossy format used in 3rd generation mobile telephony and defined in 3GPP TS 26.171 et al.

AMR-WB audio has a fixed sampling rate of 16 kHz and supports encoding to the following bit-rates (as selected by the **-C** option): 0 = 6.6 kbit/s, 1 = 8.85 kbit/s, 2 = 12.65 kbit/s, 3 = 14.25 kbit/s, 4 = 15.85 kbit/s, 5 = 18.25 kbit/s, 6 = 19.85 kbit/s, 7 = 23.05 kbit/s, 8 = 23.85 kbit/s.

This format in SoX is optional and requires access to external libraries. To see if there is support for this format on your system, enter

```
sox -h
```

and look for it under the list: *SUPPORTED FILE FORMATS*.

**ao** libao device driver. This is a pseudo-file type and can be optionally compiled into SoX. Run

```
sox -h
```

to see if you have support for this file type. It works only for playing audio files. It can play to a wide range of devices and sound systems. See its documentation for the full range. For the most part, SoX's use of libao cannot be configured directly; you must use libao configuration files.

The filename specified is used to determine which libao plugin to use. Normally, you should specify "default" as the filename. If that doesn't give the desired behavior then you can specify the short name for a given plugin (such as pulse for pulse audio plugin).

#### **.au, .snd (also with -t sndfile)**

Sun Microsystems AU files. There are many types of AU file; DEC has invented its own with a different magic number and byte order. SoX can read these files but will not write them. Some .au files are known to have invalid AU headers; these are probably original Sun  $\mu$ -law 8000 Hz files and can be dealt with using the **.ul** format (see below).

It is possible to override AU file header information with the **-r** and **-c** options, in which case SoX will issue a warning to that effect.

**auto** This format type name exists for backwards compatibility only. If given for an input file it will be silently ignored, if given for an output file it will cause SoX to exit with an error.

**.avr** Audio Visual Research. The AVR format is produced by a number of commercial packages on the Mac.

#### **.caf (libsndfile)**

Core Audio File format.

#### **.cdda, .cdr**

'Red Book' Compact Disc Digital Audio. CDDA has two audio channels formatted as 16-bit signed integers at a sample rate of 44.1 kHz. The number of (stereo) samples in each CDDA track is always a multiple of 588 which is why it needs its own handler.

#### **.cvsd, .cvs**

Continuously Variable Slope Delta modulation. A headerless format used to compress speech audio for applications such as voice mail. This format is sometimes used with bit-reversed samples—the **-X** format option can be used to set the bit-order.

**.dat** Text Data files. These files contain a textual representation of the sample data. There is one line at the beginning that contains the sample rate. Subsequent lines contain two numeric data items: the time since the beginning of the first sample and the sample value. Values are normalized so that

the maximum and minimum are 1 and -1. This file format can be used to create data files for external programs such as FFT analysers or graph routines. SoX can also convert a file in this format back into one of the other file formats.

**.dvms, .vms**

Used in Germany to compress speech audio for voice mail. A self-describing variant of **cvsd**.

**.fap (libsndfile)**

See **.paf**.

**ffmpeg** This is a pseudo-type that forces ffmpeg to be used. The actual file type is deduced from the file name (it cannot be used on stdio). This pseudo-type depends on SoX having been built with optional ffmpeg support. It can read a wide range of audio files, not all of which are documented here, and also the audio track of many video files (including AVI, WMV and MPEG). At present only the first audio track of a file can be read.

**.flac (also with -t sndfile)**

Free Lossless Audio CODEC compressed audio. FLAC is an open, patent-free CODEC designed for compressing music. It is similar to MP3 and Ogg Vorbis, but lossless, meaning that audio is compressed in FLAC without any loss in quality.

SoX can decode native FLAC files (.flac) but not Ogg FLAC files (.ogg). [But see **.ogg** below for information relating to support for Ogg Vorbis files.]

SoX has basic support for writing FLAC files: it can encode to native FLAC using compression levels 0 to 8. 8 is the default compression level and gives the best (but slowest) compression; 0 gives the least (but fastest) compression. The compression level can be selected using the **-C** option (see above) with a whole number from 0 to 8.

FLAC support in SoX is optional and requires optional FLAC libraries. To see if there is support for FLAC run

```
sox -h
```

and look for it under the list of supported file formats as 'flac'.

**.fssd** An alias for the **.ub** format.

**.gsm (also with -t sndfile)**

GSM 06.10 Lossy Speech Compression. A lossy format for compressing speech which is used in the Global Standard for Mobile telecommunications (GSM). It's good for its purpose, shrinking audio data size, but it will introduce lots of noise when a given audio signal is encoded and decoded multiple times. This format is used by some voice mail applications. It is rather CPU intensive.

GSM in SoX is optional and requires access to an external GSM library. To see if there is support for GSM run

```
sox -h
```

and look for it under the list of supported file formats.

**.hcom** Macintosh HCOM files. These are (apparently) Mac FSSD files with some variant of Huffman compression. The Macintosh has wacky file formats and this format handler apparently doesn't handle all the ones it should. Mac users will need their usual arsenal of file converters to deal with an HCOM file on other systems.

**ircam (also with -t sndfile)**

Another name for **.sf**.

**.ima (also with -t sndfile)**

A headerless file of IMA ADPCM audio data. IMA ADPCM claims 16-bit precision packed into only 4 bits, but in fact sounds no better than **.vox**.

### **.lpc, .lpc10**

LPC-10 is a compression scheme for speech developed in the United States. See <http://www.arl.wustl.edu/~jaf/lpc/> for details. There is no associated file format, so SoX's implementation is headerless.

### **.mat, .mat4, .mat5 (libsndfile)**

Matlab 4.2/5.0 (respectively GNU Octave 2.0/2.1) format (.mat is the same as .mat4).

### **.m3u**

A *playlist* format; contains a list of audio files. See [1] for details of this format.

### **.maud**

An IFF-conforming audio file type, registered by MS MacroSystem Computer GmbH, published along with the 'Toccata' sound-card on the Amiga. Allows 8bit linear, 16bit linear, A-Law,  $\mu$ -law in mono and stereo.

### **.mp3, .mp2**

MP3 compressed audio. MP3 (MPEG Layer 3) is part of the MPEG standards for audio and video compression. It is a lossy compression format that achieves good compression rates with little quality loss. See also **Ogg Vorbis** for a similar format.

MP3 support in SoX is optional and requires access to either or both the external libmad and libmp3lame libraries. To see if there is support for MP3 run

```
sox -h
```

and look for it under the list of supported file formats as 'mp3'.

### **.mp4, .m4a (ffmpeg)**

MP4 compressed audio. MP4 (MPEG 4) is part of the MPEG standards for audio and video compression. See **mp3** for more information.

MP4 support in SoX is optional and requires access to the external ffmpeg libraries.

### **.nist (also with -t sndfile)**

See **.sph**.

### **.ogg, .vorbis**

Ogg Vorbis compressed audio. Ogg Vorbis is a open, patent-free CODEC designed for compressing music and streaming audio. It is a lossy compression format (similar to MP3, VQF & AAC) that achieves good compression rates with a minimum amount of quality loss. See also **MP3** for a similar format.

SoX can decode all types of Ogg Vorbis files, and can encode at different compression levels/qualities given as a number from -1 (highest compression/lowest quality) to 10 (lowest compression, highest quality). By default the encoding quality level is 3 (which gives an encoded rate of approx. 112kbps), but this can be changed using the **-C** option (see above) with a number from -1 to 10; fractional numbers (e.g. 3.6) are also allowed.

Decoding is somewhat CPU intensive and encoding is very CPU intensive.

Ogg Vorbis in SoX is optional and requires access to external Ogg Vorbis libraries. To see if there is support for Ogg Vorbis run

```
sox -h
```

and look for it under the list of supported file formats as 'vorbis'.

### **oss**

OSS /dev/dsp device driver. This is a pseudo-file that can be optionally compiled into SoX. Run

```
sox -h
```

to see if it is supported. When this driver is used it allows you to play and record sounds on supported systems. When playing audio files it attempts to set up the OSS driver to use the same format as the input file. It is suggested to always override the output values to use the highest quality format your OSS system can handle. Example:

```
sox infile -t oss -2 -s /dev/dsp
```

**.paf, .fap (libsndfile)**

Ensoniq PARIS file format (big and little-endian respectively).

**.pls** A *playlist* format; contains a list of audio files. See [2] for details of this format.

Note: SHOUTcast PLS relies on **wget(1)** and is only partially supported: it's necessary to specify the audio type manually, e.g.

```
play -t mp3 "http://a.server/pls?rn=265&file=filename.pls"
```

and SoX does not know about alternative servers—hit Ctrl-C twice in quick succession to quit.

**.prc** Psion Record. Used in Psion EPOC PDAs (Series 5, Revo and similar) for System alarms and recordings made by the built-in Record application. When writing, SoX defaults to A-law, which is recommended; if you must use ADPCM, then use the **-i** switch. The sound quality is poor because Psion Record seems to insist on frames of 800 samples or fewer, so that the ADPCM CODEC has to be reset at every 800 frames, which causes the sound to glitch every tenth of a second.

**.pvf (libsndfile)**

Portable Voice Format.

**.sd2 (libsndfile)**

Sound Designer 2 format.

**.sds (libsndfile)**

MIDI Sample Dump Standard.

**.sf (also with -t sndfile)**

IRCAM SDIF (Institut de Recherche et Coordination Acoustique/Musique Sound Description Interchange Format). Used by academic music software such as the CSound package, and the MixView sound sample editor.

**.sph, .nist (also with -t sndfile)**

SPHERE (SPeech HEader Resources) is a file format defined by NIST (National Institute of Standards and Technology) and is used with speech audio. SoX can read these files when they contain  $\mu$ -law and PCM data. It will ignore any header information that says the data is compressed using *shorten* compression and will treat the data as either  $\mu$ -law or PCM. This will allow SoX and the command line *shorten* program to be run together using pipes to encompass the data and then pass the result to SoX for processing.

**.smp** Turtle Beach SampleVision files. SMP files are for use with the PC-DOS package SampleVision by Turtle Beach Softworks. This package is for communication to several MIDI samplers. All sample rates are supported by the package, although not all are supported by the samplers themselves. Currently loop points are ignored.

**.snd** See **.au**.

**sndfile** This is a pseudo-type that forces libsndfile to be used. For writing files, the actual file type is then taken from the output file name; for reading them, it is deduced from the file. This pseudo-type depends on SoX having been built with optional libsndfile support.

**.sndt** Sndtool files. This format dates from the MS-DOS era. Bizarrely, this file type can also be used to read Sounder files.

**.sou** An alias for the **.ub** format.

**sunau** Sun /dev/audio device driver. This is a pseudo-file type and can be optionally compiled into SoX. Run

```
sox -h
```

to see if you have support for this file type. When this driver is used it allows you to open up a

Sun /dev/audio file and configure it to use the same data type as passed in to SoX. It works for both playing and recording audio files. When playing audio files it attempts to set up the audio driver to use the same format as the input file. It is suggested to always override the output values to use the highest quality format your hardware can handle. Example:

```
sox infile -t sunau -2 -s /dev/audio
```

or

```
sox infile -t sunau -U -c 1 /dev/audio
```

for older sun equipment.

**.txw** Yamaha TX-16W sampler. A file format from a Yamaha sampling keyboard which wrote IBM-PC format 3.5" floppies. Handles reading of files which do not have the sample rate field set to one of the expected by looking at some other bytes in the attack/loop length fields, and defaulting to 33 kHz if the sample rate is still unknown.

**.vms** See **.dvms**.

**.voc (also with -t sndfile)**

Sound Blaster VOC files. VOC files are multi-part and contain silence parts, looping, and different sample rates for different chunks. On input, the silence parts are filled out, loops are rejected, and sample data with a new sample rate is rejected. Silence with a different sample rate is generated appropriately. On output, silence is not detected, nor are impossible sample rates. Note, this version now supports playing VOC files with multiple blocks and supports playing files containing  $\mu$ -law and A-law samples.

**.vorbis** See **.ogg**.

**.vox (also with -t sndfile)**

A headerless file of Dialogic/OKI ADPCM audio data commonly comes with the extension **.vox**. This ADPCM data has 12-bit precision packed into only 4-bits.

Note: some early Dialogic hardware does not always reset the ADPCM encoder at the start of each vox file. This can result in clipping and/or DC offset problems when it comes to decoding the audio. Whilst little can be done about the clipping, a DC offset can be removed by passing the decoded audio through a high-pass filter, e.g.:

```
sox input.vox output.au highpass 10
```

**.w64 (libsndfile)**

Sonic Foundry's 64-bit RIFF/WAV format.

**.wav (also with -t sndfile)**

Microsoft **.WAV** RIFF files. This is the native audio file format of Windows, and widely used for uncompressed audio.

Normally **.wav** files have all formatting information in their headers, and so do not need any format options specified for an input file. If any are, they will override the file header, and you will be warned to this effect. You had better know what you are doing! Output format options will cause a format conversion, and the **.wav** will be written appropriately.

SoX currently can read PCM,  $\mu$ -law, A-law, MS ADPCM, and IMA (or DVI) ADPCM. It can write all of these formats including the ADPCM encoding. Big endian versions of RIFF files, called RIFX, can also be read and written. To write a RIFX file, use the **-B** option with the output file options.

**.wavpcm**

A non-standard variant of **.wav**. Some applications cannot read a standard WAV file header for PCM-encoded data with sample-size greater than 16-bits, but can read a non-standard WAV header. It is likely that such applications will eventually be updated to support the standard header, but in the mean time, this SoX format can be used to create files with the non-standard

header that should work with these applications. (Note that SoX will automatically detect and read WAV files with the non-standard header.)

**.wve** Psion 8-bit A-law. Used on Psion SIBO PDAs (Series 3 and similar).

**.xa** Maxis XA files. These are 16-bit ADPCM audio files used by Maxis games. Writing .xa files is currently not supported, although adding write support should not be very difficult.

**.xi (libsndfile)**

Fastracker 2 Extended Instrument format.

## SEE ALSO

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

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

## References

- [1] Wikipedia, *M3U*, <http://en.wikipedia.org/wiki/M3U>
- [2] Wikipedia, *PLS*, [http://en.wikipedia.org/wiki/PLS\\_\(file\\_format\)](http://en.wikipedia.org/wiki/PLS_(file_format))

## AUTHORS

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