NAME

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

DESCRIPTION

This manual describes SoX supported file formats and audio device types; the SoX manual set starts with **sox**(1).

Format types that can SoX can determine by a filename extension are listed with their names preceded by a dot. Format types that are optionally built into SoX are marked '(optional)'.

Format types that can be handled by an external library via an optional pseudo file type (currently **sndfile** 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.

To see if SoX has support for an optional format or device, enter **sox −h** and look for its name under the list: 'AUDIO FILE FORMATS' or 'AUDIO DEVICE DRIVERS'.

SOX FORMATS & DEVICE DRIVERS

.raw (also with **−t sndfile**),
.f4, **.f8**,
.s1, **.s2**, **.s3**, **.s4**,
.u1, **.u2**, **.u3**, **.u4**,
.ul, **.al**, **.lu**, **.la**,
.sb, **.sw**, **.ub**, **.uw**

Raw (headerless) audio files. For **raw**, the sample rate and the data encoding must be given using command-line format options; for the other listed types, the sample rate defaults to 8kHz (but may be overridden), and the data encoding is defined by the given suffix. Thus **f4** and **f8** indicate files encoded as 4 and 8-byte (IEEE single and double precision) floating point PCM respectively; **s1**, **s2**, **s3**, and **s4** indicate 1, 2, 3, and 4-byte signed integer PCM respectively; **u1**, **u2**, **u3**, and **u4** indicate 1, 2, 3, and 4-byte unsigned integer PCM respectively; **ul** indicates 'μ-law' (byte), **al** indicates 'A-law' (byte), and **lu** and **la** are inverse bit order 'μ-law' and inverse bit order 'A-law' respectively. **sb**, **sw**, **ub**, **uw**, and **sl** are aliases for **s1**, **s2**, **u1**, **u2**, and **s4** respectively. For all raw formats, the number of channels defaults to 1 (but may be overridden).

Headerless audio files on a SPARC computer are likely to be of format **ul**; on a Mac, they're likely to be **u1** but with a sample rate of 11025 or 22050 Hz.

See **.ima** and **.vox** for raw ADPCM formats.

.8svx (also with **−t sndfile**)

Amiga 8SVX musical instrument description format.

.aiff, **.aif** (also with **−t sndfile**)

AIFF files used on Apple IIc/IIgs and SGI. Currently, SoX's AIFF support does not include 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.

.aiffc, **.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 (optional)

Advanced Linux Sound Architecture device driver; supports both playing and recording audio. ALSA is only used in Linux-based operating systems, though these often support OSS (see below) as well. Examples:

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

```
sox infile -t alsa hw:0
sox -2 -t alsa hw:1 outfile
```

See also **play**(1) and **rec**(1).

.amb Ambisonic B-Format: a specialisation of **.wav** with between 3 and 16 channels of audio for use with an Ambisonic decoder. See http://www.ambisonia.com/Members/mleese/file-format-for-b-format for details. It is up to the user to get the channels together in the right order and at the correct amplitude.

.amr−nb (optional)

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.

.amr−wb (optional)

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.

ao (optional)

Xiph.org's Audio Output device driver; works only for playing audio. It supports 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; instead, libao configuration files must be used.

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). Examples:

```
sox infile -t ao
sox infile -t ao default
sox infile -t ao pulse
```

See also **play**(1).

.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. To write a DEC file, use the **−L** option with the output file options.

Some .au files are known to have invalid AU headers; these are probably original Sun μ-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.

.avr Audio Visual Research format; used by a number of commercial packages on the Mac.

.caf (optional)

Apple's 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 (optional)
See **.paf**.

ffmpeg (optional)
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). 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 (optional; also with **−t sndfile**)
Xiph.org's 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 read native FLAC files (.flac) but not Ogg FLAC files (.ogg). [But see **.ogg** below for information relating to support for Ogg Vorbis files.]

SoX can write native FLAC files according to a given or default compression level. 8 is the default compression level and gives the best (but slowest) compression; 0 gives the least (but fastest) compression. The compression level is selected using the **−C** option [see **sox**(1)] with a whole number from 0 to 8.

.fssd An alias for the **.u1** format.

.gsm (optional; 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.

.hcom Macintosh HCOM files. These are Mac FSSD files with Huffman compression.

.htk Single channel 16-bit PCM format used by HTK, a toolkit for building Hidden Markov Model speech processing tools.

.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** (optional)
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. SoX can read, but not write this file format. 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, μ-law in mono and stereo.

.mp3, **.mp2** (optional read, optional write)
MP3 compressed audio; MP3 (MPEG Layer 3) is a part of the patent-encumbered MPEG standards for audio and video compression. It is a lossy compression format that achieves good compression rates with little quality loss.

Because MP3 is patented, SoX cannot be distributed with MP3 support without incurring the patent holder's fees. Users who require SoX with MP3 support must currently compile and build SoX with the MP3 libraries (LAME & MAD) from source code.

See also **Ogg Vorbis** for a similar format.

.mp4, **.m4a** (optional)
MP4 compressed audio. MP3 (MPEG 4) is part of the MPEG standards for audio and video compression. See **mp3** for more information.

.nist (also with **−t sndfile**)
See **.sph**.

.ogg, **.vorbis** (optional)
Xiph.org's Ogg Vorbis compressed audio; an open, patent-free CODEC designed for 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.

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.

See also **.mp3** for a similar format.

oss (optional)
Open Sound System /dev/dsp device driver; supports both playing and recording audio. OSS support is available in Unix-like operating systems, sometimes together with alternative sound systems (such as ALSA). Examples:

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

See also **play**(1) and **rec**(1).

.paf, **.fap** (optional)
Ensoniq PARIS file format (big and little-endian respectively).

.pls A *playlist* format; contains a list of audio files. SoX can read, but not write this file format. See [2] for details of this format.

Note: SoX support for 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 (optional)
Portable Voice Format.

.sd2 (optional)
Sound Designer 2 format.

.sds (optional)
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 μ-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 μ-law or PCM. This will allow SoX and the command line *shorten* program to be run together using pipes to encompasses 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**, **.sndr** and **.sndt**.

sndfile (optional)
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.

.sndr Sounder files. An MS-DOS/Windows format from the early ’90s. Sounder files usually have the extension ‘.SND’.

.sndt SoundTool files. An MS-DOS/Windows format from the early ’90s. SoundTool files usually have the extension ‘.SND’.

.sou An alias for the **.u1** raw format.

sunau (optional)
Sun /dev/audio device driver; supports both playing and recording audio. For example:

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

or

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

for older sun equipment.

See also **play**(1) and **rec**(1).

.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. SoX supports reading (but not writing) VOC files with multiple blocks, and files containing μ-law, A-law, and 2/3/4-bit ADPCM 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 (optional)

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 written appropriately.

SoX can read and write PCM, μ-law, A-law, MS ADPCM, and IMA (or DVI) ADPCM. Big endian versions of RIFF files, called RIFX, are also supported. To write a RIFX file, use the **−B** option with the output file options.

.wavpcm

A non-standard, but widely used, variant of **.wav**. Some applications cannot read a standard WAV file header for PCM-encoded data with sample-size greater than 16-bits or with more than two channels, 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.)

The most common use of this file-type is likely to be along the following lines:

```
sox infile.any -t wavpcm -s outfile.wav
```

.wv (optional)

WavPack lossless audio compression. Note that, when converting **.wav** to this format and back again, the RIFF header is not necessarily preserved losslessly (though the audio is).

.wve (also with **−t sndfile**)

Psion 8-bit A-law. Used on Psion SIBO PDAs (Series 3 and similar). This format is deprecated in SoX, but will continue to be used in libsndfile.

.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 (optional)
Fasttracker 2 Extended Instrument format.

SEE ALSO

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

The SoX web page at http://sox.sourceforge.net
SoX scripting examples at http://sox.sourceforge.net/Docs/Scripts

References

[1] Wikipedia, *M3U*, http://en.wikipedia.org/wiki/M3U

[2] Wikipedia, *PLS*, 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.