NAME

soxformat_ng – Audio file formats supported by sox_ng and libsox_ng

DESCRIPTION

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

Format types that 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 are handled by the external library **sndfile** are marked '(with sndfile)' and format types that can only be read using the external program **ffmpeg** are marked '(with ffmpeg)'

Formats for which SoX has internal drivers but that are also supported by **sndfile** or **ffmpeg** are marked (also with **–t sndfile**) or (also with **–t ffmpeg**). This might be useful if you have a file that doesn't work with SoX's built-in readers and writers.

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

To know everything about a format – whether it reads it or writes it or both and what encodings it can write – enter something like **sox_ng –h wav** or, to see details for all formats, **sox_ng––**help–format all**.**

FORMATS & DEVICE DRIVERS

.raw (also with **–t sndfile**), **.f32**, **.f64**, **.s8**, **.s16**, **.s24**, **.s32**, **.u8**, **.u16**, **.u24**, **.u32**, **.ul**, **.al**, **.lu**, **.la**

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 and the data encoding is defined by the given suffix. Thus **f32** and **f64** indicate files encoded as 32 and 64-bit IEEE-754 single and double precision floating point PCM respectively; **s8**, **s16**, **s24** and **s32** indicate 8, 16, 24 and 32-bit signed integer PCM respectively; **u8**, **u16**, **u24** and **u32** indicate 8, 16, 24 and 32-bit unsigned integer PCM respectively; **ul** indicates 'μ-law' (8-bit), **al** indicates 'A-law' (8-bit) and **lu** and **la** are inverse bit-order 'μ-law' and 'A-law' respectively. For all raw formats, the number of channels defaults to 1.

Headerless audio files on a SPARC computer are likely to be of format **ul**; on a Mac, they're likely to be **u8** but with a sample rate of 11025 or 22050Hz.

See **.ima** and **.vox** for raw ADPCM formats and **.cdda** for raw CD digital audio.

.f4, **.f8**, **.s1**, **.s2**, **.s3**, **.s4**, **.u1**, **.u2**, **.u3**, **.u4**, **.sb**, **.sw**, **.sl**, **.ub**, **.uw**

Deprecated aliases for **.f32**, **.f64**, **.s8**, **.s16**, **.s24**, **.s32**, **.u8**, **.u16**, **.u24**, **.u32**, **.s8**, **.s16**, **.s32**, **.u8** and **.u16** respectively.

.3gp, **.3gpp** (with ffmpeg)

Third Generation Partnership Project format.

.3g2, **.3gp2**, **.3gpp2** (with ffmpeg)

Third Generation Partnership Project 2 format.

.8svx (also with **–t sndfile**)

Amiga 8SVX musical instrument description format.

.aa (with ffmpeg)

Audible Audiobook format.

SoX can autodetect this format but, if it's coming from a pipe, **ffmpeg** fails to decode it.

.aac (with ffmpeg)

Advanced Audio Coding format.

.ac3 (with ffmpeg)

Audio Codec 3 (Dolby Digital) format.

.act (with ffmpeg)
G729a compressed speech recorded by some Chinese-made MP3 players.

.adts (with ffmpeg)
Audio Data Transport Stream format.

.aiff, **.aif** (also with **−t sndfile** or **−t ffmpeg**)
AIFF files as used on old Apple Macs, Apple IIc/IIgs and SGI. SoX's AIFF support does not include multiple audio chunks nor 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. On MacOS X, AIFF has been superseded by CAF.

.aiffc, **.aifc** (also with **−t sndfile**)
AIFF-C is based on AIFF but also handles compressed audio. It can also handle little-endian uncompressed linear data that is often referred to as **sowt** encoding. This encoding has become the defacto format produced by modern Macs as well as iTunes on any platform. AIFF-C files produced by other applications typically have the file extension .aif and require looking at its header to detect the true format. **sowt**, **a-law** and **u-law** are the only encodings that SoX can read and write natively; for other compression types like GSM try **−t ffmpeg**.

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

alsa (optional)
The 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_ng infile −t alsa
sox_ng infile −t alsa default
sox_ng infile −t alsa plughw:0,0
sox_ng −b 16 −t alsa hw:1 outfile
```

.amb Ambisonic B-Format is a specialization 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 you to get the channels together in the right order and at the correct amplitude.

.amr−nb, **.amr−wb** (both optional, also with **−t ffmpeg**)
Adaptive Multi Rate Narrow and Wide Band are lossy formats for speech used in 3rd generation mobile telephony and defined in 3GPP TS 26.071 and TS 26.171

AMR-NB audio has a fixed sampling rate of 8kHz and AMR-WB of 16kHz and they support encoding to the following bit rates, selected by the **−C** option:

amr−nb		**amr−wb**	
-C	kbit/s	**-C**	kbit/s
0	4.75	0	6.6
1	5.15	1	8.85
2	5.9	2	12.65
3	6.7	3	14.25
4	7.4	4	15.85
5	7.95	5	18.25
6	10.2	6	19.85
7	12.2	7	23.05
		8	23.85

ao (optional)
Xiph.org's Audio Output device driver only works 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 is used to determine which libao plugin to use and normally, you should specify 'default'. If that doesn't give the desired behavior, you can specify the short name for a given plugin (such as **pulse** for the PulseAudio plugin or **null** for testing. See http://xiph.org/ao). Examples:

```
sox_ng infile -t ao
sox_ng infile -t ao default
sox_ng infile -t ao pulse
```

.ape (with ffmpeg)
Monkey's Audio format.

.apm (with ffmpeg)
Ubisoft Rayman 2 APM format.

.aptx (with ffmpeg)
Audio Processing Technology for Bluetooth format.

SoX can only autodetect this type of file from its filename extension; if it is read from 'standard input' (stdin) or from a file whose name does not end in '.aptx', you will need to prefix it with '**-t ffmpeg**'.

.argo_asf (with ffmpeg)
Argonaut Games ASF format.

.asf (with ffmpeg)
Advanced / Active Streaming Format.

.ast (with ffmpeg)
AST Audio Stream format.

.au, **.snd** (also with **–t sndfile** or **–t ffmpeg**)
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** (little-endian) output file option.

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.

It is possible to override AU file header information with the **–r** (sampling rate) and **–c** (number of channels) options, in which case SoX will issue a warning about the mismatch.

.avi (with ffmpeg)
Audio Video Interleaved format.

.avr (also with **–t ffmpeg** or **–t sndfile**)
Audio Visual Research format, used by a number of commercial packages on the Mac.

.caf (with sndfile, also with **–t ffmpeg**)
Apple's Core Audio File format.

.cdda, **.cda**, **.cdr**
'Red Book' Compact Disc Digital Audio (raw audio). CDDA has two audio channels formatted as 16-bit big-endian 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.

By default, SoX expects and produces CDDA in big-endian format. However, the endianess of CDDA files can vary depending on the tools and hardware used. To produce and to read little-endian CDDA files, predece the filename with the **–L** option.

coreaudio (optional)
The MacOS X CoreAudio device driver supports both playing and recording. If a filename is not specified or is "default", the default audio device is selected. Any other name will be used to select a specific device.

Examples:

```
sox_ng infile -t coreaudio
sox_ng infile -t coreaudio "Internal Speakers"
```

You can also select a device by its number; to see the device numbers:

```
sox_ng -V -n -t coreaudio nonexistent
```

.cvsd, **.cvs**
Continuously Variable Slope Delta modulation is a headerless format used to compress speech audio for applications such as voice mail with a fixed bit rate of 8kHz. This format is sometimes used with bit-reversed samples; the **−X** option can be used to set the bit order.

.cvu Unfiltered Continuously Variable Slope Delta modulation is an alternative handler for CVSD that is unfiltered but can be used with any sampling rate. As it is a headerless format, you have to specify the sampling rate with **−r** if it is different from 8kHz.

```
sox_ng infile outfile.cvu rate 28k
play -r 28k outfile.cvu sinc -3.4k
```

.dat Text Data files contain a textual representation of sample data. There is one line at the beginning that contains the sample rate and one that contains the number of channels. Subsequent lines contain two or more numeric data items: the time since the beginning of the first sample and the sample value for each channel.

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

Example containing only 2 stereo samples of silence:

```
; Sample Rate 8012
; Channels 2
            0     0     0
0.00012481278     0     0
```

.dfpwm (with ffmpeg)
Dynamic Filter Pulse Width Modulation format 1a.

SoX can only autodetect this type of file from its filename extension; if it is read from 'standard input' (stdin) or from a file whose name does not end in '.dfpwm', you will need to prefix it with '**-t ffmpeg**'.

.dts (with ffmpeg)
Digital Theatre Systems format.

SoX can only autodetect this type of file from its filename extension; if it is read from 'standard input' (stdin) or from a file whose name does not end in '.dts', you will need to prefix it with '**-t ffmpeg**'.

.dff Direct Stream Digital Interchange File Format (DSDIFF) is a format defined by Philips for storing 1-bit DSD data, used in SACD mastering and occasionally for online distribution. SoX can write single-, double- and quad-sample rates of 64, 128 and 256 times 44100Hz (2822400, 5644800 and 11289600Hz), known as DSD64, DSD128 and DSD256, but not the octuple- and sexdecuple- variants (DSD512 and DSD1024).

.dsf, **.wsd**
DSD Stream File and Wideband Single-bit Data are formats defined by Sony and by Sharp and Pioneer for storing 1-bit DSD data, commonly used for online distribution of audiophile recordings, supporting the same sample rates as **dff**. The **wsd** variant can be read but not written.

.dvms, **.vms**
The Digital Voice Messaging System format is used in Germany to compress speech audio for voice mail. It is a self-describing variant of **cvsd**.

.ea (with ffmpeg)
Electronic Arts Multimedia format.

.eac3 (with ffmpeg)
Enhanced AC-3 Audio.

.f4v (with ffmpeg)
Another name for **.mov**.

.fap (with sndfile)
See **.paf**.

ffmpeg (optional)
This is a pseudo-type that uses the external program **ffmpeg** if it is installed. It can only read files, not write them, and will extract the sound track from many video file formats. **ffmpeg** deduces the actual file type from the file's contents with a far more advanced algorithm than that used by SoX, which only recognizes up to two fixed byte sequences at fixed offsets.

.flac (optional; also with **−t sndfile** or **−t ffmpeg)**
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, so the audio is compressed without any loss in quality.

SoX can read native FLAC files (.flac) but can only read Ogg FLAC files (**.oga**) if **ffmpeg** is installed.

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 with a whole number from 0 to 8.

.flv (with ffmpeg)
Macromedia Flash Video format.

.fssd Flexible Sound Studio Data format, a raw format that defaults to **.u8** at 8kHz.

.gsrt Grandstream ring-tone files. Whilst this file format can contain A-Law, μ-law, GSM, G.722, G.723, G.726, G.728, or iLBC encoded audio, SoX supports reading and writing only A-Law and μ-law. E.g.

```
sox_ng music.wav −t gsrt ring.bin
play ring.bin
```

.gsm (optional; also with **−t sndfile** or **−t ffmpeg**)
GSM 06.10 Lossy Speech Compression. A lossy format for compressing speech, used in the Global Standard for Mobile telecommunications (GSM) and in some voice mail applications, with a fixed sample rate of 8kHz.

Although SoX can create **gsm** files with up to sixteen channels, it is a headerless format so input is assumed to be mono unless it is told otherwise.

.gxf (with ffmpeg)
General eXchange Format.

.hcom (also with **−t ffmpeg**)
Macintosh HCOM files. These are Mac FSSD files with Huffman compression.

.hip The **−t hip** format forces decoding of MP3 to use **mp3lame**'s decoder based on mpglib by Michael Hipp.

.htk (also with **–t sndfile**)
Single channel 16-bit PCM format used by HTK, a toolkit for building Hidden Markov Model speech processing tools.

.ircam (also with **–t sndfile** or **–t ffmpeg**)
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**.

.ism (with ffmpeg)
ISM streaming video format.

.kvag (with ffmpeg)
Simon & Schuster Interactive VAG format.

.lpc, **.lpc10**
LPC-10 is a compression scheme for speech developed by the United States Department of Defense. See https://github.com/jafingerhut/lpc10 for details. There is no associated file format, so SoX's implementation is headerless.

.m4a (with ffmpeg)
MPEG-4 Audio format.

.m4b (with ffmpeg)
Another name for **.mov**.

.m4v, **.mp4** (with ffmpeg)
MPEG-4 Video format.

.mat, **.mat4**, **.mat5** (with sndfile)
Matlab 4.2/5.0 (respectively GNU Octave 2.0/2.1) format. **.mat** is the same as **.mat4**.

.m3u A playlist format, containing 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 and published along with the 'Toccata' sound card on the Amiga allows 8bit linear, 16bit linear, A-Law and μ-law in mono and stereo.

.mj2 (with ffmpeg)
Another name for **.mov**.

.mkv, **.webm** (with ffmpeg)
Matroska video format.

.mlp (with ffmpeg)
Meridian Lossless Packing format.

.mod Amiga Module format, typically created with Protracker, decoding only.

.mov (with ffmpeg)
MPEG-1 Systems / MPEG program stream format.

.mp1, **.m1a**, **.mp2**, **.mp3** (optional, also with **-t sndfile**, **–t ffmpeg** or **–t hip)**
MP1, MP2 and MP3 compressed audio (MPEG 1 Layers 1, 2 and 3) are part of the MPEG standards for audio and video compression whose patents have expired. They are lossy compression formats that achieve good compression rates with little quality loss.

Decoding

libmad, which SoX uses to decode MPEG files, does not work on files with a bit-rate higher than 192K but you can read those using:

–t sndfile
which uses libmpg123, generally gives better quality results, is better at decoding damaged or corrupt files and is twice as fast as **libmad** if you have an FPU;

–t ffmpeg
which uses yet another decoder, the one in **libavcodec**;

–t hip which uses LAME's internal libmpg but the LAME program is able to decode more MPEG files than **–t hip** as it decodes the file header itself before calling the hip decoder, whereas SoX makes the decoder do all the work.

Yet more ways to decode MPEG files are input filenames like:

```
'|madplay -q --no-tty-control -o wav:- in.mp3'
'|mpg123 -q --no-control --wav - in.mp3'
'|lame --quiet --decode in.mp3 -'
'|ffmpeg -loglevel quiet -nostdin -i in.mp3 -'
```

As these piped commands cannot rewind their output pipe to fix the file length in the header, you may prefer to add **--ignore-length** before them, which makes SoX warn "premature EOF on input file" but keeps reading to the end.

The decoding speeds of the default libmad, **–t hip** and **–t sndfile** are in proportions 1:2:2.5 with an FPU. Without one, libmad will be the fastest instead of the slowest.

With **mad** or **sndfile**, SoX decodes MPEG files with a precision of 28 bits but it declares a precision of 16 bits so that decoding MP3s to WAVs gives the CD quality that most people expect. You can specify a higher precision before the output filename to keep this extra information.

Encoding

SoX can only read MP1 files, not write them.

SoX uses **twolame** to write MP2 files, and the target bit rate is set using the **–C** option in kbps: 32, 48, 56, 64, 80, 96, 112, 128, 160 or 192 for a constant bit rate (CBR) for mono files and double those for stereo ones. Variable bit rate encoding (VBR) is selected by values from −50 to 50 (of which only −10 to 10 are said to be useful); if you really want VBR at qualities 32 or 48, use something like 32.01 or 48.01.

Unfortunately, **twolame** outputs 0.005 seconds of quiet garbage at the start and truncates the end by 0.002 seconds.

It can also encode MP2s with **–t sndfile**, which also uses **twolame** and has the same defect.

SoX uses libmp3lame to write MP3 files and can use up to 25 bits of precision but, at present, SoX's use of libmp3lame adds 0.01 seconds of quiet garbage at the start and at the end; encoding with **–t sndfile** gets the length right and no garbage.

MP3 compression parameters can be selected using SoX's **–C** option as follows:

The primary parameter to the LAME MP3 encoder is the bit rate. If the value of the **–C** value is a positive integer, it's taken as the bitrate in kbps (e.g. if you specify 128, it uses 128 kbps).

The second most important parameter is "quality" which allows balancing encoding speed vs. quality. In LAME, 0 specifies highest quality but is very slow, while 9 selects poor quality, but is fast. (5 is the default and 2 is recommended as a good trade-off for high quality encodes.)

Because the **–C** value is a float, the fractional part is used to select quality. 128.2 selects 128 kbps encoding with a quality of 2. There is one problem with this approach. We need 128 to specify 128 kbps encoding with default quality, so 0 means use default. Instead of 0 you have to use .01 (or .99) to specify the highest or lowest quality (128.01 or 128.99).

LAME uses bitrate to specify a constant bitrate but higher quality can be achieved using Variable Bit Rate (VBR). VBR quality (really size) is selected using a number from 0 to 9. Use a value of 0

for high quality, larger files and 9 for smaller files of lower quality. 4 is the default.

In order to squeeze the selection of VBR into the the **−C** value we use negative numbers to select VBR. -4.2 would select default VBR encoding (size) with high quality (speed). One special case is 0, which is a valid VBR encoding parameter but not a valid bitrate. Compression value of 0 is always treated as a high quality VBR, as a result both -0.2 and 0.2 are treated as highest quality VBR (size) and high quality (speed).

.mp4 (with ffmpeg)
MPEG-4 video format.

.mpeg, **.mpg** (with ffmpeg)
MPEG-1 Systems / MPEG program stream format.

.mpegts (with ffmpeg)
MPEG-TS (MPEG-2 Transport Stream) format.

.mul Tomb Raider Multiplex Stream (write-only support).

.mxf, **.mxf_opatom** (with ffmpeg)
Material eXchange Format Operational Pattern OP1A "OP-Atom" format (SMPTE 390M).

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

.nsp (also with **−t ffmpeg**)
SoX can read Computerized Speech Lab NSP files that may contain both audio and bioelectric data. Typically, the first channel is sound pressure (audio) and additional channels are data such as laryngeal kinematic or aerodynamic (air pressure or air flow).

The NSP file format was also used for the Phonetic Database (PDB) from Speech Technology Research who had a free NSP Player, SpeakNSP. CSL NSP file reading and writing is supported by the WaveSurfer package.

.nut (with ffmpeg)
NUT is a low overhead generic container format that stores audio, video, subtitle and user-defined streams in a simple yet efficient way.

.oga (with ffmpeg)
Various Xiph.org audio formats in an Ogg container.

.ogg, **.vorbis** (optional, also with **−t sndfile** or **−t ffmpeg**))
Xiph.org's Ogg Vorbis compressed audio; an open, patent-free codec designed for music and streaming audio. It is a lossy compression format that achieves good compression rates with a minimal 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 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.

.opus (optional, also with **−t sndfile**)
Xiph.org's Opus compressed audio is an open, lossy, low-latency codec that uses the Ogg container.

It is said to supercede the other free codecs in terms of quality versus bit rate, but can only encode from a signal at 8, 12, 16, 24 or 48 kHz sample rate, resampling other rates before encoding, always decodes at 48kHz. It can handle up to 255 channels, and The channel mappings for up to 8 channels are the same as the Vorbis mappings:

1 monophonic

2 stereo: left, right

3 a 1d-surround encoding: left, center, right

4 quadraphonic surround: front left, front right, rear left, rear right

5 five-channel surround: front left, center, front right, rear left, rear right

6 5.1 surround: front left, center, front right, rear left, rear right, LFE

7 6.1 surround: front left, center, front right, side left, side right, rear center, LFE

8 7.1 surround: front left, center, front right, side left, side right, rear left, rear right, LFE

As with **mp3** encoding, the bitrate can be set with **−C** as a floating-point number where the integer part gives the bitrate in kbps from 6 to 256 kbps per channel, negative values give a variable bit rate (VBR) and positive ones a constant bit rate (CBR). The fractional part gives the computational complexity from 0 (fastest, worst quality) to 10 (slowest, best quality). If no fractional part is given, you get quality 10. The quality is an integer rounded from the value given, so to use a quality of 0, give a decimal part less than 0.05.

SoX can read and write Opus files; with **−t sndfile** it can only read them.

oss (optional)
The 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_ng infile -t oss
sox_ng infile -t oss /dev/dsp
sox_ng -b 16 -t oss /dev/dsp outfile
```

.paf, **.fap** (with sndfile, also with **−t ffmpeg)**
Ensoniq PARIS file format (big and little-endian respectively).

.pls A *playlist* format containing 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.

.pcm Audio track format for the MSU-1 (Media Streaming Unit, revision 1), an enhancement chip specification for the Super Nintendo Entertainment System (SNES) / Super Famicom (SFC), designed by the late Byuu (aka Near). Audio files are 44.1 kHz, 16-bit stereo uncompressed signed PCM in little-endian order ("Red Book" CD audio is big-endian), left channel first. They begin with an 8-byte header: the first four bytes are "MSU1" in ASCII and the next four are a 32-bit unsigned integer that define the loop point in samples.

.prc Psion Record are 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, use the **−e ima-adpcm** 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.

pulseaudio (optional)
PulseAudio is a cross-platform networked sound server. The PulseAudio driver supports both playing and recording of audio.

If a file name is given to this driver, it selects the pulse server, the audio source or sink device and/or the application name under which SoX registers itself with PulseAudio. **app:foo** changes the application name from the default of **SoX**, anything else with a colon on it is a server name and anything without a colon is the device name. To specify more than one of these, or multiple servers to try in sequence, use a single filename with the entries separated by spaces; as application names can contain spaces, an "app:" field must come last. For example:

```
sox_ng in.au -t pulseaudio alsa_output.pci-0000_01_00.1.hdmi-stereo
```

Where **pactl list sinks short** gives a list of playing devices and **pactl list sources short** a list of recording devices.

You can also set the PulseAudio device with SoX's **AUDIODEV** and PulseAudio's **PULSE_SOURCE** and **PULSE_SINK** environment variables (**AUDIODEV** overrides **PULSE_*** and a filename overrides both) and the server with **PULSE_SERVER**. For details of the server name's syntax, search for "pulseaudio format of server strings".

.pvf (with sndfile)
Portable Voice Format.

.ra (with ffmpeg)
RealAudio format.

raw Headerless audio data. See the first entry in this list for details.

.rm (with ffmpeg)
RealMedia format.

.rso (with ffmpeg)
Lego Mindstorms RSO format.

SoX can only autodetect this type of file from its filename extension; if it is read from 'standard input' (stdin) or from a file whose name does not end in '.rso', you will need to prefix it with '-t ffmpeg'.

.sbc (with ffmpeg)
Bluetooth SIG low-complexity subband audio format.

SoX can only autodetect this type of file from its filename extension; if it is read from 'standard input' (stdin) or from a file whose name does not end in '.sbc', you will need to prefix it with '-t ffmpeg'.

.sd2 (with sndfile)
Sound Designer 2 format.

.sds (with sndfile, also with **–t ffmpeg**)
MIDI Sample Dump Standard.

.sf (also with **–t sndfile** or **–t ffmpeg**)
IRCAM SDIF (Institut de Recherche et Coordination Acoustique/Musique Sound Description Interchange Format) is used by academic music software such as the CSound package and the MixView sound sample editor.

.sln (also with **–t ffmpeg**)
Asterisk PBX 'signed linear' 8khz, 16-bit signed integer, little-endian raw format.

.smjpeg (with ffmpeg)
Loki SDL MJPEG.

.smp SMP files are for use with the PC-DOS package SampleVision by Turtle Beach Softworks, which communicates with several MIDI samplers. All sample rates are supported by the package although not all are supported by the samplers themselves. Loop points are currently ignored.

.snd Several file formats use the **.snd** extension.

The main one was by NeXT, essentially the same as Sun Microsystems' **.au** format - see **.au**

Apple made another **.snd** format in which the first two bytes are a 16-bit integer representing the numbers 1 or 2 but which can often be read as a raw format.

Akai had a **.snd** file format for its MPC range of samplers of which the first byte contains the number 1 and the second the number 4. See **.mpc2k**

There are also Sounder and SoundTool files from MS-DOS/Windows in the early '90s. See **.sndr** and **.sndt**.

Lastly, the HOM-BOT Robot Vacuum Cleaner and the V.Flash Home Entertainment System use **.snd** audio files which are raw single-channel 16-bit 16kHz PCM and the Unity Game Engine uses a compressed format called **.snd**.

sndfile (optional)
This is a pseudo-type that forces libsndfile to be used. For writing files, the actual file type is taken from the output file name; for reading them, it is deduced from the file.

sndio (optional)
The OpenBSD audio device driver supports both playing and recording audio.

.sndr Sounder files are an MS-DOS/Windows format from the early '90s that usually have the extension '.snd'.

.sndt SoundTool files are another MS-DOS/Windows format from the early '90s that usually have the extension '.snd'.

.sou An alias for the **.u8** raw format.

.sox (also with **−t ffmpeg**)
SoX's native uncompressed PCM format is intended for storing or piping audio at intermediate processing points between SoX invocations. It has much in common with WAV, AIFF and AU uncompressed PCM formats but has the following specific characteristics: the PCM samples are stored as 32 bit signed integers, the samples are stored (by default) as 'native endian' and the number of samples in the file is recorded as a 64-bit integer. Comments are also supported.

See the section '**Special Filenames**' in **sox_ng**(1) for examples of using the **.sox** format with pipes.

.spdif (with ffmpeg)
IEC 61937 S/PDIF format.

SoX can only autodetect this type of file from its filename extension; if it is read from 'standard input' (stdin) or from a file whose name does not end in '.spdif', you will need to prefix it with '-t ffmpeg'.

.sph, **.nist** (also with **−t sndfile** or **−t ffmpeg**)
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 header information that says the data is compressed using *shorten* compression and will treat the data as either μ-law or PCM. SoX and the command line *shorten* program can be run together using pipes to encompasses the data and then pass the result to SoX for processing.

.spx, **.speex** (with ffmpeg)
Ogg Speex format is for high compression of speech that, in VBR mode, achieves higher quality than AMR or GSM, but is now considered superseded by their more recent Opus codec.

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

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

or

```
sox_ng infile -t sunau -e mu-law -c 1 /dev/audio
```

for older Sun equipment.

.svcd (with ffmpeg)
Another name for **.mov**.

.tta (with ffmpeg)
True Audio format.

.vag (with ffmpeg)
Sony PS2 VAG format.

.txw TXW is a file format from the Yamaha TX-16W sampling keyboard which wrote samples onto IBM/PC-format 3·5" floppies at sample rates of 16kHz, 33kHz and 50kHz, all exact divisors of 100kHz.

SoX handles reading of files which do not have the sample rate field set to one of the expected rates by looking at some other bytes in the attack/loop length fields and defaulting to 33kHz if the sample rate is still unknown.

.vcd (with ffmpeg)
Another name for **.mov**.

.vms See **.dvms**.

.vob (with ffmpeg)
Another name for **.mov**.

.voc (also with **–t sndfile** or **–t ffmpeg**)
Sound Blaster 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 reads but cannot write VOC files with multiple blocks and files containing μ-law, A-law and 2/3/4-bit ADPCM samples.

.vorbis See **.ogg**.

.vox Headerless files of Dialogic/OKI ADPCM audio data commonly come 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. While 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_ng input.vox output.wav highpass 10
```

.w64 (with sndfile, also with **–t ffmpeg**)
Sonic Foundry's 64-bit RIFF/WAV format.

SoX can only autodetect this type of file from its filename extension; if it is read from 'standard input' (stdin) or from a file whose name does not end in '.w64', you will need to prefix it with '-t w64'.

.wav (also with **–t sndfile** or **–t ffmpeg**)
Microsoft .WAV RIFF files are the native audio file format of Windows and widely used for uncompressed audio.

Normally **.wav** files have all formatting information in their headers, so format options do not

usually need to be specified for input files. If any are, they override the file header and you will be warned to this effect. Output format options will cause a format conversion and the **.wav** is written appropriately.

SoX can read and write linear PCM, floating point, μ-law, A-law, MS ADPCM and IMA (or DVI) ADPCM-encoded samples. WAV files can also contain audio encoded in other ways not currently supported with SoX (e.g. MP3); in some cases such a file can still be read by SoX by overriding the file type, e.g.

```
play -t mp3 mp3-encoded.wav
```

Natively, SoX can only read WAV files with a bit-depth of 8, 16, 24 or 32; files with other bit-depths can be read by preceding them with `-t sndfile`.

Big endian versions of RIFF files, called RIFX, are also supported. To write a RIFX file, use the **–B** output file option.

See also **.wavpcm**.

waveaudio (optional)
The MS-Windows native audio device driver. Examples:

```
sox_ng infile -t waveaudio
sox_ng infile -t waveaudio default
sox_ng infile -t waveaudio 1
sox_ng infile -t waveaudio "High Definition Audio Device"
```

If the device name is omitted, **-1**, or **default**, you get the 'Microsoft Wave Mapper' device. Wave Mapper means 'use the system default audio devices' and you can control what 'default' means via the OS Control Panel.

If the given device name is some other number, you get that audio device by its index, so recording with device name **0** would get the first input device (perhaps the microphone), **1** would get the second (perhaps line in), etc. Playback using device name **0** will get the first output device (usually the only audio device).

If the given device name is something other than a number, SoX tries to match it (to a maximum of 31 characters) against the names of the available devices.

.wavpcm
A non-standard but widely used variant of **.wav**. Some applications cannot read a standard WAV file header for PCM-encoded data with a 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. SoX will automatically detect and read WAV files with a non-standard header.

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

```
sox_ng infile.any -t wavpcm -e signed-integer outfile.wav
```

.webm (with ffmpeg)
See **.mkv**.

.wma (with ffmpeg)
Windows Media Audio format.

.wsaud (with ffmpeg)
Westwood Studios audio format.

.wsd Wideband Single-bit Data is the same as **.dsf** but with a different header.

.wtv (with ffmpeg)
Windows Television format.

.wv (also with **−t sndfile** or **−t ffmpeg**)
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 is used on Psion SIBO PDAs (Series 3 and similar).

.xa (also with **−t ffmpeg**)
Maxis XA files 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 (with sndfile)
Fasttracker 2 Extended Instrument format.

REFERENCES

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

[2] Wikipedia, *PLS*, http://en.wikipedia.org/wiki/PLS_(file_format)

SEE ALSO

sox_ng(1), `https://codeberg.org/sox_ng`

AUTHORS

Lance Norskog, Chris Bagwell and many other authors and contributors listed in the README file that is distributed with the source code.