NAME

audioconvert – convert audio file formats

SYNOPSIS

audioconvert [–**pF**] [–**f** *outfmt*] [–**o** *outfile*] [[–**i** *infmt*] [*file* . . .]] . . .

DESCRIPTION

audioconvert converts audio data between a set of supported audio encodings and file formats. It can be used to compress and decompress audio data, to add audio file headers to raw audio data files, and to convert between standard data encodings, such as µ-law and linear PCM.

If no filenames are present, **audioconvert** reads the data from the standard input stream and writes an audio file to the standard output. Otherwise, input files are processed in order, concatenated, and written to the output file.

Input files are expected to contain audio file headers that identify the audio data format. If the audio data does not contain a recognizable header, the format must be specified with the –**i** option, using the *rate*, *encoding*, and *channels* keywords to identify the input data format.

The output file format is derived by updating the format of the first input file with the format options in the –**f** specification. If –**p** is not specified, all subsequent input files are converted to this resulting format and concatenated together. The output file will contain an audio file header, unless *format=raw* is specified in the output format options.

Input files may be converted in place by using the –**p** option. When –**p** is in effect, the format of each input file is modified according to the –**f** option to determine the output format. The existing files are then overwritten with the converted data.

The **file**(1) command decodes and prints the audio data format of Sun audio files.

OPTIONS

–**p** *In Place*: The input files are individually converted to the format specified by the –**f** option and rewritten. If a target file is a symbolic link, the underlying file will be rewritten. The –**o** option may not be specified with –**p**.

–**F** *Force*: This option forces **audioconvert** to ignore any file header for input files whose format is specified by the –**i** option. If –**F** is not specified, **audioconvert** ignores the –**i** option for input files that contain valid audio file headers.

–**f** *outfmt* *Output Format*: This option is used to specify the file format and data encoding of the output file. Defaults for unspecified fields are derived from the input file format. Valid keywords and values are listed in the next section.

–**o** *outfile* *Output File*: All input files are concatenated, converted to the output format, and written to the named output file. If –**o** and –**p** are not specified, the concatenated output is written to the standard output. The –**p** option may not be specified with –**o**.

–i *infmt* — *Input Format*: This option is used to specify the data encoding of raw input files. Ordinarily, the input data format is derived from the audio file header. This option is required when converting audio data that is not preceded by a valid audio file header. If **–i** is specified for an input file that contains an audio file header, the input format string will be ignored, unless **–F** is present. The format specification syntax is the same as the **–f** output file format.

Multiple input formats may be specified. An input format describes all input files following that specification, until a new input format is specified.

file — *File Specification*: The named audio files are concatenated, converted to the output format, and written out. If no filename is present, or if the special filename '–' is specified, audio data is read from the standard input.

–? — *Help*: Print a command line usage message.

FORMAT SPECIFICATION

The syntax for the input and output format specification is:

keyword=value[,*keyword=value* . . .]

with no intervening whitespace. Unambiguous values may be used without the preceding *keyword*=.

rate — The audio sampling rate is specified in samples per second. If a number is followed by the letter **k**, it is multiplied by 1000 (for example, 44.1k = 44100). Standard of the commonly used sample rates are: 8k, 16k, 32k, 44.1k, and 48k.

channels — The number of interleaved channels is specified as an integer. The words **mono** and **stereo** may also be used to specify one and two channel data, respectively.

encoding — This option specifies the digital audio data representation. Encodings determine precision implicitly (**ulaw** implies 8-bit precision) or explicitly as part of the name (for example, **linear16**). Valid encoding values are:

- **ulaw** — CCITT G.711 μ-law encoding. This is an 8-bit format primarily used for telephone quality speech.
- **alaw** — CCITT G.711 A-law encoding. This is an 8-bit format primarily used for telephone quality speech in Europe.
- **linear8, linear16, linear32** — Linear Pulse Code Modulation (PCM) encoding. The name identifies the number of bits of precision. **linear16** is typically used for high quality audio data.
- **pcm** — Same as **linear16**.

g721 CCITT G.721 compression format. This encoding uses Adaptive Delta Pulse Code Modulation (ADPCM) with 4-bit precision. It is primarily used for compressing µ-law voice data (achieving a 2:1 compression ratio).

g723 CCITT G.723 compression format. This encoding uses Adaptive Delta Pulse Code Modulation (ADPCM) with 3-bit precision. It is primarily used for compressing µ-law voice data (achieving an 8:3 compression ratio). The audio quality is similar to G.721, but may result in lower quality when used for non-speech data.

The following encoding values are also accepted as shorthand to set the sample rate, channels, and encoding:

voice Equivalent to **encoding=ulaw,rate=8k,channels=mono**.

cd Equivalent to **encoding=linear16,rate=44.1k,channels=stereo**.

dat Equivalent to **encoding=linear16,rate=48k,channels=stereo**.

format This option specifies the audio file format. Valid formats are:

sun Sun compatible file format (the default).

raw Use this format when reading or writing raw audio data (with no audio header), or in conjunction with an **offset** to import a foreign audio file format.

offset (**–i** *only*) Specify a byte offset to locate the start of the audio data. This option may be used to import audio data that contains an unrecognized file header.

EXAMPLES

Record voice data and compress it before storing it to a file:

example% audiorecord | audioconvert –f g721 > mydata.au

Concatenate two Sun format audio files, regardless of their data format, and output an 8-bit µ-law, 16 kHz, mono file:

example% audioconvert –f ulaw,rate=16k,mono –o outfile.au infile1 infile2

Convert a directory containing raw voice data files, in place, to Sun format (adds a file header to each file):

example% audioconvert –p –i voice –f sun ∗.au

SEE ALSO

audioplay(1), **audiorecord**(1), **file**(1)

NOTES

The algorithm used for converting multi-channel data to mono is implemented by simply summing the channels together. If the input data is perfectly in phase (as would be the case if a mono file is converted to stereo and back to mono), the resulting data may contain some distortion.