NAME

cdda2wav − a sampling utility that dumps CD audio data into wav sound files

SYNOPSIS

cdda2wav [**-c** *chans*] [**-s**] [**-m**] [**-b** *bits*] [**-r** *rate*] [**-a** *divider*] [**-t** *track*[*+endtrack*]] [**-i** *index*] [**-o** *offset*] [**-d** *duration*] [**-x**] [**-q**] [**-w**] [**-v**] [**-V**] [**-J**] [**-R**] [**-P** *sectors*] [**-F**] [**-G**] [**-T**] [**-e**] [**-p** *percentage*] [**-n** *sectors*] [**-l** *buffers*] [**-N**] [**-J**] [**-H**] [**-B**] [**-D** *device*] [**-A** *auxdevice*] [**-I** *interface*] [**-O** *audiotype*] [**-C** *input-endianess*] [**-E** *output-endianess*] [**-M** *count*] [**-S** *speed*] [*audio.wav*]

DESCRIPTION

cdda2wav can retrieve audio tracks from CDROM drives (see README for a list of drives) that are capable of reading audio data digitally to the host (**CDDA**). Other drives may have been added as well when you read this.

OPTIONS

−D *device* *--device*
: uses **device** as the source for CDDA reading: For example /dev/cdrom for the **cooked_ioctl** interface and Bus,ID,Lun for the **generic_scsi** interface. This has to correspond with the interface setting (see below). The setting of the environment variable **CDDA_DEVICE** is overridden by this option.

−A *auxdevice* *--auxdevice*
: uses **auxdevice** as CDROM drive for ioctl usage.

−I *interface* *--interface*
: specifies the interface for CDROM access: **generic_scsi** or (on Linux systems) **cooked_ioctl**.

−c *channels* *--channels*
: uses **1** for mono, or **2** for stereo recording, or **s** for stereo recording with both channels swapped.

−s **--stereo**
: sets to stereo recording.

−m **--mono**
: sets to mono recording.

−x **--max**
: sets maximum (CD) quality.

−b *bits* *--bits-per-sample*
: sets bits per sample per channel: **8**, **12** or **16**.

−r *rate* *--rate*
: sets rate in samples per second. Possible values are listed with the **−R** option.

−a *divider* *--divider*
: sets rate to 44100Hz / divider. Possible values are listed with the **−R** option.

−R **--dump-rates**
: shows a list of all sample rates and their dividers.

−P **sectors** **--set-overlap**
: sets the initial number of overlap *sectors* for jitter correction.

−n *sectors* *--sectors-per-request*
: reads *sectors* per request.

−l *buffers* *--buffers-in-ring*
: uses a ring buffer with *buffers* total.

−t *track+endtrack* *--track*
: selects the start track and optionally the end track.

−i *index* *--index*
selects the start index.

−o *offset* *--offset*
starts *offset* sectors behind start track (one sector equivalents 1/75 seconds).

−O audiotype --output-format
can be *wav* (for wav files) or *aiff* (for apple/sgi aiff files) or *aifc* (for apple/sgi aifc files) or *au* or *sun* (for sun .au PCM files) or *cdr* or *raw* (for headerless files to be used for cd writers).

−C *endianess* *--cdrom-endianess*
sets endianess of the input samples to 'little', 'big' or 'guess' to override defaults.

−E *endianess* *--output-endianess*
sets endianess of the output samples to 'little' or 'big' to override defaults.

−d *duration* *--duration*
sets recording time in seconds or **0** for whole track.

−B --bulk
copies each track into a seperate file.

−w --wait
waits for signal, then start recording.

−F --find-extremes
finds extrem amplitudes in samples.

−G --find-mono
finds if input samples are in mono.

−T --deemphasize
undo the effect of pre-emphasis in the input samples.

−e --echo
copies audio data to sound device e.g. **/dev/dsp**.

−p percentage --set-pitch
changes pitch of audio data copied to sound device.

−v level --verbose-level
prints verbose information about the CD. **Level** is a sum of powers of two in the range 0 up to 63. Each power of two controls the type of information to be reported. center box ; r| l|.

Power Description _ 1 show table of contents 2 show a summary of the recording parameters 4 determine and display index offsets 8 retrieve and display the media catalog number MCN 16 retrieve and display all Intern. Standard Recording Codes ISRC 32 show the table of contents in start sector notation 64 show the table of contents with track titles (when available)

−N --no-write
: does not write to a file, it just reads (for debugging purposes).

−J --info-only
: does not write to a file, it just gives information about the disc.

−H --no-infofile
: does not write an info file and a cddb file.

−M count --md5
: enables calculation of MD-5 checksum for 'count' bytes from a beginning of a track.

−S speed --speed-select
: sets the cdrom device to one of the selectable speeds for reading.

−q --quiet
: quiet operation, no screen output.

−V --verbose-SCSI
: enable SCSI command logging to the console. This is mainly used for debugging

−J --version
: display version of cdda2wav on standard output.

Defaults depend on the
: **Makefile** and **environment variable** settings (currently **CDDA_DEVICE**).

ENVIRONMENT VARIABLES

CDDA_DEVICE is used to set the device name. The device naming is compatible with Jvrg Schilling's cdrecord package.

DISCUSSION

cdda2wav is able to read parts of an **audio** CD or **multimedia** CDROM (containing audio parts) directly digitally. These parts can be written to a file, a pipe, or to a sound device.

cdda2wav stands for **CDDA** to **WAV** (where **CDDA** stands for compact disc digital audio and **WAV** is a sound sample format introduced by MS Windows). It allows copying **CDDA** audio data from the CDROM drive into a file on your ram-, hard-, floppy- or whatever-disk in **WAV** or other formats.

The latest versions try to get higher real-time scheduling priorities to ensure smooth (uninterrupted) operation. These priorities are available for super users and are higher than those of 'normal' processes. Thus delays are minimized. Please note that you need newer kernels and c libraries to take advantage of this feature.

If your CDROM (not yet for SCSI drives) is on device **/dev/cdrom** and it is loaded with an audio CD, you may simply invoke **cdda2wav** and it will create the sound file **audio.wav** recording the whole track beginning with track 1 in stereo at 16 bit at 44100 Hz sample rate, if your file system has enough space free. Otherwise recording time will be limited. SCSI drives have to use different devices (see files **README** and **README.INSTALL** for details).

HINTS ON OPTIONS

Options Most of the options are used to control the format of the WAV file. In the following text all of them are described.

Select Device
: **−D** *device* selects the CDROM drive device to be used. The specifier given should correspond to the selected interface (see below). **CHANGE!** For the cooked_ioctl interface this is the cdrom device descriptor as before. **The SCSI devices used with the addressed with their**

SCSI-Bus, SCSI-Id, and SCSI device descriptor!!! One example for a SCSI CDROM drive on bus 0 with SCSI ID 3 and lun 0 is -D0,3,0.

Select Auxiliary device

−A *auxdevice* is necessary for CD-Extra handling. For Non-SCSI-CDROM drives this is the same device as given by -D (see above). For SCSI-CDROM drives it is the CDROM drive (SCSI) device (i.e. **/dev/sr0**) corresponding to the SCSI device (i.e. **0,3,0**). It has to match the device used for sampling.

Select Interface

−I *interface* selects the CDROM drive interface. For SCSI drives use generic_scsi (cooked_ioctl may not yet be available for all devices): **generic_scsi** and **cooked_ioctl**. The first uses the generic SCSI interface, the latter uses the ioctl of the CDROM driver. The latter variant works only when the kernel driver supports **CDDA** reading. This entry has to match the selected CDROM device (see above).

Enable echo to soundcard

−e copies audio data to the sound card while recording, so you hear it nearly simultaneously. The soundcard gets the same data that is recorded. This is time critical, so it works best with the **−q** option. To use **cdda2wav** as a pseudo CD player without recording in a file you could use **cdda2wav −q −e −t2 −d0 −N** to play the whole second track. This feature reduces the recording speed to at most onefold speed. You cannot make better recordings than your sound card can play (since the same data is used).

Change pitch of echoed audio

−p percentage changes the pitch of all audio echoed to a sound card. Only the copy to the soundcard is affected, the recorded audio samples in a file remain the same. Normal pitch, which is the default, is given by 100%. Lower percentages correspond to lower pitches, i.e. -p 50 transposes the audio output one octave lower. See also the script **pitchplay** as an example. This option was contributed by Raul Sobon.

Select mono or stereo recording

−m or **−c 1** selects mono recording (both stereo channels are mixed), **−s** or **−c 2** or **−c s** selects stereo recording (doubles file size). Parameter s will swap both sound channels.

Select maximum quality

−x will set stereo, 16 bits per sample at 44.1 KHz (full CD quality). Note that other format options given later can change this setting.

Select sample quality

−b 8 specifies 8 bit (1 Byte) for each sample in each channel; **−b 12** specifies 12 bit (2 Byte) for each sample in each channel; **−b 16** specifies 16 bit (2 Byte) for each sample in each channel (Ensure that your sample player or sound card is capable of playing 12-bit or 16-bit samples). Selecting 12 or 16 bits doubles file size. 12-bit samples are aligned to 16-bit samples, so they waste some disk space.

Select sample rate

−r *samplerate* selects a sample rate. *samplerate* can be in a range between 44100 and 900. Option **−R** lists all available rates.

Select sample rate divider

−a *divider* selects a sample rate divider. *divider* can be minimally 1 and maximally 50.5 and everything between in steps of 0.5. Option **−R** lists all available rates.

To make the sound smoother at lower sampling rates, **cdda2wav** sums over *n* samples (where *n* is the specific dividend). So for 22050 Hertz output we have to sum over 2 samples, for 900 Hertz we have to sum over 49 samples. This cancels higher frequencies. Standard sector size of an audio CD (ignoring additional information) is 2352 Bytes. In order to finish summing for an output sample at sector boundaries the rates above have to be choosen. Arbitrary sampling rates in high quality would require some interpolation scheme, which needs much more sophisticated programming.

List a table of all sampling rates

−R shows a list of all sample rates and their dividers. Dividers can range from 1 to 50.5 in steps of 0.5.

Select start track and optionally end track

−t *n+m* selects **n** as the start track and optionally **m** as the last track of a range to be recorded. These tracks must be from the table of contents. This sets the track where recording begins. Recording can advance through the following tracks as well (limited by the optional end track or otherwise depending on recording time). Whether one file or different files are then created depends on the **−B** option (see below).

Select start index

−i *n* selects the index to start recording with. Indices other than 1 will invoke the index scanner, which will take some time to find the correct start position. An offset may be given additionally (see below).

Set recording time

−d n sets recording time to *n* seconds or set recording time for whole track if *n* is zero. Please note that if track ranges are being used they define the recording time as well thus overriding any **−d** option specified times.

Recording time is defined as the time the generated sample will play (at the defined sample rate). Since it's related to the amount of generated samples, it's not the time of the sampling process itself (which can be less or more). It's neither strictly coupled with the time information on the audio CD (shown by your hifi CD player). Differences can occur by the usage of the **−o** option (see below). Notice that recording time will be shortened, unless enough disk space exists. Recording can be aborted at anytime by pressing the break character (signal SIGQUIT).

Record a complete audio CD in one go in different files

−B copies each track into a seperate file. A base name can be given. File names have an appended track number and an extension corresponding to the audio format. To record all audio tracks of a CD, use a sufficient high duration (i.e. -d99999).

Set start sector offset

−o *sectors* increments start sector of the track by *sectors*. By this option you are able to skip a certain amount at the beginning of a track so you can pick exactly the part you want. Each sector runs for 1/75 seconds, so you have very fine control. If your offset is so high that it would not fit into the current track, a warning message is issued and the offset is ignored. Recording time is not reduced. (To skip introductory quiet passages automagically, use the **−w** option see below.)

Wait for signal option

−w Turning on this option will suppress all silent output at startup, reducing possibly file size. **cdda2wav** will watch for any signal in the output signal and switches on writing to file.

Find extrem samples

−F Turning on this option will display the most negative and the most positive sample value found during recording for both channels. This can be useful for readjusting the volume. The values shown are not reset at track boundaries, they cover the complete sampling process. They are taken from the original samples and have the same format (i.e. they are independent of the selected output format).

Find if input samples are in mono

−G If this option is given, input samples for both channels will be compared. At the end of the program the result is printed. Differences in the channels indicate stereo, otherwise when both channels are equal it will indicate mono.

Undo the pre-emphasis in the input samples

−T Some older audio CDs are recorded with a modified frequency response called pre-emphasis. This is found mostly in classical recordings. The correction can be seen in the flags of the Table Of Contents often. But there are recordings, that show this setting only in the sub-channels. If you know you have a pre-emphasized recording and want cdda2wav to undo this, you have to specify this option. A reverse filtering is done before the samples are written into the audio file.

Set audio format

−O audiotype can be *wav* (for wav files) or *au* or *sun* (for sun PCM files) or *cdr* or *raw* (for headerless files to be used for cd writers). All file samples are coded in linear pulse code modulation (as done in the audio compact disc format). This holds for all audio formats. Wav files are compatible to Wind*ws sound files, they have lsb,msb byte order as being used on the audio cd. The default filename extension is ’.wav’. Sun type files are not like the older common logarithmically coded .au files, but instead as mentioned above linear PCM is used. The byte order is msb,lsb to be compatible. The default filename extension is ’.au’. The AIFF and the newer variant AIFC from the Apple/SGI world store their samples in bigendian format (msb,lsb). In AIFC no compression is used. Finally the easiest ’format’, the cdr aka raw format. It is done per default in msb,lsb byte order to satisfy the order wanted by most cd writers. Since there is no header information in this format, the sample parameters can only be identified by playing the samples on a soundcard or similiar. The default filename extension is ’.cdr’ or ’.raw’.

Select cdrom drive reading speed

−S speed allows to switch the cdrom drive to a certain level of speed in order to reduce read errors. The actual speed factor can generally be given completely freely, since there are often less settings than factors possible. Details depend very much on the cdrom drives. An argument of 0 for example is often the default speed of the drive, a value of 1 often selects single speed.

Enable MD5 checksums

−M count enables calculation of MD-5 checksum for ’count’ bytes from the beginning of a track. This was introduced for quick comparisons of tracks.

Do linear or overlapping reading of sectors

−P sectors sets the given number of sectors for initial overlap sampling for jitter correction. Two cases are to be distinguished. For nonzero values, some sectors are read twice to enable cdda2wav’s jitter correction. If an argument of zero is given, no overlap sampling will be used. **NEWS!** For nonzero overlap sectors cdda2wav dynamically adjusts the setting during sampling (like cdparanoia does). If no match can be found, cdda2wav retries the read with an increased overlap. If the amount of jitter is lower than the current overlapped samples, cdda2wav reduces the overlap setting, resulting in a higher reading speed. The argument given has to be lower than the total number of sectors per request (see option -*n* below). Cdda2wav will check this setting and issues a error message otherwise. The case of zero sectors is nice on low load situations or errorfree (perfect) cdrom drives and perfect (not scratched) audio cds.

Set the transfer size

−n sectors will set the transfer size to the specified sectors per request.

Set number of ring buffer elements

−l buffers will allocate the specified number of ring buffer elements.

Set endianess of input samples

−C endianess will override the default settings of the input format. Endianess can be set explicitly to "little" or "big" or to the automatic endianess detection based on voting with "guess".

Set endianess of output samples

−E endianess (endianess can be "little" or "big") will override the default settings of the output format.

Verbose option

−v level prints more information. A binary mask allows selection of different information.

0 keeps quiet

1 displays the table of contents

2 displays a summary of recording parameters

4 invokes the index scanner and displays start positions of indices

8 retrieves and displays a media catalog number

16 retrieves and displays international standard recording codes

32 displays track start positions in absolute sector notation

To combine several requests just add the numbers and give the sum as argument.

The table of contents

The display will show the table of contents with number of tracks and total time (displayed in *mm*:*ss*.*hh* format, *mm*=minutes, *ss*=seconds, *hh*=rounded 1/100 seconds). The following list displays track number and track time for each entry. The summary gives a line per track describing the type of the track.

track preemphasis copypermitted tracktype chans

The **track** column holds the track number. **preemphasis** shows if that track has been given a non linear frequency response. NOTE: You can undo this effect with the **−T** option. **copypermitted** indicates if this track is allowed to copy. **tracktype** can be data or audio. On multimedia CDs (except hidden track CDs) both of them should be present. **channels** is defined for audio tracks only. There can be two or four channels.

No file output

−N this debugging option switches off writing to a file.

No infofile generation

−H this option switches off creation of an info file and a cddb file.

Verbose SCSI logging

−V this option switches on logging of SCSI commands. This will produce a lot of output, which is needed for debugging purposes. The format is the same as being used with the cdrecord package from Jvrg Schilling. I will not describe it here.

Quiet option

−q suppresses all screen output except error messages. Further it avoids any sub q channel requests for position reports in the recording display. That reduces cpu time resources and allows feeding a soundcard with audio data on the fly.

Just show information option

−J does not write a file, it only prints information about the disc (depending on the **−v** option). This is just for information purposes.

HINTS ON USAGE

Don’t create samples you cannot read. First check your sample player software and sound card hardware. I experienced problems with very low sample rates (stereo <= 1575 Hz, mono <= 3675 Hz) when trying to play them with standard WAV players for sound blaster (maybe they are not legal in **WAV** format). Most CD-Writers insist on audio samples in a bigendian format. Now cdda2wav supports the **−E endianess** option to control the endianess of the written samples.

If your hardware is fast enough to run cdda2wav uninterrupted and your CD drive is one of the ’perfect’ ones, you will gain speed when switching all overlap sampling off with the **−P 0** option. Further fine tuning can be done with the **−n sectors** option. You can specify how much sectors should be requested in one go.

Cdda2wav supports **pipes** now. Use a filename of − to let cdda2wav output its samples to standard output.

Conversion to other sound formats can be done using the **sox** program package (although the use of **sox -x** to get byte swaps should be no more necessary; see option **−E** to change the output byteorder).

If you really want to sample more than one track in one go into different files, this is currently possible with the **−B** option. When recording time exceeds the track limit a new file will be opened for the next track.

WARNING

IMPORTANT: it is prohibited to sell copies of copyrighted material by noncopyright holders. This program may not be used to circumvent copyrights.

BUGS

A kernel bug (version < 1.1.20) prevented successful operation of mode select commands.

Performance may not be optimal on slower systems.

The index scanner may give timeouts.

The resampling (rate conversion code) uses polynomial interpolation, which is not optimal.

Enhanced CD extra information gathering is incomplete due to lack of documentation. Donations for purchase of a copy of the Blue Book might help.

CD Text extra information gathering is incomplete due to lack of the latest Red Book additions. Donations for purchase of a copy of the Red Book are welcome.

Cdda2wav should use threads.

The paranoia code is currently disabled.

When using jitter correction with ATAPI cdrom drives, track endings are not handled correctly.

ACKNOWLEDGEMENTS

Thanks goto Project MODE (http://www.mode.net/) and Fraunhofer Institut f|r integrierte Schaltungen (FhG-IIS) (http://www.iis.fhg.de/) for financial support. Rammi has helped a lot with the debugging and showed a lot of stamina when hearing 100 times the first 16 seconds of the first track of the Krupps CD. Paranoia patches contributed by Monty xiphmont@mit.edu.

AUTHOR

Heiko Eissfeldt heiko@colossus.escape.de

DATE

25 Mar 1999