Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'cd-paranoia.1' command

$ man cd-paranoia.1

cd-paranoia(1) General Commands Manual cd-paranoia(1)

NAME

cd-paranoia - 9.8 (Paranoia release III via libcdio) - an audio CD reading utility which includes extra data verification features

SYNOPSIS

cd-paranoia [options] span [outfile]

DESCRIPTION

cd-paranoia retrieves audio tracks from CDDA capable CD-ROM drives. The data can be saved to a file or directed to standard output in WAV, AIFF, AIFF-C or raw format. Most ATAPI, SCSI and several proprietary CD-ROM drive makes are supported; cd-paranoia can determine if the tar? get drive is CDDA capable.

In addition to simple reading, cd-paranoia adds extra-robust data veri? fication, synchronization, error handling and scratch reconstruction capability.

This version uses the libcdio library for interaction with a CD-ROM drive. The jitter and error correction however are the same as used in Xiph's cdparanoia.

OPTIONS

-A --analyze-drive

Run and log a complete analysis of drive caching, timing and reading behavior; verifies that cdparanoia is correctly model? ling a specific drive's cache and read behavior. Implies -vQL.

-v --verbose

Be absurdly verbose about the autosensing and reading process. Good for setup and debugging.

-q --quiet

Do not print any progress or error information during the read? ing process.

-e --stderr-progress

Force output of progress information to stderr (for wrapper scripts).

-V --version

Print the program version and quit.

-Q --query

Perform CD-ROM drive autosense, query and print the CD-ROM table of contents, then quit.

-h --help

Print a brief synopsis of cd-paranoia usage and options.

-l --log-summary file

Save result summary to file.

-L --log-debug file

Save detailed device autosense and debugging output to a file.

-p --output-raw

Output headerless data as raw 16 bit PCM data with interleaved samples in host byte order. To force little or big endian byte order, use -r or -R as described below.

-r --output-raw-little-endian

Output headerless data as raw 16 bit PCM data with interleaved samples in LSB first byte order.

-R --output-raw-big-endian

Output headerless data as raw 16 bit PCM data with interleaved samples in MSB first byte order.

-w --output-wav

Output data in Microsoft RIFF WAV format (note that WAV data is always LSB first byte order).

-f --output-aiff

Output data in Apple AIFF format (note that AIFC data is always in MSB first byte order).

-a --output-aifc

Output data in uncompressed Apple AIFF-C format (note that AIFF-C data is always in MSB first byte order).

-B --batch

Cdda2wav-style batch output flag; cd-paranoia will split the output into multiple files at track boundaries.   Output file names are prepended with 'track#.'

-c --force-cdrom-little-endian

Some CD-ROM drives misreport their endianness (or do not report it at all); it's possible that cd-paranoia will guess wrong. Use -c to force cd-paranoia to treat the drive as a little en? dian device.

-C --force-cdrom-big-endian

As above but force cd-paranoia to treat the drive as a big en? dian device.

-n --force-default-sectors n

Force the interface backend to do atomic reads of n sectors per read. This number can be misleading; the kernel will often split read requests into multiple atomic reads (the automated Paranoia code is aware of this) or allow reads only wihin a re? stricted size range. This option should generally not be used.

-d --force-cdrom-device device

Force the interface backend to read from device rather than the first readable CD-ROM drive it finds containing a CD-DA disc. This can be used to specify devices of any valid interface type (ATAPI, SCSI or proprietary).

-g --force-generic-device device

This option is an alias for -d and is retained for compatibil? ity.

-S --force-read-speed number

Use this option explicitly to set the read rate of the CD drive (where supported). This can reduce underruns on machines with slow disks, or which are low on memory.

-t --toc-offset number

Use this option to force the entire disc LBA addressing to shift by the given amount; the value is added to the beginning offsets in the TOC. This can be used to shift track boundaries for the whole disc manually on sector granularity. The next option does something similar...

-T --toc-bias

Some drives (usually random Toshibas) report the actual track beginning offset values in the TOC, but then treat the beginning of track 1 index 1 as sector 0 for all read operations. This results in every track seeming to start too late (losing a bit of the beginning and catching a bit of the next track). -T ac?counts for this behavior. Note that this option will cause cd-paranoia to attempt to read sectors before or past the known user data area of the disc, resulting in read errors at disc edges on most drives and possibly even hard lockups on some buggy hardware.

-O --sample-offset number

Some CD-ROM/CD-R drives will add an offset to the position on reading audio data. This is usually around 500-700 audio samples (ca. 1/75 second) on reading. So when cd-paranoia queries a spe?cific sector, it might not receive exactly that sector, but shifted by some amount.

Use this option to force the entire disc to shift sample position out?put by the given amount; This can be used to shift track boundaries for the whole disc manually on sample granularity. Note that if you are ripping something including the ending of the CD (e.g. the entire disk), this option will cause cd-paranoia to attempt to read partial sectors before or past the known user data area, probably causing read errors on most drives and possibly even hard lockups on some buggy hardware.

-E--force-overread

Force overreading into the lead-out portion of the disc. This
option is only applicable when using the +.B -O +option with a
positive sample offset value. Many drives are not capable of
reading into this portion of the disc and attempting to do so on
those drives will produce read errors and possibly hard lockups.

-Z --disable-paranoia

Disable all data verification and correction features. When us?
ing -Z, cd-paranoia reads data exactly as would cdda2wav with an
overlap setting of zero. This option implies that -Y is active.

-z --never-skip[=max_retries]

Do not accept any skips; retry forever if needed. An optional
maximum number of retries can be specified; for comparison, de?
fault without -z is currently 20.

-Y --disable-extra-paranoia

Disables intra-read data verification; only overlap checking at
read boundaries is performed. It can wedge if errors occur in
the attempted overlap area. Not recommended.

-X --abort-on-skip

If the read skips due to imperfect data, a scratch, whatever,
abort reading this track. If output is to a file, delete the
partially completed file.

-x --test-flags mask

Simulate CD-reading errors. This is used in regression testing,
but other uses might be to see how well a CD-ROM performs under
(simulated) CD degradation. mask specifies the artificial kinds
of errors to introduced; "or"-ing values from the selection be?
low will simulate the kind of specified failure.

0x10 - Simulate under-run reading

OUTPUT SMILIES

:-) Normal operation, low/no jitter

:-| Normal operation, considerable jitter

:-/ Read drift

:-P Unreported loss of streaming in atomic read operation

8-| Finding read problems at same point during reread; hard to cor?rect

:-0 SCSI/ATAPI transport error

:-(Scratch detected

;-(Gave up trying to perform a correction

8-X Aborted read due to known, uncorrectable error

:^D Finished extracting

PROGRESS BAR SYMBOLS

<space>

No corrections needed

-   Jitter correction required

+   Unreported loss of streaming/other error in read

!   Errors found after stage 1 correction; the drive is making the same error through multiple re-reads, and cd-paranoia is having trouble detecting them.

e   SCSI/ATAPI transport error (corrected)

V   Uncorrected error/skip

SPAN ARGUMENT

The span argument specifies which track, tracks or subsections of tracks to read. This argument is required. NOTE: Unless the span is a simple number, it's generally a good idea to quote the span argument to protect it from the shell.

The span argument may be a simple track number or an offset/span speci?fication. The syntax of an offset/span takes the rough form:

1[ww:xx:yy.zz]-2[aa:bb:cc.dd]

Here, 1 and 2 are track numbers; the numbers in brackets provide a finer grained offset within a particular track. [aa:bb:cc.dd] is in hours/minutes/seconds/sectors format. Zero fields need not be speci?fied: [::20], [:20], [20], [20.], etc, would be interpreted as twenty seconds, [10:] would be ten minutes, [.30] would be thirty sectors (75 sectors per second).

When only a single offset is supplied, it is interpreted as a starting offset and ripping will continue to the end of the track. If a single offset is preceeded or followed by a hyphen, the implicit missing off?set is taken to be the start or end of the disc, respectively. Thus:

1:[20.35]

Specifies ripping from track 1, second 20, sector 35 to the end of track 1.

1:[20.35]-

Specifies ripping from 1[20.35] to the end of the disc

-2 Specifies ripping from the beginning of the disc up to (and in?cluding) track 2

-2:[30.35]

Specifies ripping from the beginning of the disc up to 2:[30.35]

2-4 Specifies ripping from the beginning of track 2 to the end of track 4.

Again, don't forget to protect square brackets and preceeding hyphens from the shell.

EXAMPLES

A few examples, protected from the shell:

Query only with exhaustive search for a drive and full reporting of au?tosense:

```
cd-paranoia -vsQ
```

Extract an entire disc, putting each track in a separate file:

```
cd-paranoia -B
```

Extract from track 1, time 0:30.12 to 1:10.00:

```
cd-paranoia "1[:30.12]-1[1:10]"
```

Extract from the beginning of the disc up to track 3:

```
cd-paranoia -- "-3"
```

The "--" above is to distinguish "-3" from an option flag.

OUTPUT

The output file argument is optional; if it is not specified, cd-para?noia will output samples to one of cdda.wav, cdda.aifc, or cdda.raw de?pending on whether -w, -a, -r or -R is used (-w is the implicit de?

fault). The output file argument of - specifies standard output; all data formats may be piped.

ACKNOWLEDGEMENTS

cd-paranoia sprang from and once drew heavily from the interface of Heiko Eissfeldt's (heiko@colossus.escape.de) 'cdda2wav' package. cd-paranoia would not have happened without it.

Joerg Schilling has also contributed SCSI expertise through his generic SCSI transport library.

AUTHOR

Monty <monty@xiph.org>

Cdparanoia's homepage may be found at: https://www.xiph.org/paranoia/

Revised for use with libcdio by Rocky <rocky@gnu.org>

The libcdio homepage may be found at: https://www.gnu.org/soft?ware/libcdio/

version III release alpha 9.8 libcdio cd-paranoia(1)