

NAME

cdrdao - reads and writes CDs in disc-at-once mode

SYNOPSIS**cdrdao**

```
{ show-toc|toc-info|toc-size|read-toc|read-cd|read-cddb|show-data|read-test|disk-info|discid|msinfo|drive-info|unl
[--device device] [--source-device device] [--driver driver-id] [--source-driver driver-id] [--simulate]
[--speed writing-speed] [--blank-mode mode] [--datafile file] [--read-raw] [--read-subchan
[--no-mode2-mixed] mode] [--tao-source] [--tao-source-adjust link-blocks] [--fast-toc] [--buffers
buffer-count] [--multi] [--overburn] [--eject] [--swap] [--session] [--force] [--reload] [--keepimage]
[--on-the-fly] [--paranoia-mode mode] [--with-cddb] [--cddb-servers server-list] [--cddb-timeout
timeout] [--cddb-directory directory] [--tmpdir directory] [--keep] [--save] [-n] [-v verbose-level]
toc-file
```

DESCRIPTION

cdrdao creates audio and data CD-Rs in disk-at-once (DAO) mode driven by a description file called *toc-file*. In DAO mode it is possible to create non standard track pre-gaps that have other lengths than 2 seconds and contain nonzero audio data. This is for example useful to divide live recordings into tracks where 2 second gaps would be kind of irritating.

Instead of a *toc-file* a cue file (used by a famous DOS/Windows mastering tool) may be used. See the CUE FILES section for more details.

COMMANDS

The first argument must be one of the following commands:

show-toc

Print out a summary about what will be written to the CD-R.

toc-info

Prints out short toc-file summary.

toc-size

Prints total number of blocks for toc.

read-toc

Analyze each track of the inserted CD and create a *toc-file* that can be used to make a more or less exact copy of the CD. This command does not read out the audio or data tracks, use **read-cd** for this purpose.

You can specify a filename for the data file via the **--datafile** option.

read-cd

Copies all tracks of the inserted CD to an image file and creates a corresponding *toc-file*. The name of the image file defaults to "data.bin" if no **--datafile** option is given.

read-cddb

Tries to retrieve title and artist data from a CDDDB server for the CD represented by the given *toc-file*. The retrieved data is added as CD-TEXT data for language 0 to the *toc-file*. Existing CD-TEXT data for language 0 will be overwritten.

show-data

Print out all samples that would be written to the CD-R. Each line contains the sample number (starting at 0) and the decimal sample value for the left and right channel. Useful to check if the byte order of audio files is correct.

read-test

Check if all data can be read from the audio files that are defined in the *toc-file*. This will also check the communication with the slave process that is responsible for writing the audio data to the CD-recorder. Mainly used for testing.

disk-info

Shows information about the inserted CD-R. If the CD-R has an open session it will also print the start of the last and current session which is used by mkisofs to create an image for a second or higher session.

discid

Prints out CDDDB information.

msinfo

Shows information required for creating multi session disks with mkisofs. The output is meant for processing by scripts.

drive-info

Shows drive information.

unlock

Tries to unlock the recorder device after a failed write or simulation run. If you cannot eject the CD after a cdrdao run try this command.

blank

Blanks a CD-RW. The CD-RW is minimally blanked by default. Use option **--blank-mode** to select another blanking mode. Sometimes the blanking speed must be manually reduced for a successful blanking operation. Use option **--speed** to select another blanking speed.

scanbus

Scan for devices.

simulate

Like **write** but laser stays cold. It is a shortcut for **write --simulate**.

write

Write the CD-R according to the specifications in the *toc-file*.

copy

Performs all steps to copy a CD. The device containing the source CD must be specified with option **--source-device** and the recorder device with option **--device**. If only a single device is available the option **--source-device** must be omitted and cdrdao will prompt to insert the CD-R after an image of the source CD was created.

The image file with name "cddata<pid>.bin" will be created in the current working directory if no **--datafile** option is given. The created image will be removed after it has been written.

If option **--on-the-fly** is given no image file is created and the data will be directly piped from the reading device to the CD recorder.

OPTIONS**--device** [*prot:*]*bus,id,lun*

Sets the SCSI address of the CD-recorder in form of a bus/id/lun triple, e.g. '0,2,0' for the logical unit 0 of SCSI device with ID 2 on bus 0. ATAPI devices can be specified by using the prefix 'ATAPI:', e.g. 'ATAPI:0,0,0'. On some systems a device node may be specified directly, e.g. '/dev/sg0' on Linux systems. Linux 2.6 users may also try the newer ATAPI interface with the 'ATA:' prefix.

--source-device [*prot:*]*bus,id,lun*

Like above but used for the **copy** command to specify the source device.

--driver *driver-id:option-flags*

Force usage of specified driver instead of the automatically determined driver. Available driver IDs:

cdd2600, plextor, plextor-scan, generic-mmc, generic-mmc-raw, ricoh-mp6200, yamaha-cdr10x, teac-cdr55, sony-cdu920, sony-cdu948, taiyo-yuden, toshiba.

Specifying an illegal driver ID will give a list of available drivers. Option flags may be used to modify the behavior of some drivers. See **README** for details.

--source-driver *driver-id:option-flags*

Like above but used for the device specified with option **--source-device**.

--speed *value*

Set the writing speed to *value*. Default is the highest possible speed.

--blank-mode *mode*

Sets the blanking mode. Available modes are **full** and **minimal**. Please consider that the data of minimally blanked disks may be easily recovered. Use the **full** blanking mode for completely erasing all data. The default blanking mode is **minimal**.

--datafile *file*

Used for **read-toc**, **read-cd** and **copy**. Set the default data file placed in the toc-file by **read-toc**. Use "-" to indicate STDIN. For commands **read-cd** and **copy** it specifies the name of the created image file.

--read-raw

Only used for commands **read-cd** and **read-toc**. All data sectors will be written as 2352 byte blocks including the sector header and L-EC data to the image file. The track mode will be set to **MODE1_RAW** or **MODE2_RAW** in the created *toc-file*.

--read-subchan *mode*

Used by commands **read-cd**, **read-toc** and **copy**. Specifies the type of sub-channel data that is extracted from the source CD and written to the track image or copied to the destination CD. Mode may be **rw** for reading packed R-W sub-channel data (de-interleaved and error corrected) and **rw_raw** for reading raw R-W sub-channel data (not de-interleaved, not error corrected, L-EC data included in the track image). If this option is not specified no sub-channel data will be extracted.

--no-mode2-mixed

Only used for commands **read-cd** and **read-toc**. If we have **MODE2_FORM1** or **MODE2_FORM2**, don't extract it as **MODE2_FORM_MIX**. *toc-file*.

--tao-source

This option indicates to the commands *read-toc* and *read-cd* that the source CD was written in TAO mode. It will be assumed that the pre-gap length between all tracks (except between two audio tracks) is the standard 150 blocks plus the number of link blocks (usually 2). The number of link blocks can be controlled with option *--tao-source-adjust*.

Use this option only if *read-toc* or *read-cd* give error messages in the transition areas between two tracks. If you use this option with pressed CDs or CDs written in DAO mode you will get wrong results.

--tao-source-adjust *link-blocks*

Specifies the number of link blocks for tracks written in TAO mode. This option has only an effect if option *--tao-source* is given.

--fast-toc

Only used for command **read-toc**. This option suppresses the pre-gap length and index mark extraction which speeds up the read-toc process. Standard 2 second pre-gaps (but no silence!) will be placed into the toc-file. The resulting CD will sound like the source CD. Only the CD player's display will behave slightly different in the transition area between two tracks.

This option might help, too, if read-toc fails with your drive otherwise.

--buffers *buffer-count*

Specifies the number of buffers that are allocated to avoid buffer under runs. The minimal buffer count is fixed to 10, default is 32 except on FreeBSD systems, on which default is 20. Each buffer holds 1 second of audio data so that dividing *buffer-count* by the writing speed gives the maximum time for which reading of audio data may be stalled.

--multi

If this option is given the session will not be closed after the audio data is successfully written. It is possible to append another session on such disks, e.g. to create a CD-EXTRA.

--overburn

By default cdrdao will not allow one to write more data on a medium than specified by the current medium. This option allows to ignore this condition.

--eject

Eject the CD-R after writing or write simulation.

--swap

Swap the byte order of all samples that are send to the CD-recorder.

--session *session-nr*

Used for **read-toc** and **read-cd** to specify the session which should be processed on multi session CDs.

--reload

Indicates that the tray may be opened before writing without prompting the user to reset the disk status after a simulation run.

--force

Forces the execution of an operation that otherwise would not be performed.

--paranoia-mode *mode*

Sets the correction mode for digital audio extraction. 0: No checking, data is copied directly from the drive. 1: Perform overlapped reading to avoid jitter. 2: Like 1 but with additional checks of the read audio data. 3: Like 2 but with additional scratch detection and repair.

The extraction speed reduces from 0 to 3.

Default is the full paranoia mode (3).

--keepimage

If a CD is copied with command *copy* this option will cause that the created image is not removed after the copy process has finished.

--on-the-fly

Perform CD copy on the fly without creating an image file.

--with-cddb

Enables the automatic fetching of CDDb data for use as CD-TEXT data for commands *copy*, *read-toc* and *read-cd*.

--cddb-servers *server-list*

Sets space or ',' separated list of CDDb servers used for command *read-cddb* or for commands where the *--with-cddb* option is active. A server entry may have the following forms:

<server>

Connect to <server>, default cddb port (888), use cddb protocol.

<server>:<port>

Connect to <server>, port <port>, use cddb protocol.

<server>:<cgi-bin-path>

Connect to <server>, default http port (80), use http protocol, url: <cgi-bin-path>.

<server>:<port>:<cgi-bin-path>

Connect to <server>, port <port>, use http protocol, url: <cgi-bin-path>.

<server>:<port>:<cgi-bin-path>:<proxy-server>

Connect to <proxy-server>, default http port (80), use http protocol, url:
http://<server>:<port>/<cgi-bin-path>.

<server>:<port>:<cgi-bin-path>:<proxy-server>:<proxy-port>

Connect to <proxy-server>, port <proxy-port>, use http protocol, url: http://<server>:<port>/<cgi-bin-path>.

The <cgi-bin-path> is usually "/~cddb/cddb.cgi".

All servers of the server list will be tried in the given order until a successful connection can be established. For http proxy servers the first successful connected http proxy server will be used independent of the ability to connect to the target http server.

Example: freedb.freedb.org/~cddb/cddb.cgi

--cddb-timeout *timeout*

Sets the timeout in seconds used for connections to CDDB servers.

--cddb-directory *directory*

Specifies the local CDDB database directory where fetched CDDB records will be stored. If this option is not given a fetched CDDB record will not be stored locally.

--tmpdir *directory*

Specifies the directory in which to store temporary data files created from decoding MP3 and Ogg Vorbis files. By default, "/tmp" is used.

--keep

Upon exit from cdrdao, do not delete temporary WAV files created from MP3 and Ogg Vorbis files.

--save

Saves some of the current options to the settings file "\$HOME/.cdrdao" and exit. See section 'SETTINGS' for more details.

-n Suppresses the 10 second pause before writing or simulating.

-v *verbose-level*

Sets verbose level. Levels > 2 are debug levels which produce a lot of output.

TOC FILES

The *toc-file* describes what data is written to the CD-R and allows control over track/index positions, pre-gaps and sub-channel information. It is a simple text file, use your favorite text editor to create it.

A *toc-file* contains an optional header and a sequence of track specifications. Comments starting with *'//'* reaching until end of line can be placed anywhere.

Header

CATALOG "ddddddddddddd"

Specifies the optional catalog number of the CD. The string must contain exactly 13 digits.

The following flags specify the type of session that will be created. It is used to create the correct CD-TOC format and to check the consistency of the track modes for the desired session type. If multiple flags are given the last one will take effect.

CD_DA

The disc contains only audio tracks.

CD_ROM

The disc contains just mode 1 tracks or mode 1 and audio tracks (mixed mode CD).

CD_ROM_XA

The disc contains mode 2 form 1 or mode 2 form 2 tracks. Audio tracks are allowed, too. This type must be used if multi session disks are created (option *--multi*).

CD_TEXT { ... }

Defines global CD-TEXT data like the album title and the used languages. See the CD-TEXT section below for the syntax of the CD-TEXT block contents.

Track Specification

TRACK <track-mode> [<sub-channel-mode>]

Starts a new track, the track number is incremented by 1. The length of a track must be at least 4 seconds. The block length of the input data depends on the <track-mode>: AUDIO: 2352 bytes (588 samples), MODE1: 2048 bytes, MODE1_RAW: 2352 bytes, MODE2: 2336 bytes, MODE2_FORM1: 2048 bytes, MODE2_FORM2: 2324 bytes, MODE2_FORM_MIX: 2336 bytes including the sub-header, MODE2_RAW: 2352 bytes. The <sub-channel-mode> is optional. If given it specifies the type of sub-channel data for each sector. RW: packed R-W sub-channel data (96 bytes, L-EC data will be generated if required), RW_RAW: raw R-W sub-channel data (interleaved and L-EC data already calculated, 96 bytes). The block length is increased by the sub-channel data length if a <sub-channel-mode> is specified. If the input data length is not a multiple of the block length it will be padded with zeros.

The following flags may follow the track start statement. They are used to set sub-channel information for the current track. Each flag is optional. If not given the following defaults are used: copy not permitted, no pre emphasis, two channel audio, no ISRC code.

[NO] COPY

Sets or clears the copy permitted flag.

[NO] PRE_EMPHASIS

Sets or clears the pre emphasis flag (only for audio tracks).

TWO_CHANNEL_AUDIO

Indicates that track contains two channel audio data (only for audio tracks).

FOUR_CHANNEL_AUDIO

Indicates that track contains four channel audio data (only for audio tracks).

ISRC "CCOOOYYSSSS"

Sets ISRC code of track (only for audio tracks).

C: country code (upper case letters or digits)

O: owner code (upper case letters or digits)

Y: year (digits)

S: serial number (digits)

An optional CD-TEXT block that defines the CD-TEXT data for this track may follow. See the CD-TEXT section below for the syntax of the CD-TEXT block contents.

CD_TEXT { ... }

At least one of the following statements must appear to specify the data for the current track. Lengths and start positions may be expressed in samples (1/44100 seconds) for audio tracks or in bytes for data tracks. It is also possible to give the length in blocks with the MSF format 'MM:SS:FF' specifying minutes, seconds and frames ($0 \leq \text{'FF'} < 75$). A frame equals one block.

If more than one statement is used the track will be composed by concatenating the data in the specified order.

SILENCE <length>

Adds zero audio data of specified length to the current audio track. Useful to create silent pre-gaps.

ZERO <length>

Adds zero data to data tracks. Must be used to define pre- or post-gaps between tracks of different mode.

[**FILE** | **AUDIOFILE**] "<filename>" <start> [<length>]

Adds the audio data of specified file to the current audio track. It is possible to select a portion of an audio file with <start> and <length> which allows non destructive cutting. The first sample of an audio file is addressed with <start> = 0. If <length> is omitted or set to 0 all audio data from <start> until the end of file is used.

Audio files may have raw or WAVE format with 16 bits per sample, 44.1 kHz sampling rate, stereo. Raw files must have the layout 'MSBLeft LSBLeft MSBRight LSBRight ...' (big endian byte order). WAVE files are expected to have little endian byte order. The option --swap reverses the expected byte order for all raw and WAVE files. Only filenames with a ".wav" ending are treated as WAVE files, all other names are assumed to be raw audio files. Use tools like sox(1) to convert other file formats to supported formats.

Specifying a "-" as filename causes data to be read from STDIN. Currently only raw files are supported from STDIN.

If you are unsure about the byte order of your audio files try the command 'show-data'. If the byte order is correct you will see a sequence of increasing or decreasing numbers for both channels. Otherwise numbers are jumping between very high and low values - high volume static.

DATAFILE "<filename>" [<length>]

Adds data from given file to the current data track. If <length> is omitted the actual file length will be used.

FIFO "<fifo path>" <length>

Adds data from specified FIFO path to the current audio or data track. <length> must specify the amount of data that will be read from the FIFO. The value is always in terms of bytes (scalar value) or in terms of the block length (MSF value).

START [MM:SS:FF]

Defines the length of the pre-gap (position where index switches from 0 to 1). If the MSF value is omitted the current track length is used. If the current track length is not a multiple of the block length the pre-gap length will be rounded up to next block boundary.

If no START statement is given the track will not have a pre-gap.

PREGAP MM:SS:FF

This is an alternate way to specify a pre-gap with zero audio data. It may appear before the first SILENCE, ZERO or FILE statement. Either PREGAP or START can be used within a track specification. It is equivalent to the sequence

SILENCE MM:SS:FF

START

for audio tracks or

ZERO MM:SS:FF

START

for data tracks.

Nothing prevents mixing 'DATAFILE'/'ZERO' and 'AUDIOFILE'/'SILENCE' statements within the same track. The results, however, are undefined.

The end of a track specification may contain zero or more index increment statements:

INDEX MM:SS:FF

Increments the index number at given position within the track. The first statement will increment from 1 to 2. The position is relative to the real track start, not counting an existing pre-gap.

CD-TEXT Blocks

A CD-TEXT block may be placed in the global section to define data valid for the whole CD and in each track specification of a *toc-file*. The global section must define a language map that is used to map a *language-number* to country codes. Up to 8 different languages can be defined:

LANGUAGE_MAP { 0 : c1 1 : c2 ... 7 : c7 }

The country code may be an integer value in the range 0..255 or one of the following countries (the corresponding integer value is placed in braces behind the token): EN(9, English)

It is just necessary to define a mapping for the used languages.

If no mapping exists for a *language-number* the data for this language will be ignored.

For each language a language block must exist that defines the actual data for a certain language.

LANGUAGE language-number { cd-text-item cd-text-data cd-text-item cd-text-data ... }

Defines the CD-TEXT items for given *language-number* which must be defined in the language map.

The *cd-text-data* may be either a string enclosed by " or binary data like

{ 0, 10, 255, ... }

where each integer number must be in the range 0..255.

The *cd-text-item* may be one of the following:

TITLE

String data: Title of CD or track.

PERFORMER

String data.

SONGWRITER

String data.

COMPOSER

String data.

ARRANGER

String data.

MESSAGE

String data. Message to the user.

DISC_ID

String data: Should only appear in the global CD-TEXT block. The format is usually: XY12345

GENRE

Mixture of binary data (genre code) and string data. Should only appear in the global CD-TEXT

block. Useful entries will be created by *gcdmaster*.

TOC_INFO1

Binary data: Optional table of contents 1. Should only appear in the global CD-TEXT block.

TOC_INFO2

Binary data: Optional table of contents 2. Should only appear in the global CD-TEXT block.

UPC_EAN

String data: This item should only appear in the global CD-TEXT block. Was always an empty string on the CD-TEXT CDs I had access to.

ISRC

String data: ISRC code of track. The format is usually: CC-OOO-YY-SSSSS

SIZE_INFO

Binary data: Contains summary about all CD-TEXT data and should only appear in the global CD-TEXT block. The data will be automatically (re)created when the CD-TEXT data is written.

If one of the CD-TEXT items TITLE, PERFORMER, SONGWRITER, COMPOSER, ARRANGER, ISRC is defined for at least on track or in the global section it must be defined for all tracks and in the global section. If a DISC_ID item is defined in the global section, an ISRC entry must be defined for each track.

Examples

Simple track without pre-gap with all audio data from WAVE file "data.wav":

```
CD_DA
TRACK AUDIO
FILE "data.wav" 0
```

Standard track with two second pre-gap, ISRC code and CD-TEXT:

```
CD_DA
CD_TEXT {
  LANGUAGE_MAP {
    0 : EN
  }

  LANGUAGE 0 {
    TITLE "CD Title"
```

```

    PERFORMER "Performer"
    DISC_ID "XY12345"
    UPC_EAN ""
  }
}

```

```

TRACK AUDIO
ISRC "DEXXX9800001"
CD_TEXT {
  LANGUAGE 0 {
    TITLE "Track Title"
    PERFORMER "Performer"
    ISRC "DE-XXX-98-00001"
  }
}
PREGAP 0:2:0
FILE "data.wav" 0

```

Track with 10 second pre-gap containing audio data from raw file "data.cdr":

```

CD_DA
TRACK AUDIO
FILE "data.cdr" 0
START 0:10:0

```

Composed track with data from different files. Pre-gap data and length is taken from "pregapdata.wav". The first minute of "track.cdr" is omitted and two seconds silence are inserted at '2:0:0'. Index will be incremented after 2 and 4 minutes past track start:

```

CD_DA
TRACK AUDIO
FILE "pregapdata.wav" 0
START
FILE "track.cdr" 1:0:0 1:0:0
SILENCE 0:2:0
FILE "track.cdr" 2:0:0
INDEX 2:0:0
INDEX 4:0:0

```

Mixed mode CD with a data track as first track followed by two audio tracks.

```

CD_ROM
TRACK MODE1

```

```
DATAFILE "data_1"
ZERO 00:02:00 // post-gap

TRACK AUDIO
SILENCE 00:02:00 // pre-gap
START
FILE "data_2.wav" 0

TRACK AUDIO
FILE "data_3.wav" 0
```

CUE FILES

Cue files may be used wherever a *toc-file* is expected. The corresponding bin file is not taken from the FILE statement of a cue file but constructed from the cue file name by replacing ".cue" by ".bin". The cue file must have exactly one FILE statement.

Currently, following track modes are supported: MODE1/2048, MODE1/2352, MODE2/2336, MODE2/2352. The CATALOG, ISRC and POSTGAP statements are parsed but not evaluated, yet.

SETTINGS

Some of the command line options can be stored as settings at following locations. The files will be read on startup of *cdrdao* in that order:

1. /usr/local/etc/cdrdao.conf
2. \$HOME/.cdrdao

Command line options will overwrite the loaded settings. The settings file contains name - value pairs separated by a colon. String values must be enclosed by ". The file is automatically written if the command line option *--save* is used but it is also possible to modify it manually. Following values are defined:

write_device

Device used for operations *simulate*, *write*, *copy*, *blank*, *disk-info* and *unlock*. Corresponding option: *--device*

write_driver

Driver (including driver options) that is used for operations *simulate*, *write*, *copy*, *blank*, *disk-info* and *unlock*. Corresponding option: *--driver*

write_speed

Specifies writing speed. Corresponding option: *--speed*

write_buffers

Specifies fifo buffers used for recording. Corresponding option: *--buffers*

read_device

Device used for operations *read-toc*, *read-cd* and *copy*. Corresponding option: *--device* or *--source-device*

read_driver

Driver (including driver options) used for operations *read-toc*, *read-cd* and *copy*. Corresponding option: *--driver* or *--source-driver*

read_paranoia_mode

Paranoia mode used for operations *read-cd* and *copy*. Corresponding option: *--paranoia-mode*

cddb_server_list

CDDB server list for *read-cddb*. Corresponding option: *--cddb-servers*

cddb_timeout

CDDB connection timeout in seconds used by *read-cddb*. Corresponding option: *--cddb-timeout*

cddb_directory

Local directory where fetched CDDB records will be stored, used by *read-cddb*. Corresponding option: *--cddb-directory*

tmp_file_dir

Directory where temporary WAV files will be created from decoding MP3 and Ogg Vorbis files. Corresponding option: *--tmpdir*

BUGS

If the program is terminated during the write/simulation process used IPC resources may not be released. Use *ipcs(1)* and *ipcrm(1)* to delete them.

AUTHOR

Andreas Mueller mueller@daneb.ping.de [DEFUNCT]

Denis Leroy <denis@poolshark.org>

Manuel Clos <llanero@users.sourceforge.net>

SEE ALSO

cdrecord (1), cdda2wav (1), cdparanoia (1), sox (1), ipcs (1), ipcrm (1)