NAME

cdrecord − record audio or data CD, DVD or BluRay

SYNOPSIS

cdrecord [*general options*][**dev=***device*][*track options*] *track1...trackn*

DESCRIPTION

Cdrecord is used to record data or audio Compact Discs on an Orange Book CD-recorder, to write DVD media on a DVD-recorder or to write BluRay media on a BluRay-recorder.

Device naming

Most users do not need to care about device naming at all. If no **dev=** option was specified, **cdrecord** implements **auto target** support and automagically finds the drive in case that exactly one CD-ROM type drive is available in the system. In case that more than one CD-ROM type drive exists on the system, a list of possible device name parameters may be retrieved with **cdrecord −scanbus** or from the target example from the output of **cdrecord dev=help**, then the **dev=** parameter may be set based on the device listing.

The *device* parameter to the **dev=** option explained below refers to *scsibus*/*target*/*lun* of the CD/DVD/BluRay-recorder. If a file /etc/default/cdrecord exists, the parameter to the **dev=** option may also be a drive name label in said file (see FILES section).

Constraints for running cdrecord

On **SVr4** compliant systems, **cdrecord** uses the real-time class to get the highest scheduling priority that is possible (higher than all kernel processes). On systems with **POSIX real-time scheduling** cdrecord uses real-time scheduling too, but may not be able to gain a priority that is higher than all kernel processes.

In order to be able to use the SCSI transport subsystem of the OS, run at highest priority and lock itself into core **cdrecord** either needs to be run as root, needs to be installed suid root or must be called via **RBACs** pfexec mechanism.

File to track mapping

In *Track At Once* mode, each *track* corresponds to a single file that contains the prepared data for that track. If the argument is '−', standard input is used for that track. Only one track may be taken from *stdin*. In the other write modes, the direct file to track relation may not be implemented. In **−clone** mode, a single file contains all data for the whole disk. To allow DVD writing on platforms that do not implement large file support, **cdrecord** concatenates all file arguments to a single track when writing to DVD media.

GENERAL OPTIONS

General options must be before any track file name or track option.

Informative options

−help display version information for **cdrecord** on standard output.

−version

Print version information and exit.

−v Increment the level of general verbosity by one. This is used e.g. to display the progress of the writing process.

Media write mode options

−dummy

The **−dummy** option modifies the current write strategy. The CD/DVD/BluRay-recorder will go through all steps of the recording process, but the laser is turned off during this procedure. It is recommended to run several tests before actually writing to a Compact Disk or Digital

Versatile Disk, if the timing and load response of the current system is not yet known.

The −**dummy** option does not work with all media and write modes. DVD+ media and Blu-Ray media does not support **dummy** writes and most CD-recorders do not support **dummy** writes in raw mode.

−**multi** Allow multi-session CDs or multi-border DVDs to be made. This flag needs to be present on all sessions of a multi-session or multi-border disk, except you want to create a session on a CD that will be the last session on the CD-media.

For CD-media, the fixation will be done in a way that allows the CD/DVD/BluRay-recorder to append additional sessions later. This is done by generating a TOC with a link to the next program area. The so generated media is not 100% compatible to manufactured CDs (except for CDplus). Use only for recording of multi-session CDs. If this option is present, the default track type is **CD-ROM XA mode 2 form 1** and the sector size is 2048 bytes. The XA sector subheaders will be created by the drive. The *Sony* drives have no hardware support for **CD-ROM XA mode 2 form 1**. You have to specify the −**data** option in order to create multi-session disks on these drives. If you like to record a multi-session disk in SAO mode, you need to force **CD-ROM** sectors by including the −**data** option. Not all drives allow multi-session CDs in SAO mode.

For DVD media, −**multi** switches the write mode to **incremental packet recording**. There is currently no way to prevent the ability to append further sessions and there is currently only support for DVD-R/DVD-RW media. To reuse a DVD-RW that has previously been written in **incremental packet recording** mode for different write modes, you need to blank the entire media before.

−**dao**

−**sao** Set **SAO (Session At Once)** mode which is usually called **Disk At Once** mode. This currently only works with MMC drives that support **Session At Once** mode. Note that cdrecord needs to know the size of each track in advance for this mode (see the **mkisofs** −**print**−**size** option and the *EXAMPLES* section for more information).

There are several CD writers with bad firmware that result in broken disks when writing in TAO or SAO mode. If you find any problems with the layout of a disk or with subchannel content (e.g. wrong times on the display when playing the CD) and your drive supports to write in −**raw96r** or −**raw16** mode, you should give it a try.

−**tao** Set **TAO (Track At Once) writing mode.** This is the default write mode in previous **cdrecord** versions. With most drives, this write mode is required for multi-session recording.

There are several CD writers with bad firmware that result in broken disks when writing in TAO or SAO mode. If you find any problems with the layout of a disk or with subchannel content (e.g. wrong times on the display when playing the CD) and your drive supports to write in −**raw96r** or −**raw16** mode, you should give it a try.

−**raw** Set **RAW writing mode.** Using this option defaults to −**raw96r**. Note that cdrecord needs to know the size of each track in advance for this mode (see the **mkisofs** −**print**−**size** option and the *EXAMPLES* section for more information).

−**raw96r**

Set **RAW writing mode** with 2352 byte sectors plus 96 bytes of raw P-W sub-channel data resulting in a sector size of 2448 bytes. This is the preferred raw writing mode as it gives best control over the CD-writing process. Writing data disks in raw mode needs significantly more CPU time than other write modes. If your CPU is too slow, this may result in buffer underruns. Note that cdrecord needs to know the size of each track in advance for this mode (see the

mkisofs −print−size option and the *EXAMPLES* section for more information).

−raw96p

Set **RAW writing mode** with 2352 byte sectors plus 96 bytes of packed P-W sub-channel data resulting in a sector size of 2448 bytes. This is the less preferred raw writing mode as only a few recorders support it and some of these recorders have bugs in the firmware implementation. Don't use this mode if your recorder supports **−raw96r** or **−raw16**. Writing data disks in raw mode needs significantly more CPU time than other write modes. If your CPU is too slow, this may result in buffer underruns. Note that cdrecord needs to know the size of each track in advance for this mode (see the **mkisofs −print−size** option and the *EXAMPLES* section for more information).

−raw16

Set **RAW writing mode** with 2352 byte sectors plus 16 bytes of P-Q sub-channel data resulting in a sector size of 2368 bytes. If a recorder does not support **−raw96r**, this is the preferred raw writing mode. It does not allow to write *CD-Text* or *CD+Graphics* but it is the only raw writing mode in cheap CD-writers, as these cheap writers in most cases do not support **−dao** mode. Don't use this mode if your recorder supports **−raw96r**. Writing data disks in raw mode needs significantly more CPU time than other write modes. If your CPU is too slow, this may result in buffer underruns. Note that cdrecord needs to know the size of each track in advance for this mode (see the **mkisofs −print-size** option and the *EXAMPLES* section for more information).

Cdrecord functional options

−abort Try to send an **abort** sequence to the drive. If you use **cdrecord** only, this should never be needed; but other software may leave a drive in an unusable condition. Calling **cdrecord −reset** may be needed if a previous write has been interrupted and the software did not tell the drive that it will not continue to write.

−atip Retrieve and print out the ATIP (Absolute Time In Pre-groove) info of a CD/DVD/BluRay recordable or CD/DVD/BluRay re-writable media. With this option, **cdrecord** will try to retrieve the ATIP info. If the actual drive does not support to read the ATIP info, it may be that only a reduced set of information records or even nothing is displayed. Only a limited number of MMC-compliant drives support to read the ATIP info.

If **cdrecord** is able to retrieve the lead-in start time for the first session, it will try to decode and print the manufacturer info from the media. DVD media does not have ATIP information but there is equivalent prerecorded information that is read out and printed.

blank=*type*

Blank a CD-RW and exit or blank a CD-RW before writing. The blanking type may be one of:

help	Display a list of possible blanking types.
all	Blank the entire disk. This may take a long time.
fast	Minimally blank the disk. This results in erasing the PMA, the TOC and the pregap.
track	Blank the last track.
unreserve	Unreserve a reserved track.
trtail	Blank the tail of a track.
unclose	Unclose last session.
session	Blank the last session.

Not all drives support all blanking types. It may be necessary to use **blank=all** if a drive reports a specified command as being invalid. If used together with the **−force** flag, this option may be used to blank CD-RW disks that otherwise cannot be blanked. Note that you may need

to specify **blank=***all* because some drives will not continue with certain types of bad CD-RW disks. Note also that **cdrecord** does its best if the −**force** flag is used but it finally depends on the drive's firmware whether the blanking operation will succeed or not.

−**checkdrive**
Checks if a driver for the current drive is present and exit. If the drive is a known drive, **cdrecord** uses exit code 0.

−**clone** Tells **cdrecord** to handle images created by *readcd* −*clone*. The −**clone** write mode may only be used in conjunction with the −**raw96r** or −**raw16** option. Using −**clone** together with −**raw96r** is preferred as it allows to write all sub-channel data. The −**raw16** option should only be used with drives that do not support to write in −**raw96r** mode.

Note that copying in **clone** mode disables certain levels of error correction and thus always results in a quality degradation. Avoid copying audio CDs in **clone** mode for this reason.

cuefile=*filename*
Take all recording-related information from a CDRWIN-compliant **CUE sheet** file. No track-file arguments to **cdrecord** are allowed when this option is present and one of the following options: −**dao**, −**sao**, −**raw**, −**raw16**, −**raw96r** is needed in addition.

defpregap=#
Set the default pre-gap size for all tracks except track number 1. This option currently only makes sense with the TEAC drive when creating track-at-once disks without the 2-second silence before each track.
This option may go away in the future.

driver=*name*
Allows the user to manually select a driver for the device. The reason for the existence of the **driver=***name* option is to allow users to use **cdrecord** with drives that are similar to supported drives but not known directly by **cdrecord**. All drives made after 1997 should be MMC-standard-compliant and thus supported by one of the MMC drivers. It is most unlikely that **cdrecord** is unable to find the right driver automatically. Use this option with extreme care. If a wrong driver is used for a device, the possibility of creating corrupted disks is high. The minimum problem related to a wrong driver is that the −**speed** or −**dummy** will not work.

The following driver names are supported:

help To get a list of possible drivers together with a short description.

mmc_bd
The generic SCSI-3/mmc BluRay driver is auto-selected whenever **cdrecord** finds an MMC-compliant drive that does support to write BluRay media or a multi system that contains a BluRay disk as the current medium. This driver tries to close the tray, checks the medium found in the tray and then branches to the driver that matches the current medium.

mmc_bdr
The generic SCSI-3/mmc BluRay driver is auto-selected whenever **cdrecord** finds an MMC-compliant drive that does support to write BluRay BD-R media or a multi system that contains a BluRay BD-R disk as the current medium.

mmc_bdre
The generic SCSI-3/mmc BluRay driver is auto-selected whenever **cdrecord** finds an MMC-compliant drive that does support to write BluRay BD-RE media or a multi system that contains a BluRay BD-RE disk as the current medium.

mmc_cd
The generic SCSI-3/mmc CD-ROM driver is auto-selected whenever **cdrecord** finds an MMC-compliant drive that does not identify itself to support writing at all, or that only

identifies to support media or write modes not implemented in **cdrecord**.

mmc_cd_dvd
The generic SCSI-3/mmc CD/DVD/BluRay driver is auto-selected whenever **cdrecord** finds an MMC-2 or MMC-3-compliant drive that seems to support more than one medium type and the tray is open or no medium could be found to select the right driver. This driver tries to close the tray, checks the medium found in the tray and then branches to the driver that matches the current medium.

mmc_cdr
The generic SCSI-3/mmc CD-R/CD-RW driver is auto-selected whenever **cdrecord** finds an MMC-compliant drive that only supports to write CDs or a multi system drive that contains a CD as the current medium.

mmc_cdr_sony
The generic SCSI-3/mmc CD-R/CD-RW driver is auto-selected whenever **cdrecord** would otherwise select the **mmc_cdr** driver but the device seems to be made by Sony. The **mmc_cdr_sony** is definitely needed for the Sony CDU 928 as this drive does not completely implement the MMC standard and some of the MMC SCSI commands have to be replaced by Sony proprietary commands. It seems that all Sony drives (even newer ones) still implement the Sony proprietary SCSI commands so it has not yet become a problem to use this driver for all Sony drives. If you find a newer Sony drive that does not work with this driver, please report.

mmc_dvd
The generic SCSI-3/mmc-2 DVD-R/DVD-RW driver is auto-selected whenever **cdrecord** finds an MMC-2 or MMC-3-compliant drive that supports to write DVDs and an appropriate medium is loaded. There is no Track At Once mode for DVD writers.

mmc_dvdplus
The generic SCSI-3/mmc-3 DVD+R/DVD+RW driver is auto-selected whenever one of the DVD+ media types that are incompatible to each other is found. It checks media and then branches to the driver that matches the current medium.

mmc_dvdplusr
The generic SCSI-3/mmc-3 DVD+R driver is auto-selected whenever a DVD+R medium is found in an appropriate writer. Note that for unknown reason, the DVD+RW Alliance does not like that there is a simulation mode for DVD+R media. The author of **cdrecord** tries to convince manufacturers to implement a simulation mode for DVD+R and implement support. DVD+R only supports one write mode that is somewhere between Track At Once and Packet writing; this mode is selected in **cdrecord** via the **−dao**/**−sao** option.

mmc_dvdplusrw
The generic SCSI-3/mmc-3 DVD+RW driver is auto-selected whenever a DVD+RW medium is found in an appropriate writer. As DVD+RW media need to be formatted before their first use, cdrecord auto-detects this medium state and performs a format before it starts to write. Note that for unknown reason, the DVD+RW Alliance does not like that there is a simulation mode nor a way to erase DVD+RW media. DVD+RW only supports one write mode that is close to Packet writing; this mode is selected in **cdrecord** via the **−dao**/**−sao** option.

cw_7501
The driver for Matsushita/Panasonic CW-7501 is auto-selected when **cdrecord** finds this old pre-MMC drive. **Cdrecord** supports all write modes for this drive type.

kodak_pcd_600
The driver for Kodak PCD-600 is auto-selected when **cdrecord** finds this old pre-

MMC drive which has been the first high speed (6x) CD-writer for a long time. This drive behaves similarly to the Philips CDD-521 drive.

philips_cdd521
The driver for Philips CDD-521 is auto-selected when **cdrecord** finds a Philips CDD-521 drive (which is the first CD-writer ever made) or one of the other drives that are known to behave similarly to this drive. All Philips CDD-521 or similar drives (see other drivers in this list) do not support Session At Once recording.

philips_cdd521_old
The driver for Philips old CDD-521 is auto-selected when **cdrecord** finds a Philips CDD-521 with very old firmware which has some known limitations.

philips_cdd522
The driver for Philips CDD-522 is auto-selected when **cdrecord** finds a Philips CDD-522 which is the successor of the 521 or one of its variants with Kodak label. **Cdrecord** does not support Session At Once recording with these drives.

philips_dumb
The driver for Philips CDD-521 with pessimistic assumptions is never auto-selected. It may be used by hand with drives that behave similarly to the Philips CDD-521.

pioneer_dws114x
The driver for Pioneer DW-S114X is auto-selected when **cdrecord** finds one of the old non-MMC CD-writers from Pioneer.

plasmon_rf4100
The driver for Plasmon RF 4100 is auto-selected when **cdrecord** finds this specific variant of the Philips CDD-521.

ricoh_ro1060c
The driver for Ricoh RO-1060C is auto-selected when **cdrecord** finds this drive. There is no real support for this drive yet.

ricoh_ro1420c
The driver for Ricoh RO-1420C is auto-selected when **cdrecord** finds a drive with this specific variant of the Philips CDD-521 command set.

scsi2_cd
The generic SCSI-2 CD-ROM driver is auto-selected whenever **cdrecord** finds a pre-MMC drive that does not support writing or a pre-MMC writer that is not supported by **cdrecord**.

sony_cdu924
The driver for Sony CDU-924 / CDU-948 is auto-selected whenever **cdrecord** finds one of the old pre-MMC CD-writers from Sony.

teac_cdr50
The driver for Teac CD-R50S, Teac CD-R55S, JVC XR-W2010, Pinnacle RCD-5020 is auto-selected whenever one of the drives is found that is known to use the non-MMC command set used by TEAC and JVC. Note that many drives from JVC will not work because they do not correctly implement the documented command set and JVC has been unwilling to fix or document the bugs. There is no support for the Session At Once write mode yet.

tyuden_ew50
The driver for Taiyo Yuden EW-50 is auto-selected when **cdrecord** finds a drive with this specific variant of the Philips CDD-521 command set.

yamaha_cdr100
The driver for Yamaha CDR-100 / CDR-102 is auto-selected when **cdrecord** finds one

of the old pre-MMC CD-writers from Yamaha. There is no support for the Session At Once write mode yet.

bd_simul
The simulation BluRay driver allows to run timing and speed tests with parameters that match the behavior of BluRay writers.

cdr_simul
The simulation CD-R driver allows to run timing and speed tests with parameters that match the behavior of CD-writers.

dvd_simul
The simulation DVD-R driver allows to run timing and speed tests with parameters that match the behavior of DVD writers.

There are two special driver entries in the list: **cdr_simul** and **dvd_simul**. These driver entries are designed to make timing tests at any speed or timing tests for drives that do not support the **−dummy** option. The simulation drivers implement a drive with a buffer size of 1 MB that can be changed via the **CDR_SIMUL_BUFSIZE** environment variable. The simulation driver correctly simulates even a buffer underrun condition. If the **−dummy** option is present, the simulation is not aborted in case of a buffer underrun.

driveropts=*option list*
Set driver specific options. The options are specified as a comma separated list. To get a list of valid options use **driveropts=***help* together with the *−checkdrive* option. If you like to set driver options without running a typical **cdrecord** task, you need to use the **−setdropts** option in addition, otherwise the command line parser in **cdrecord** will complain. Currently implemented driver options are:

burnfree
Turn the support for Buffer Underrun Free writing on. This only works for drives that support Buffer Underrun Free technology. This may be called: **Sanyo BURN-Proof**, **Ricoh Just-Link**, **Yamaha Lossless-Link** or similar.

The default is to turn **BURN-Free** off, regardless of the defaults of the drive.

noburnfree
Turn the support for Buffer Underrun Free writing off.

varirec=*value*
Turn on the **Plextor VariRec** writing mode. The mandatory parameter *value* is the laser power offset and currently may be selected from −2, −1, 0, 1, 2. In addition, you need to set the write speed to 4 in order to allow **VariRec** to work.

gigarec=*value*
Manage the **Plextor GigaRec** writing mode. The mandatory parameter *value* is the disk capacity ratio compared to normal recording and currently may be selected from 0.6, 0.7, 0.8, 0.9, 1.0, 1,1, 1.2, 1.3, 1.4. If values < 1.0 are used, then the effect is similar to the **Yamaha Audio Master Q. R.** feature. If values > 1.0 are used, then the disk capacity is increased.

Not all drives support all **GigaRec** values. When a drive uses the **GigaRec** feature, the write speed is limited to 8x.

audiomaster
Turn on the **Yamaha Audio Master Q. R.** feature which usually should result in high quality CDs that have less reading problems in Hi-Fi players. As this is implemented as a variant of the Session At Once write mode, it will only work if you select SAO write mode and there is no need to turn it off. The **Audio Master** mode will work

with a limited speed but may also be used with data CDs. In **Audio Master** mode, the pits on the CD will be written larger than usual so the capacity of the medium is reduced when turning this feature on. A 74-minute CD will only have a capacity of 63 minutes if **Audio Master** is active and the capacity of a 80-minute CD will be reduced to 68 minutes, the capacity in will be reduced to 85% of the original capacity. On newer Plextor drives, this feature is also present but the capacity will be reduced to 86.66% of the original capacity. For other factors on Plextor drives, see the **gigarec** option above.

forcespeed
Normally, modern drives know the highest possible speed for different media and may reduce the speed in order to grant best write quality. This technology may be called: **Plextor PowerRec**, **Ricoh Just-Speed**, **Yamaha Optimum Write Speed Control** or similar. Some drives (e.g. Plextor, Ricoh and Yamaha) allow to force the drive to use the selected speed even if the medium is so bad that the write quality would be poor. This option tells such a drive to force to use the selected speed regardless of the medium quality.

Use this option with extreme care and note that the drive should know better which medium will work at full speed. The default is to turn **forcespeed** off, regardless of the defaults of the drive.

noforcespeed
Turn off the **force speed** feature.

speedread
Some ultra high speed drives such as 48x and faster drives from Plextor limit the read speed for unknown media to e.g. 40x in order to avoid damaged disks and drives. Using this option tells the drive to read any media as fast as possible. Be very careful as this may cause the media to break in the drive while reading, resulting in damaged media and drive!

nospeedread
Turn off unlimited read speed.

singlesession
Turn the drive into a single-session only drive. This allows to read defective or non-compliant (illegal) media with extremely non-standard additional (broken/illegal) TOC entries in the TOC from the second or higher session. Some of these disks become usable if only the information from the first session is used. You need to enable Single Session mode before you insert the defective disk!

nosinglesession
Turn off single-session mode. The drive will again behave as usual.

hidecdr Hide the fact that a medium might be a recordable medium. This allows to make CD-Rs look like CD-ROMs and applications believe that the media in the drive is not a CD-R.

nohidecdr
Turn off hiding CD-R media.

tattooinfo
Use this option together with −**checkdrive** to retrieve the image size information for the **Yamaha DiskT@2** feature. The images always have a line length of 3744 pixels. Line number 0 (radius 0) is mapped to the center of the disk. If you know the inner and outer radii you will be able to create a pre distorted image that later may appear undistorted on the disk.

tattoofile=*name*

Use this option together with −**checkdrive** to write an image prepared for the **Yamaha DiskT@2** feature to the medium. The file must be a file with raw image B&W data (one byte per pixel) in a size as retrieved by a previous call to **tattooinfo**. If the size of the image equals the maximum possible size (3744 x 320 pixels), **cdrecord** will use the first part of the file. This first part then will be written to the leftover space on the CD.

Note that the image must be mirrored to be readable from the pick up side of the CD.

layerbreak

Switch a drive with DVD-R/DL medium into **layer jump recording** recording mode and use automatic layer-break position setup.

By default, DVD-R/DL media is written in **sequential recording** mode that completely fills up both layers.

layerbreak=*value*

Set up a manual layer-break value for DVD-R/DL and DVD+R/DL. The specified layer-break value must not be set to less than half of the recorded data size and must not be set to more than the remaining *Layer 0 size* of the medium. The manual layer-break value needs to be a multiple of the ECC sector size which is 16 logical 2048 byte sectors in case of DVD media and 32 logical 2048 byte sectors in case of HD-DVD or BD media.

Cdrecord does not allow to write DL media in case that the total amount of data is less then the *Layer 0 size* of the medium except when a manual layer-break has been specified by using the **layerbreak=***value* option.

−**eject** Eject disk after doing the work. Some devices (e.g. Philips) need to eject the medium before creating a new disk. Doing a −dummy test and immediately creating a real disk would not work on these devices.

−**fix** The disk will only be fixated (i.e. a TOC for a CD-reader will be written). This may be used, if for some reason the disk has been written but not fixated. This option currently does not work with old TEAC drives (CD-R50S and CD-R55S).

−**force** Force to continue on some errors. Be careful when using this option. **Cdrecord** implements several checks that prevent you from doing unwanted things like damaging CD-RW media by improper drives. Many of the sanity checks are disabled when the −**force** option is used.

This option also implements some tricks that will allow you to blank bad CD-RW disks.

−**format**

Format a CD-RW/DVD-RW/DVD+RW/BD-RE disc. Formatting is currently only implemented for DVD+RW and BD-RE media. A 'maiden' DVD+RW or BD-RE medium needs to be formatted before you may write to it. However, as **cdrecord** autodetects the need for formatting in this case and auto formats the medium before it starts writing, the −**format** option is only needed if you like to forcibly reformat a DVD+RW or BD-RE medium.

fs=# Set the FIFO (ring buffer) size to #. You may use the same syntax as in **dd**(1), **sdd**(1) or **star**(1). The number representing the size is taken in bytes unless otherwise specified. If a number is followed directly by the letter 'b', 'k', 'm', 's' or 'f', the size is multiplied by 512, 1024, 1024∗1024, 2048 or 2352. If the size consists of numbers separated by 'x' or '∗', multiplication of the two numbers is performed. Thus *fs=10x63k* will specify a FIFO size of 630 kBytes.

The size specified by the *fs=* argument includes the shared memory that is needed for

administration. This is at least one page of memory. If no *fs=* option is present, **cdrecord** will try to get the FIFO size value from the **CDR_FIFOSIZE** environment. The default FIFO size is currently 4 MB.

The FIFO is used to increase buffering for the real-time writing process. It allows to run a pipe from **mkisofs** directly into **cdrecord**. If the FIFO is active and a pipe from **mkisofs** into **cdrecord** is used to create a CD, **cdrecord** will abort prior to do any modifications on the disk if **mkisofs** dies before writing starts. The recommended FIFO size is between 4 and 128 MBytes. As a rule of thumb, the FIFO size should be at least equal to the size of the internal buffer of the CD/DVD/BluRay-recorder and no more than half of the physical amount of RAM available in the machine. If the FIFO size is big enough, the FIFO statistics will print a FIFO empty count of zero and a FIFO min fill not below 20%. It is not wise to use too much space for the FIFO. If you need more than 8 MB to write a CD at a speed less than 20x from an image on a local file system on an idle machine, your machine is either underpowered, has hardware problems or is mis-configured. If you like to write DVDs or to write CDs at higher speed, it makes sense to use at least 16 MB for the FIFO.

On old and small machines, you need to be more careful with the FIFO size. If your machine has less than 256 MB of physical RAM, you should not set up a FIFO size that is more than 32 MB. The sun4c architecture (e.g. a Sparcstation-2) has only MMU page table entries for 16 MBytes per process. Using more than 14 MBytes for the FIFO may cause the operating system in this case to spend much time to constantly reload the MMU tables. Newer machines from Sun do not have this MMU hardware problem. The author has no information on PC hardware reflecting this problem.

Old Linux systems for non-x86 platforms have broken definitions for the shared memory size. You need to fix them and rebuild the kernel or manually tell **cdrecord** to use a smaller FIFO.

If you have buffer underruns or similar problems (like a constantly empty drive-buffer) and observe a zero *fifo empty count*, you have hardware problems that prevent the data from flowing fast enough from the kernel memory to the drive. The FIFO size in this case is sufficient, but you should check for a working DMA setup.

gracetime=#
Set the grace time before starting to write to # seconds. Values below 3 seconds are not allowed in order to prevent the volume management from interrupting the write process.

−ignsize
Ignore the known size of the medium. This option should be used with extreme care, it exists only for debugging purposes so do not use it for other reasons. It is not needed to write disks with more than the nominal capacity. This option implies **−overburn**.

−immed
Tell cdrecord to set the **SCSI IMMED** flag in certain commands (load, eject, blank, close_track, close_session). This can be useful on broken systems with ATAPI hard-disk and CD/DVD/BluRay writer on the same bus or with SCSI systems that do not use disconnect/reconnect. These systems will freeze while blanking or fixating a CD/DVD/BluRay or while a DVD writer is filling up a session to the minimum amount (approx. 800 MB). Setting the **−immed** flag will request the command to return immediately while the operation proceeds in background, making the bus usable for the other devices and avoiding the system freeze. This is an experimental feature which may work or not, depending on the model of the CD/DVD/BluRay writer. A correct solution would be to set up a correct cabling but there seem to be notebooks around that have been set up the wrong way by the manufacturer. As it is impossible to fix this problem in notebooks, the **−immed** option has been added.

A second experimental feature of the **−immed** flag is to tell cdrecord to try to wait short times while writing to the media. This is expected to free the IDE bus if the CD/DVD/BluRay writer and the data source are connected to the same IDE cable. In this case, the CD/DVD/BluRay writer would otherwise usually block the IDE bus for nearly all the time making it impossible to fetch data from the source drive. See also the **minbuf=** and **−v** options.

Use both features at your own risk. If it turns out that it would make sense to have a separate option for the wait feature, write to the author and convince him.

−inq Do an inquiry for the drive, print the inquiry info for the drive and exit.

−load Load the media and exit. This only works with a tray-loading mechanism but seems to be useful when using the Kodak disk transporter.

−lock Load the media, lock the door and exit. This only works with a tray-loading mechanism but seems to be useful when using the Kodak disk transporter.

mcn=*med_cat_nr*
Set the **Media Catalog Number** of the CD to *med_cat_nr*.

minbuf=*value*
The **minbuf=** option allows to define the minimum drive-buffer fill ratio for the experimental ATAPI wait mode that is intended to free the IDE bus to allow hard disk and CD/DVD/BluRay writer to be on the same IDE cable. As the wait mode currently only works when the verbose option **−v** has been specified, **cdrecord** implies the verbose option in case the **−immed** or **minbuf=** option has been specified. Valid values for **minbuf=** are between 25 and 95 for 25%...95% minimum drive-buffer fill ratio.

−media−info

−minfo Retrieve and print information about the state of the medium. This option currently only works for MMC-compliant drives.

−msinfo
Retrieve multi-session info in a form suitable for **mkisofs-1.10** or later.

This option makes only sense with a CD that contains at least one closed session and is appendable (not finally closed yet). Some drives create error messages if you try to get the multi-session info for a disk that is not suitable for this operation.

−noclose
Do not close the current track, useful only when in packet writing mode. This is an experimental interface.

−nofix Do not fixate the disk after writing the tracks. This may be used to create an audio disk in steps. An un-fixated disk can usually not be used on a non CD-writer type drive but there are audio CD-players that will be able to play such a disk.

−overburn
Allow **cdrecord** to write more than the official size of a medium. This feature is usually called *overburning* and depends on the fact that most blank media may hold more space than the official size. As the official size of the lead-out area on the disk is 90 seconds (6750 sectors) and a disk usually works if there are at least 150 sectors of lead out, all media may be overburned by at least 88 seconds (6600 sectors). Most CD-recorders only do overburning in **SAO** or **RAW** mode. Known exceptions are TEAC CD-R50S, TEAC CD-R55S and the Panasonic CW-7502. Some drives do not allow to overburn as much as you might like and limit the size of a CD to e.g. 76 minutes. This problem may be circumvented by writing the CD in RAW mode because this way the drive has no chance to find the size before starting to burn. There is no guarantee that your drive supports overburning at all. Make a test to check if your drive implements the feature.

−**packet**
Set **Packet writing mode.** This is an experimental interface.

pktsize=#
Set the packet size to #, forces fixed packet mode. This is an experimental interface.

−**prcap** Print the drive capabilities for SCSI-3/mmc-compliant drives as obtained from mode page 0x2A. Values marked with *kB* use 1000 bytes as kilo-byte, values marked with *KB* use 1024 bytes as Kilo-byte.

−**setdropts**
Set the driveropts specified by **driveropts=***option list***,** the **speed** of the drive and the **dummy** flag and exit. This allows cdrecord to set drive specific parameters that are not directly used by **cdrecord** like e.g. **single session mode**, **hide cdr** and similar. It is needed in case that **driveropts=***option list* should be called without planning to run a typical **cdrecord** task.

speed=#
Set the speed factor of the writing process to #. # is an integer, representing a multiple of what has been defined as single speed for the medium.

For CD-media, single speed is the audio playback speed. This is about 150 KB/s for CD-ROM and about 172 KB/s for CD-Audio. Single speed is about 1385 kB/s for DVD media and about 4496 kB/s for BluRay media.

If no *speed* option is present, **cdrecord** will try to get a drive specific speed value from the file **/etc/default/cdrecord** and if it cannot find one, it will try to get the speed value from the **CDR_SPEED** environment and later from the **CDR_SPEED=** entry in **/etc/default/cdrecord**. If no speed value could be found, cdrecord uses a drive specific default speed. The default for all new (MMC-compliant) drives is to use the maximum supported by the drive. If you use *speed=0* with a MMC-compliant drive, **cdrecord** will switch to the lowest possible speed for drive and medium. If you are using an old (non-MMC) drive that has problems with *speed=2* or *speed=4*, you should try *speed=0*.

−**text** Write CD-Text information based on information taken from a file that contains ascii information for the text strings. **Cdrecord** supports CD-Text information based on the content of the ∗**.inf** files created by **cdda2wav** and CD-Text information based on the content from a **CUE sheet** file. If a **CUE sheet** file contains both (binary CDTEXTFILE and text based SONGWRITER) entries, then the information based on the CDTEXTFILE entry will win.

You need to use the −**useinfo** option in addition in order to tell **cdrecord** to read the ∗**.inf** files or **cuefile=***filename* in order to tell **cdrecord** to read a **CUE sheet** file in addition. If you like to write your own CD-Text information, edit the ∗**.inf** files or the **CUE sheet** file with a text editor and change the fields that are relevant for CD-Text.

textfile=*filename*
Write CD-Text based on information found in the binary file *filename*. This file must contain information in a data format defined in the SCSI-3 MMC-2 standard and in the Red Book. The four-byte-sized header that is defined in the SCSI standard is optional and allows to make the recognition of correct data less ambiguous. This is the best option to be used to copy CD-Text data from existing CDs that already carry CD-Text information. To get data in a format suitable for this option use **cdrecord** −**vv** −**toc** to extract the information from disk. If both, **textfile=***filename* and CD-Text information from ∗**.inf** or ∗**.cue** files are present, **textfile=***filename* will overwrite the other information.

−**toc** Retrieve and print out the table of contents or PMA of a CD. With this option, **cdrecord** will work with CD-R drives and with CD-ROM drives.

−**waiti** Wait for input to become available on standard input before trying to open the SCSI driver.

This allows **cdrecord** to read its input from a pipe even when writing additional sessions to a multi-session disk. When writing another session to a multi-session disk, **mkisofs** needs to read the old session from the device before writing output. This cannot be done if **cdrecord** opens the SCSI driver at the same time.

−useinfo

Use ∗**.inf** files to overwrite audio options. If this option is used, the pregap size information, the index information, the pre-emphasis information and the CD-Text information is read from the ∗**.inf** file that is associated with the file that contains the audio data for a track.

If used together with the −**audio** option, **cdrecord** may be used to write audio CDs from a pipe from **cdda2wav** if you call **cdrecord** with the ∗**.inf** files as track parameter list instead of using audio files. The audio data is read from **stdin** in this case. See *EXAMPLES* section below. **Cdrecord** first verifies that **stdin** is not connected to a terminal and runs some heuristic consistency checks on the ∗**.inf** files and then sets the track lengths from the information in the ∗**.inf** files.

If you like to write from **stdin**, make sure that cdrecord is called with a large enough FIFO size, reduce the write speed to a value below the read speed of the source drive and switch the burn-free option for the recording drive on.

SCSI options

dev=*target*

Set the SCSI target for the CD/DVD/BluRay-recorder, see notes above. A typical target device specification is **dev=***1,6,0* . If a filename must be provided together with the numerical target specification, the filename is implementation specific. The correct filename in this case can be found in the system specific manuals of the target operating system. On a *FreeBSD* system without *CAM* support, you need to use the control device (e.g. */dev/rcd0.ctl*). A correct device specification in this case may be **dev=***/dev/rcd0.ctl:@* .

General SCSI addressing

The *target device* to the **dev=** option refers to *scsibus*/*target*/*lun* of the CD/DVD/BluRay-recorder. Communication on *SunOS* is done with the SCSI general driver **scg.** Other operating systems are using a library simulation of this driver. Possible syntax is: **dev=** *scsibus*,*target*,*lun* or **dev=** *target*,*lun*. In the latter case, the CD/DVD/BluRay-recorder has to be connected to the default SCSI bus of the machine. *Scsibus*, *target* and *lun* are integer numbers. Some operating systems or SCSI transport implementations may require to specify a filename in addition. In this case the correct syntax for the device is: **dev=** *devicename*:*scsibus*,*target*, or **dev=** *devicename*:*target*,*lun*. If the name of the device node that has been specified on such a system refers to exactly one SCSI device, a shorthand in the form **dev=** *devicename*:@ or **dev=** *devicename*:@,*lun* may be used instead of **dev=** *devicename*:*scsibus*,*target*,

Remote SCSI addressing

To access remote SCSI devices, you need to prepend the SCSI device name by a remote device indicator. The remote device indicator is either **REMOTE:***user@host:* or **REMOTE:***host:* A valid remote SCSI device name may be: **REMOTE:***user@host:* to allow remote SCSI bus scanning or **REMOTE:***user@host:1,0,0* to access the SCSI device at *host* connected to SCSI bus # 1,target 0, lun 0. In order to allow remote access to a specific *host*, the **rscsi**(1) program needs to be present and configured on the *host*.

Alternate SCSI transports

Cdrecord is completely based on **SCSI** commands but this is no problem as all CD/DVD/BluRay writers ever made use **SCSI** commands for the communication. Even **ATAPI**

drives are just **SCSI** drives that inherently use the *ATA packet interface* as **SCSI** command transport layer build into the IDE (ATA) transport. You may need to specify an alternate transport layer on the command line if your OS does not implement a fully integrated kernel driver subsystem that allows to access any drive using **SCSI** commands via a single unique user interface.

To access SCSI devices via alternate transport layers, you need to prepend the SCSI device name by a transport layer indicator. The transport layer indicator may be something like **USCSI:** or **ATAPI:**. To get a list of supported transport layers for your platform, use **dev=** *HELP*:

Portability Background

To make **cdrecord** portable to all UNIX platforms, the syntax **dev=** *devicename*:*scsibus*,*target*, is preferred as it hides OS specific knowledge about device names from the user. A specific OS may not necessarily support a way to specify a real device file name nor a way to specify *scsibus*,*target*,*lun*.

Scsibus 0 is the default SCSI bus on the machine. Watch the boot messages for more information or look into **/var/adm/messages** for more information about the SCSI configuration of your machine. If you have problems to figure out what values for *scsibus*,*target*,*lun* should be used, try the −**scanbus** option of **cdrecord** described below.

Using logical names for devices

If no *dev* option is present, **cdrecord** will try to get the device from the **CDR_DEVICE** environment.

If a file /etc/default/cdrecord exists, and if the argument to the **dev=** option or the **CDR_DEVICE** environment does not contain the characters ',', '/', '@' or ':', it is interpreted as a device label name that was defined in the file /etc/default/cdrecord (see FILES section).

Autotarget Mode

If no **dev=** option and no **CDR_DEVICE** environment is present, or if it only contains a transport specifier but no address notation, **cdrecord** tries to scan the SCSI address space for CD-ROM drives. If exactly one is found, this is used by default.

debug=#, −**d**

Set the misc debug value to # (with debug=#) or increment the misc debug level by one (with −d). If you specify −*dd*, this equals to **debug=***2*. This may help to find problems while opening a driver for libscg as well as with sector sizes and sector types. Using −**debug** slows down the process and may be the reason for a buffer underrun.

kdebug=#, **kd=**#

Tell the **scg**-driver to modify the kernel debug value while SCSI commands are running.

−**reset** Try to reset the SCSI bus where the CD-recorder is located. This does not work on all operating systems.

−**scanbus**

Scan all SCSI devices on all SCSI busses and print the inquiry strings. This option may be used to find SCSI address of the CD/DVD/BluRay-recorder on a system. The numbers printed out as labels are computed by: **bus * 100 + target**

−**silent**, −**s**

Do not print out a status report for failed SCSI commands.

timeout=#

Set the default SCSI command timeout value to # seconds. The default SCSI command

timeout is the minimum timeout used for sending SCSI commands. If a SCSI command fails due to a timeout, you may try to raise the default SCSI command timeout above the timeout value of the failed command. If the command runs correctly with a raised command timeout, please report the better timeout value and the corresponding command to the author of the program. If no **timeout=** option is present, a default timeout of 40 seconds is used.

ts=# Set the maximum transfer size for a single SCSI command to #. The syntax for the **ts=** option is the same as for cdrecord fs=# or sdd bs=#.

If no **ts=** option has been specified, **cdrecord** defaults to a transfer size of 63 kB. If libscg gets lower values from the operating system, the value is reduced to the maximum value that is possible with the current operating system. Sometimes, it may help to further reduce the transfer size or to enhance it, but note that it may take a long time to find a better value by experimenting with the **ts=** option.

−**V** Increment the verbose level in respect of SCSI command transport by one. This helps to debug problems during the writing process, that occur in the CD/DVD/BluRay-recorder. If you get incomprehensible error messages you should use this flag to get more detailed output. −**VV** will show data buffer content in addition. Using −**V** or −**VV** slows down the process and may be the reason for a buffer underrun.

TRACK OPTIONS

Track options may be mixed with track file names.

−**audio** If this flag is present, all subsequent tracks are written in **CD-DA** (similar to Red Book) audio format. The file with data for this tracks should contain stereo, 16-bit digital audio with 44100 samples/s. The byte order should be the following: MSB left, LSB left, MSB right, LSB right, MSB left and so on. The track should be a multiple of 2352 bytes. It is not possible to put the master image of an audio track on a raw disk because data will be read in multiple of 2352 bytes during the recording process.

If a filename ends in *.au* or *.wav* the file is considered to be a structured audio data file. **Cdrecord** assumes that the file in this case is a Sun audio file or a Microsoft .WAV file and extracts the audio data from the files by skipping over the non-audio header information. In all other cases, cdrecord will only work correctly if the audio data stream does not have any header. Because many structured audio files do not have an integral number of blocks (1/75th second each) in length, it is often necessary to specify the −**pad** option as well. **cdrecord** recognizes that audio data in a .WAV file is stored in Intel (little-endian) byte order, and will automatically byte-swap the data if the CD-recorder requires big-endian data. **Cdrecord** will reject any audio file that does not match the Red Book requirements of 16-bit stereo samples in PCM coding at 44100 samples/second.

Using other structured audio data formats as input to **cdrecord** will usually work if the structure of the data is the structure described above (raw pcm data in big-endian byte order). However, if the data format includes a header, you will hear a click at the start of the track.

If neither −*data* nor −*audio* have been specified, **cdrecord** defaults to −*audio* for all filenames that end in *.au* or *.wav* and to −*data* for all other files.

−**cdi** If this flag is present, the TOC type for the disk is set to **CDI**. This only makes sense with XA disks.

−**copy** If this flag is present, all TOC entries for subsequent audio tracks of the resulting CD will indicate that the audio data has permission to be copied without limit. This option has no effect on data tracks.

−**data** If this flag is present, all subsequent tracks are written in **CD-ROM mode 1** (Yellow Book)

format. The data size is a multiple of 2048 bytes. The file with track data should contain an **ISO-9660** or **Rock Ridge** filesystem image (see **mkisofs** for more details). If the track data is an **ufs** filesystem image, fragment size should be set to 2 KB or more to allow CD-drives with 2 KB sector size to be used for reading.

−*data* is the default, if no other flag is present and the file does not appear to be of one of the well known audio file types.

If neither −*data* nor −*audio* have been specified, **cdrecord** defaults to −*audio* for all filenames that end in *.au* or *.wav* and to −*data* for all other files.

index=*list*

Sets an index list for the next track. In index list is a comma separated list of numbers that are counting from index 1. The first entry in this list must contain a 0, the following numbers must be an ascending list of numbers (counting in 1/75 seconds) that represent the start of the indices. An index list in the form: 0,7500,15000 sets index 1 to the start of the track, index 2 100 seconds from the start of the track and index 3 200 seconds from the start of the track.

−**isosize**

Use the **ISO-9660** file system size as the size of the next track. This option is needed if you want **cdrecord** to directly read the image of a track from a raw disk partition or from a *TAO* master CD. In the first case the option −**isosize** is needed to limit the size of the CD to the size of the ISO filesystem. In the second case the option −**isosize** is needed to prevent **cdrecord** from reading the two run-out blocks that are appended by each CD-recorder in track-at-once mode. These two run-out blocks cannot be read and would cause a buffer underrun that would cause a defective copy.

Note that if this option is used on files created by **mkisofs**, the padding data that was added by **mkisofs** is lost and replaced by padding added by cdrecord. This may also change the amount of padding.

In case **cdrecord** reads the track data from *stdin*, only the first track may be used with the −**isosize** option.

If −**isosize** is used for a track, **cdrecord** will automatically add padding for this track as if the −**pad** option had been used but the amount of padding may be less than the padding written by **mkisofs**. Note that if you use −**isosize** on a track that contains Sparc boot information, the boot information will be lost.

Note also that this option cannot be used to determine the size of a file system if the −**multi** option is present.

isrc=*ISRC_number*

Set the **International Standard Recording Number** for the next track to *ISRC_number*.

−**mode2**

If this flag is present, all subsequent tracks are written in **CD-ROM mode 2** format. The data size is a multiple of 2336 bytes.

−**nocopy**

If this flag is present, all TOC entries for subsequent audio tracks of the resulting CD will indicate that the audio data has permission to be copied only once for personal use − this is the default.

−**nopad**

Do not pad the following tracks − the default.

−**nopreemp**

If this flag is present, all TOC entries for subsequent audio tracks will indicate that the audio

data has been mastered with linear data − this is the default.

−**noshorttrack**
Re-enforce the Red Book track length standard. Tracks must be at least 4 seconds.

−**pad** If the track is a data track, 15 sectors of zeroed data will be added to the end of this and each subsequent data track. In this case, the −**pad** option is superseded by the **padsize=** option. It will remain however as a shorthand for **padsize=***15s*. If the −*pad* option refers to an audio track, **cdrecord** will pad the audio data to be a multiple of 2352 bytes. The audio data padding is done with binary zeroes which is equal to absolute silence.

−**pad** remains valid until disabled by −**nopad**.

padsize=#
Set the amount of data to be appended as padding to the next track to #. Opposed to the behavior of the −**pad** option, the value for *padsize=* is reset to zero for each new track. Cdrecord assumes a sector size of 2048 bytes for the *padsize=* option, independent from the real sector size and independent from the write mode. The megabytes mentioned in the verbose mode output however are counting the output sector size which is e.g. 2448 bytes when writing in RAW/RAW96 mode. See the **fs=** option for possible arguments. To pad the equivalent of 20 minutes on a CD, you may write **padsize=**20x60x75s. Use this option if your CD-drive is not able to read the last sectors of a track or if you want to be able to read the CD on a **Linux** system with the ISO-9660 filesystem read-ahead bug. If an empty file is used for track data, this option may be used to create a disk that is entirely made of padding. This may e.g. be used to find out how much overburning is possible with a specific medium.

−**preemp**
If this flag is present, all TOC entries for subsequent audio tracks will indicate that the audio data has been sampled with 50/15 µsec pre-emphasis. The data however is not modified during the process of transferring from file to disk. This option has no effect on data tracks.

pregap=#
Set the pre-gap size for the next track. This option currently only makes sense with the TEAC drive when creating track-at-once disks without the 2-second silence before each track.
This option may go away in the future.

−**scms** If this flag is present, all TOC entries for subsequent audio tracks of the resulting CD will indicate that the audio data has no permission to be copied anymore.

−**shorttrack**
Allow all subsequent tracks to violate the Red Book track length standard which requires a minimum track length of 4 seconds. This option is only useful when used in SAO or RAW mode. Not all drives support this feature. The drive must accept the resulting CUE sheet or support RAW writing.

−**swab** If this flag is present, audio data is assumed to be in byte-swapped (little-endian) order. Some types of CD-writers e.g. Yamaha, Sony and the new SCSI-3/mmc drives require audio data to be presented in little-endian order, while other writers require audio data to be presented in the big-endian (network) byte order normally used by the SCSI protocol. **Cdrecord** knows if a CD-recorder needs audio data in big- or little-endian order, and corrects the byte order of the data stream to match the needs of the recorder. You only need the −*swab* flag if your data stream is in Intel (little-endian) byte order.

Note that the verbose output of **cdrecord** will show you if swapping is necessary to make the byte order of the input data fit the required byte order of the recorder. **Cdrecord** will not show you if the −*swab* flag was actually present for a track.

tsize=# If the master image for the next track has been stored on a raw disk, use this option to specify the valid amount of data on this disk. If the image of the next track is stored in a regular file,

the size of that file is taken to determine the length of this track. If the track contains an ISO-9660 filesystem image use the −*isosize* option to determine the length of that filesystem image. In Disk At Once mode and with some drives that use the TEAC programming interface, even in Track At Once mode, **cdrecord** needs to know the size of each track before starting to write the disk. Cdrecord now checks this and aborts before starting to write. If this happens you will need to run **mkisofs −print−size** before and use the output (with 's' appended) as an argument to the **tsize**= option of **cdrecord** (e.g. tsize=250000s).
See **fs**= option for possible arguments.

−xa If this flag is present, all subsequent tracks are written in **CD-ROM XA mode 2 form 1** format. The data size is a multiple of 2048 bytes. The XA sector sub-headers will be created by the drive. With this option, the write mode is the same as with the **−multi** option.

−xa1 If this flag is present, all subsequent tracks are written in **CD-ROM XA mode 2 form 1** format. The data size is a multiple of 2056 bytes. The XA sector sub-headers are part of the user data and have to be supplied by the application that prepares the data to be written.

−xa2 If this flag is present, all subsequent tracks are written in **CD-ROM XA mode 2 form 2** format. The data is a multiple of 2324 bytes. The XA sector sub-headers will be created by the drive.

−xamix If this flag is present, all subsequent tracks are written in a way that allows a mix of **CD-ROM XA mode 2 form 1/2** format. The data size is a multiple of 2332 bytes. The XA sector sub-headers are part of the user data and have to be supplied by the application that prepares the data to be written. The CRC and the P/Q parity ECC/EDC information (depending on the sector type) have to be supplied by the application that prepares the data to be written.

EXAMPLES

For all examples below, it will be assumed that the machine includes two drives. The reader is assumed to be target 1 on the primary SCSI bus. The CD/DVD/BluRay-recorder is assumed to be target 2 on the primary SCSI bus of the machine.

If there is only one drive in the machine, the **dev=** option may be omitted in the examples below, but in this case the examples for replication without intermediate files do not apply.

Replicating an Audio CD

To copy an audio CD in the most accurate way, first run

```
cdda2wav dev=1,0 paraopts=proof −vall cddb=0 −B −Owav
```

and then run

```
cdrecord dev=2,0 −v −dao −useinfo −text *.wav
```

This will try to copy track indices and to read CD-Text information from disk. If there is no CD-Text information, **cdda2wav** will try to get the information from freedb.org instead.

To copy an audio CD from a pipe (without intermediate files), first run

```
cdda2wav dev=1,0 −vall cddb=0 −info−only
```

and then run

```
cdda2wav dev=1,0 −no−infofile −B −Oraw − | \
cdrecord dev=2,0 −v −dao −audio −useinfo −text *.inf
```

This will get all information (including track size info) from the ***.inf** files and then read the audio data from stdin.

If you like to write from **stdin**, make sure that cdrecord is called with a large enough FIFO size (e.g. **fs=128m**), reduce the write speed to a value below the read speed of the source drive (e.g. **speed=12**),

and switch the burn-free option for the recording drive on by adding **driveropts=burnfree**. For the same reason, it is not recommended to extract the audio data in paranoia mode in this case.

Replicating a simple CD-ROM/DVD-ROM/BD-ROM

To copy a simple disk, first read the master using:

readcd dev=1,0 f=somefile

Then write the disk using:

cdrecord dev=2,0 −raw96r −clone −v somefile

Replicating a CD-ROM in clone mode

To copy a CD in clone mode, first read the master CD using:

readcd dev=1,0 −clone f=somefile

or (in case the CD contains many sectors that are unreadable by intention) by calling:

readcd dev=1,0 −clone −nocorr f=somefile

This will create the files *somefile* and *somefile.toc*. Then write the CD using:

cdrecord dev=2,0 −raw96r −clone −v somefile

Creating an Audio CD

To record a pure CD-DA (audio) at single speed, with each track contained in files named *track01.cdaudio*, *track02.cdaudio*, etc.:

cdrecord −v speed=1 dev=2,0 −dao −audio track∗.cdaudio

To check if it will be OK to use double speed for the example above, use the dummy write option:

cdrecord −v −dummy speed=2 dev=2,0 −dao −audio track∗.cdaudio

Creating a mixed Audio-Data CD

To record a mixed-mode CD with an ISO-9660 filesystem from *cdimage.raw* on the first track, the other tracks being audio tracks from the files *track01.cdaudio*, *track02.cdaudio*, etc.:

cdrecord −v dev=2,0 −dao cdimage.raw −audio track∗.cdaudio

Creating a CD-ROM/DVD-ROM/BD-ROM

To record a pure disk at double speed, using data from the file *cdimage.raw*:

cdrecord −v speed=2 dev=2,0 −dao cdimage.raw

To create an image for an ISO-9660 filesystem with Rock Ridge extensions:

mkisofs −R −o cdimage.raw /home/joerg/master/tree

To check the resulting file before writing to disk on Solaris:

mount −r −F fbk −o type=hsfs /dev/fbk0:cdimage.raw /mnt

The **fbk** driver first appeared in 1988.

Solaris 9 or newer comes with a variant of the original **fbk** idea called **lofi**. The command for the lofi variant is:

mount −r −F hsfs ‘ lofiadm −a /tmp/cdimage.raw ‘ /mnt

Note that lofiadm needs absolute path names.

On Linux:

mount cdimage.raw −r −t iso9660 −o loop /mnt

Go on with:

ls −lR /mnt
umount /mnt

If the overall speed of the system is sufficient and the structure of the filesystem is not too complex, cdrecord will run without creating an image of the ISO-9660 filesystem. Simply run the pipeline:

mkisofs −R /master/tree | cdrecord −v −dao fs=6m speed=2 dev=2,0 −

The recommended minimum FIFO size for running this pipeline is 4 MBytes. As the default FIFO size is 4 MB, the **fs=** option needs to be present only if you want to use a different FIFO size. If your system is loaded, you should run mkisofs in the real-time class too. To raise the priority of **mkisofs** replace the command

mkisofs −R /master/tree

by

priocntl −e −c RT −p 59 mkisofs −R /master/tree

on Solaris and by

nice −−18 mkisofs −R /master/tree

on systems that do not have **UNIX International**-compliant real-time scheduling.

Cdrecord runs at priority 59 on Solaris, you should run mkisofs at no more than priority 58. On other systems, you should run mkisofs at no less than nice −−18.

Creating a CD-ROM without file system image on disk has been tested on a Sparcstation-2 with a Yamaha CDR-400. It did work up to quad speed when the machine was not loaded. A faster machine may be able to handle quad speed also in the loaded case.

To handle drives that need to know the size of a track before starting to write, first run

mkisofs −R −quiet −print−size /master/tree

and then run

mkisofs −R /master/tree | cdrecord −v −dao speed=2 dev=2,0 tsize=XXXs −

where *XXX* is replaced by the output of the previous run of mkisofs.

Setting drive options

To set drive options without writing a disk (e.g. to switch a drive to single-session mode), run

cdrecord dev=2,0 −setdropts driveropts=singlesession

If you like to do this when no disk is in the drive, call

cdrecord dev=2,0 −force −setdropts driveropts=singlesession

ENVIRONMENT

CDR_DEVICE
: This may either hold a device identifier that is suitable to the open call of the SCSI transport library or a label in the file /etc/default/cdrecord.

CDR_SPEED
: Sets the default speed value for writing (see also −**speed** option).

CDR_FIFOSIZE
: Sets the default size of the FIFO (see also **fs=**# option).

CDR_FORCERAWSPEED

If this environment variable is set, **cdrecord** will allow you to write at the full RAW encoding speed a single CPU supports. This will create high potential of buffer underruns. Use with care.

CDR_FORCESPEED

If this environment variable is set, **cdrecord** will allow you to write at the full DMA speed the system supports. There is no DMA reserve for reading the data that is to be written from disk. This will create high potential of buffer underruns. Use with care.

If this environment variable is set to the value **any**, **cdrecord** allows to write at any speed even though it may fail later with a buffer underrun.

RSH If the **RSH** environment is present, the remote connection will not be created via **rcmd**(3) but by calling the program pointed to by **RSH**. Use e.g. **RSH=**/usr/bin/ssh to create a secure shell connection.

Note that this forces **cdrecord** to create a pipe to the **rsh(1)** program and disallows **cdrecord** to directly access the network socket to the remote server. This makes it impossible to set up performance parameters and slows down the connection compared to a **root**-initiated **rcmd(3)** connection.

RSCSI If the **RSCSI** environment is present, the remote SCSI server will not be the program **/opt/schily/sbin/rscsi** but the program pointed to by **RSCSI**. Note that the remote SCSI server program name will be ignored if you log in using an account that has been created with a remote SCSI server program as login shell.

FILES

/etc/default/cdrecord

Default values can be set for the following options in /etc/default/cdrecord. For example: CDR_FIFOSIZE=8m or CDR_SPEED=2

CDR_DEVICE

This may either hold a device identifier that is suitable to the open call of the SCSI transport library or a label in the file /etc/default/cdrecord that allows to identify a specific drive on the system.

CDR_SPEED

Sets the default speed value for writing (see also −**speed** option).

CDR_FIFOSIZE

Sets the default size of the FIFO (see also **fs=**# option).

CDR_MAXFIFOSIZE

Sets the maximum size of the FIFO (see also **fs=**# option).

Any other label

is an identifier for a specific drive on the system. Such an identifier may not contain the characters ’,’, ’/’, ’@’ or ’:’.

Each line that follows a label contains a TAB separated list of items. Currently, four items are recognized: the SCSI ID of the drive, the default speed that should be used for this drive, the default FIFO size that should be used for this drive and drive specific options. The values for *speed* and *fifosize* may be set to −1 to tell cdrecord to use the global defaults. The value for driveropts may be set to "" if no driveropts are used. A typical line may look this way:

teac1= 0,5,0 4 8m ""

yamaha= 1,6,0 −1 −1 burnfree

This tells **cdrecord** that a drive named *teac1* is at scsibus 0, target 5, lun 0 and should be used with speed 4 and a FIFO size of 8 MB. A second drive may be found at scsibus 1, target 6, lun 0 and uses the default speed and the default FIFO size.

***.inf** The ***.inf** files are created by **cdda2wav** where * is replaced by the actual audio file prefix. They are read and used by **cdrecord** in case cdrecord was called with the −**useinfo** option.

There are three general types of parameters:

numerical parameters
A numerical parameter is a number and directly follows the tag label without any quoting.

unquoted string type parameters
An unquoted parameter is make from one or more words that directly follow the tag label. How many words from the parameter list are used by cdrecord depends on the tag label.

quoted string type parameters
A string type parameter is enclosed in single quotes. The string starts after the first single quote character that follows the tag label and ends before the last single quote on the same line. It needs no escape sequences in case that a single quote appears inside the string. Any text to the right of the rightmost single quote character is ignored.

The order of the tag labels in the file is not important.

The following tag labels may appear in a ***.inf** file:

CDINDEX_DISCID=
The cdindex disk ID is used by the **musicbrainz** CD-database.

This tag label uses a quoted string type parameter.

This tag label is ignored by **cdrecord**.

CDDB_DISCID=
The cddb disk ID is used by the **cddb** and the **freedb** CD-database.

This tag label uses a numerical parameter.

This tag label is ignored by **cdrecord**.

MCN= The Media Catalog Number (MCN) is a 13 digit number that follows UPC/EAN-13 rules.

The data is used by cdrecord to create sub-channel data.

ISRC= The International Standard Recording Code (ISRC) is a 12 byte string that is created from two uppercase characters for the country code, followed by three uppercase characters for the owner, followed by two digits for the year of recording followed by five digits for the recording serial number.

To increase the readability of the ISRC tag, there may be a minus sign between every two fields of the ISRC string.

The data is used by cdrecord to create sub-channel data.

Albumtitle=

The **Album Title** is the name of the disk in the CD-Text information.

This tag label uses a quoted string type parameter.

Tracktitle=

The **Track Title** is the name of the current track in the CD-Text information.

This tag label uses a quoted string type parameter.

Albumperformer=

The **Album Performer** is the global name of the of the performer of the disk in the CD-Text information.

This tag label uses a quoted string type parameter.

Performer=

The **Performer** is the name of the of the performer of the current track in the CD-Text information.

This tag label uses a quoted string type parameter.

Albumsongwriter=

The **Album Songwriter** is the global name of the of the songwriter of the disk in the CD-Text information.

This tag label uses a quoted string type parameter.

Songwriter=

The **Songwriter** is the name of the of the songwriter of the current track in the CD-Text information.

This tag label uses a quoted string type parameter.

Albumcomposer=

The **Album Composer** is the global name of the of the composer of the disk in the CD-Text information.

This tag label uses a quoted string type parameter.

Composer=

The **Composer** is the name of the of the composer of the current track in the CD-Text information.

This tag label uses a quoted string type parameter.

Albumarranger=

The **Album Arranger** is the global name of the of the arranger of the disk in the CD-Text information.

This tag label uses a quoted string type parameter.

Arranger=

The **Arranger** is the name of the of the arranger of the current track in the CD-Text information.

This tag label uses a quoted string type parameter.

Albummessage=

The **Album Message** is the global message text of the disk in the CD-Text information.

This tag label uses a quoted string type parameter.

Message=
The **Message** is the message text of the current track in the CD-Text information.

This tag label uses a quoted string type parameter.

Albumclosed_info=
The **Album Closed_info** is the global closed info text of the disk in the CD-Text information.

This tag label uses a quoted string type parameter.

Closed_info=
The **Closed_info** is the closed info text of the current track in the CD-Text information.

This tag label uses a quoted string type parameter.

Track= The parameter contains the relative number of the current track on the original disk. The first track always has the track number 1, a hidden track uses track number 0.

This tag label uses a numerical parameter.

This tag label is ignored by **cdrecord** except when checking the the **Trackstart** for track #1.

Tracknumber=
The parameter contains the absolute number of the current track, taken from the TOC on the original disk. The first track on the original disk may have a number greater than 1, a hidden track always uses track number 0.

This tag label uses a numerical parameter.

This tag label is currently ignored by **cdrecord** as cdrecord assigns track numbers when compiling the disk information.

Trackstart=
The parameter contains the track start offset in sectors on the original disk. If the current track becomes the first track on the new disk and if the track was the first track on the original disk. **cdrecord** uses this number to set up the offset for index 1 on the new disk.

This tag label uses a numerical parameter.

Tracklength=
The parameter is used by **cdrecord** to set up the size of the track on the new disk.

This tag label uses an unquoted string type parameter in the form "sectors, samples".

This label is mandatory for **cdrecord**.

Pre-emphasis=
The pre-emphasis parameter controls whether the related pre-emphasis bit in the subchannel data is set by cdrecord. Permitted values for this parameter are **yes** and **no**.

This tag label uses an unquoted string type parameter. Valid values are **yes** and **no**.

Channels=

The parameter of this tag is the number of channels on the disk. All CD-audio disks use stereo recording and thus a 2 is the correct parameter.

This tag label uses a numerical parameter.

This label is currently ignored by **cdrecord**.

Copy_permitted=

The parameter for this tag label contains information about the copyright state of a track on the original disk.

This tag label uses an unquoted string type parameter. Valid values are:

yes The **digital copy permitted** bit is set in the TOC and in the sub-channel data. If this bit is set, the related track is not copyright protected and may be copied infinitely.

no The **digital copy permitted** bit is not set in the TOC. The **digital copy permitted** bit in the sub-channel data alters with 9.375 Hz. This is called **Serial Copy Management System (SCMS)**. The sense of this track state is to flag that the creator of the CD does not have the copyright permission to create copies of the related track. The related track is copyright protected and the creator of the CD thus is just given the permission to create one single copy from fair use rights and no further copies are permitted from this source.

once The **digital copy permitted** bit is not set in the TOC and in the sub-channel data. The sense of this track state is to flag that the related track is copyright protected and thus may not be coped infinitely. One single copy from fair use rights is permitted.

Note that many CDs sold by the music industry have **SCMS** flagged for one or more tracks, signalling that the related content company does not own the copyright to make copies from this track.

Endianess=

The parameter for this tag is the byte order used in the audio data file that was created for this track.

This tag label uses an unquoted string type parameter. Valid values are **little** and **big**.

This label is ignored by **cdrecord** as the endianess is retrieved from the audio file format.

Index= The parameter list for this tag is a list of numbers that are sector numbers counting relatively to the logical beginning of the track (which always is at index #1). As any track needs to have an entry for index #1, the first entry in the list is always 0. If more entries are present for this tag, there are more offset values that correspond to index values greater than 1.

This tag label uses an unquoted string type parameter that contains a list of space separated index offset numbers.

Index0=

The parameter for this tag is a number that represents the number of sectors relatively to the beginning (index #1) of this track. This number identifies where index #0 of the next track begins. It the parameter is set to -1, the next track has no index #0, resulting

in pregap size 0 for the next track.

Note that **cdrecord** strictly follows the CD-standard that defines that the logical beginning of a track is at the location where index #1 starts in this track. If index #0 for track **n** contains audio data, the related audio data is a logical part of track **n-1**.

This tag label uses a numerical parameter.

MD5-offset=

The parameter for this tag is the byte offset where the raw audio data begins in the related audio file.

This tag label uses a numerical parameter.

This label is ignored by **cdrecord**.

MD5-size=

The parameter for this tag is the number of bytes of raw audio data in the related audio file.

This tag label uses a numerical parameter.

This label is ignored by **cdrecord**.

MD5-sum=

The parameter for this tag is the md5 sum for the raw audio data in the related audio file.

This tag label uses a numerical parameter.

This label is ignored by **cdrecord**.

.cue** The ***.cue** files are CD-structure description files introduced by **CDRWIN**. They are read and used by **cdrecord** in case cdrecord was called with the **cuefile=name.cue* option.

The following commands are supported in CUE files:

ARRANGER *arranger-string*

This command is used to specify the name of a arranger for a disk that includes CD-Text enhancements.

The parameter is the name of a arranger. If the string contains any spaces, it must be enclosed in quotation marks.

If the **ARRANGER** command appears before any **TRACK** command, the string parameter will be encoded as the arranger of the entire disk. If the **ARRANGER** command appears after a **TRACK** command, the string parameter will be encoded the the arranger of the current track.

This command is only accepted if the **cdrecord** specific CUE extensions are permitted.

CATALOG *media-catalog-number*

This command is used to specify the disc's **Media Catalog Number**. The *media-catalog-number* is a 13 digit number that follows UPC/EAN-13 rules.

This command can appear only once in the CUE SHEET file. It must appear before

any **TRACK** command.

CDTEXTFILE *filename*

This command is used to specify the name of a file that contains binary encoded CD-Text information. **CDRWIN** only accepts headerless binary encoded CD-Text information, but **cdrecord** also accepts binary encoded CD-Text information with an MMC-compliant header. The CD-Text information is ignored by **cdrecord** unless the **−text** option is used.

If the filename contains spaces, it must be enclosed in quotation marks.

COMPOSER *composer-string*

This command is used to specify the name of a composer for a disk that includes CD-Text enhancements.

The parameter is the name of a composer. If the string contains any spaces, it must be enclosed in quotation marks.

If the **COMPOSER** command appears before any **TRACK** command, the string parameter will be encoded as the composer of the entire disk. If the **COMPOSER** command appears after a **TRACK** command, the string parameter will be encoded the the composer of the current track.

This command is only accepted if the **cdrecord** specific CUE extensions are permitted.

FILE *filename filetype*

This command is used to specify a data or audio file that contains data to be written to the medium.

If the filename contains spaces, it must be enclosed in quotation marks.

The following values are allowed for the file type parameter:

BINARY Intel binary file (LSB first)

MOTOTOLA
Motorola binary file (MSB first)

AIFF Audio AIFF file

WAVE Audio WAVE file

MP3 Audio MP3 file

AU Audio AU file (only permitted if **cdrecord** CUE extensions are enabled)

OGG Audio OGG file (only permitted if **cdrecord** CUE extensions are enabled)

All audio files (WAVE, AIFF, MP3, AU and OGG) must be in 44100 Hz 16 bit stereo format. MP3 and OGG is currently unsupported.

If an audio file is not an exact multiple of a CDROM sector (2352 bytes), then is is padded with zeroes to fill up to the needed size.

All **FILE** commands need to be before a related **TRACK** command and after the last **INDEX** command or **POSTGAP** command for the previous track.

If the **cdrecord** specific CUE extensions are enabled, then a **FILE** command may also appear between an **INDEX 00** and an **INDEX 01** command. This allows to let the create one file per track where the file starts at **INDEX 01** of the track and enda after **INDEX 00** of the following track. In this case, no **FILE** command is allowed before

the related **TRACK** command.

FLAGS *flags*
This command is used to set special subcode flags within a track.

The following flags are supported:

DCP Digital copy permitted

4CH Four channel audio

PRE Pre-emphasis enabled (audio tracks only)

SCMS Serial copy management system (not supported by all recorders)

More than one flag type argument may appear after the FLAGS command (e.g FLAGS DCP PRE).

The **FLAGS** command must appear after a **TRACK** command but before any **INDEX** command. Only one **FLAGS** command is allower per **TRACK command.**

The fourth subcode flag that marks data tracks is set automatically for data tracks.

INDEX *number mm:ss:ff*
This command is used to specify indexes within a track.

The first parameter is the index number in the range 0-99.

The second parameter is a relative time in minutes, seconds and frames (there are 75 frames/second).

All index numbers must be between 0 and 99 inclusive. The first index for a track must be either 0 or 1 with all indexes being sequential to the first one. The first index for a file must start at 00:00.00.

INDEX 00 specifies the starting time of the **pregap** of the track.

INDEX 01 specifies the starting time of the track. This is the index that is stored in the table of content for the disk as the track start.

INDEX > 1
specifies a subindex within a track.

ISRC *recording code*
This command is used to specify the **International Standard Recording Code (ISRC)** of a track. This is a code that should exist for all commercial audio tracks.

The ISRC code must be 12 characters in length. The first two characters are characters that are from the two character country code. The next three characters are alphanumeric and describe the studio code. The next two characters are the last two digits from the recording year. The last 5 characters are digits that form a serial number that is unique for the same studio and year.

If **cdrecord** specific CUE extensions are permitted, the four fields of the ISRC may be separated by a minus sign.

If the **ISRC** command is used, it must appear after a **TRACK** command but before any **INDEX** command.

MESSAGE *message-string*
This command is used to specify the test of a message for a disk that includes CD-Text enhancements.

The parameter is the test of a message. If the string contains any spaces, it must be enclosed in quotation marks.

If the **MESSAGE** command appears before any **TRACK** command, the string parameter will be encoded as the message of the entire disk. If the **MESSAGE** command appears after a **TRACK** command, the string parameter will be encoded the the message of the current track.

This command is only accepted if the **cdrecord** specific CUE extensions are permitted.

PERFORMER *performer-string*

This command is used to specify the name of a performer for a disk that includes CD-Text enhancements.

The parameter is the name of the performer. If the string contains any spaces, it must be enclosed in quotation marks.

If the **PERFORMER** command appears before any **TRACK** command, the string parameter will be encoded as the performer of the entire disk. If the **PERFORMER** command appears after a **TRACK** command, the string parameter will be encoded the the performer of the current track.

POSTGAP *mm:ss:ff*

This command is used to specify the length of a postgap at the end of a track. The postgap data is generated internally by **cdrecord**. No data is consumed from the current data file.

The parameter specifies the postgap length in minutes, seconds and frames.

The **POSTGAP** command must appear after all **INDEX** commands for the current track. Only one **POSTGAP** command is allowed per track.

PREGAP *mm:ss:ff*

This command is used to specify the length of a pregap at the beginning of a track. The pregap data is generated internally by **cdrecord**. No data is consumed from the current data file.

The parameter specifies the postgap length in minutes, seconds and frames.

The **PREGAP** command must appear after a **TRACK** command but before any **INDEX** command. Only one **PREGAP** command is allowed per track.

REM *comment*

This command is used to put comments into a CUE file.

The text that appears in the line after a **REM** command is usually ignored. There is an exception: The special comment **REM CDRTOOLS** is used to enable **cdrecord** specific CUE extensions in the parser.

SONGWRITER *songwriter-string*

This command is used to specify the name of a songwriter for a disk that includes CD-Text enhancements.

The parameter is the name of a songwriter. If the string contains any spaces, it must be enclosed in quotation marks.

If the **SONGWRITER** command appears before any **TRACK** command, the string parameter will be encoded as the songwriter of the entire disk. If the **SONGWRITER** command appears after a **TRACK** command, the string parameter will be encoded the the songwriter of the current track.

TITLE *title-string*
This command is used to specify a title for a disk that includes CD-Text enhancements.

The parameter is the title for a track or for the disk. If the string contains any spaces, it must be enclosed in quotation marks.

If the **TITLE** command appears before any **TRACK** command, the string parameter will be encoded as the title of the entire disk. If the **TITLE** command appears after a **TRACK** command, the string parameter will be encoded the the title of the current track.

TRACK *number datatype*
This command is used to start a new **TRACK**.

The first parameter is a track number in the range 1-99.

The second parameter specifies the track data type.

The following datatypes are permitted:

AUDIO Audio/Music (2352)

CDG Karaoke CD+G (2448)

MODE1/2048
CDROM Mode1 Data (cooked)

MODE1/2352
CDROM Mode1 Data (raw)

MODE2/2336
CDROM-XA Mode2 Data

MODE2/2352
CDROM-XA Mode2 Data

CDI/2336 CDI Mode2 Data

CDI/2352 CDI Mode2 Data

All track numbers must be between 1 and 99 inclusive. The first track number can be greater than one, but all track numbers after the first must be sequential. There must be at least one track per file.

SEE ALSO

cdda2wav(1), **readcd**(1), **scg**(7), **fbk**(7), **mkisofs**(8), **rcmd**(3), **ssh**(1).

NOTES

Not all options described in this manual may be supported by the OpenSource variant of cdrecord. Cdrecord issues a warning if an attempt is made to use an option that has been disabled in the OpenSource variant.

On Solaris before Solaris 10 Update 1, you need to stop the volume management if you like to use the USCSI fallback SCSI transport code. Even things like **cdrecord -scanbus** will not work if the volume management is running.

Disks made in **Track At Once** mode are not suitable as a master for direct mass production by CD-manufacturers. You will need the **disk at once** option to record such disks. Nevertheless the disks made in **Track At Once** will normally be read in all CD-players. Some old audio CD-players however may produce a two second click between two audio tracks.

The minimal size of a track is 4 seconds or 300 sectors. If you write smaller tracks, the CD-recorder will add dummy blocks. This is not an error, even though the SCSI-error message looks this way.

Cdrecord has been tested on an upgraded Philips CDD-521 recorder at single and double speed on a SparcStation 20/502 with no problems, slower computer systems should work also. The newer Philips/HP/Plasmon/Grundig drives as well as Yamaha CDR-100 and CDR-102 work also. The Plasmon RF-4100 works, but has not been tested in multi-session. A Philips CDD-521 that has not been upgraded will not work. The Sony CDU-924 has been tested, but does not support XA-mode2 in hardware. The Sony therefore cannot create conforming multi-session disks. The Ricoh RO-1420C works, but some people seem to have problems to use them with speed=2, try speed=0 in this case.

The Yamaha CDR-400 and all new SCSI-3/mmc conforming drives are supported in single and multi-session.

You should run several tests in all supported speeds of your drive with the **−dummy** option turned on if you are using **cdrecord** on an unknown system. Writing a CD is a real-time process. **NFS** will not always deliver constantly the needed data rates. If you want to use **cdrecord** with CD-images that are located on a **NFS** mounted filesystem, be sure that the FIFO size is big enough. The author used **cdrecord** with medium load on a SS20/502 and even at quad speed on a Sparcstation-2 which was heavily loaded, but it is recommended to leave the system as lightly loaded as possible while writing a CD. If you want to make sure that buffer underruns are not caused by your source disk, you may use the command

cdrecord −dummy dev=2,0 padsize=600m /dev/null

to create a disk that is entirely made of dummy data. **Cdrecord** needs to run as root to get access to the **/dev/scg?** device nodes and to be able to lock itself into memory.

If you don’t want to allow users to become root on your system, **cdrecord** may safely be installed suid root. This allows all users or a group of users with no root privileges to use **cdrecord. Cdrecord** in this case checks if the real user would have been able to read the specified files. To give all users access to use **cdrecord,** enter:

```
chown root /opt/schily/bin/cdrecord
chmod 4711 /opt/schily/bin/cdrecord
```

To give a restricted group of users access to cdrecord enter:

```
chown root /opt/schily/bin/cdrecord
chgrp cdburners /opt/schily/bin/cdrecord
chmod 4710 /opt/schily/bin/cdrecord
```

and add a group *cdburners* on your system.

Never give write permissions for non root users to the */dev/scg?* devices unless you would allow anybody to read/write/format all your disks.

You should not connect old drives that do not support disconnect/reconnect to either the SCSI bus that is connected to the CD-recorder or the source disk.

A Compact Disc can have no more than 99 tracks.

When creating a disc with both audio and data tracks, the data should be on track 1 otherwise you should create a CDplus disk which is a multi-session disk with the first session containing the audio tracks and the following session containing the data track.

Many operating systems are not able to read more than a single data track, or need special software to do so.

More information on the SCSI command set of a HP CD-recorder can be found at:

http://www.hp.com/isgsupport/cdr/index.html

If you have more information or SCSI command manuals for currently unsupported CD/DVD/BluRay-recorders please contact the author.

The Philips CDD 521 CD-recorder (even in the upgraded version) has several firmware bugs. Some of them will force you to power cycle the device or to reboot the machine.

When using **cdrecord** with the **Linux SCSI generic driver**, you should note that **cdrecord** uses a layer, that tries to emulate the functionality of the scg driver on top of the drives of the local operating system. Unfortunately, the sg driver on **Linux** has several flaws:

- It cannot see if a SCSI command could not be sent at all.
- It cannot get the SCSI status byte. **Cdrecord** for that reason cannot report failing SCSI commands in some situations.
- It cannot get real DMA count of transfer. **Cdrecord** cannot tell you if there is a DMA residual count.
- It cannot get number of bytes valid in auto sense data. **Cdrecord** cannot tell you if device transfers no sense data at all.
- It fetches too few data in auto request sense (CCS/SCSI-2/SCSI-3 needs >= 18).

The FIFO percent output is computed just after a block of data has been written to the CD/DVD/BluRay-recorder. For this reason, there will never be 100% FIFO fill ratio while the FIFO is in streaming mode.

DIAGNOSTICS

You have 9 seconds to type ^C to abort **cdrecord** after you see the message:

Starting to write CD at speed %d in %s mode for %s session.

A typical error message for a SCSI command looks like:

```
cdrecord: I/O error. test unit ready: scsi sendcmd: no error
CDB:  00 20 00 00 00 00
status: 0x2 (CHECK CONDITION)
Sense Bytes: 70 00 05 00 00 00 00 0A 00 00 00 00 25 00 00 00 00 00
Sense Key: 0x5 Illegal Request, Segment 0
Sense Code: 0x25 Qual 0x00 (logical unit not supported) Fru 0x0
Sense flags: Blk 0 (not valid)
cmd finished after 0.002s timeout 40s
```

The first line gives information about the transport of the command. The text after the first colon gives the error text for the system call from the view of the kernel. It usually is: **I/O error** unless other problems happen. The next words contain a short description for the SCSI command that fails. The rest of the line tells you if there were any problems for the transport of the command over the SCSI bus. **fatal error** means that it was not possible to transport the command (i.e. no device present at the requested SCSI address).

The second line prints the SCSI command descriptor block for the failed command.

The third line gives information on the SCSI status code returned by the command, if the transport of the command succeeds. This is error information from the SCSI device.

The fourth line is a hex dump of the auto request sense information for the command.

The fifth line is the error text for the sense key if available, followed by the segment number which is only valid if the command was a *copy* command. If the error message is not directly related to the current command, the text *deferred error* is appended.

The sixth line is the error text for the sense code and the sense qualifier if available. If the type of the device is known, the sense data is decoded from tables in *scsierrs.c* . The text is followed by the error value for a field replaceable unit.

The seventh line prints the block number that is related to the failed command and text for several error flags. The block number may not be valid.

The eighth line reports the timeout set up for this command and the time that the command really needed to complete.

The following message is not an error:

```
Track 01: Total bytes read/written: 2048/2048 (1 sectors).
cdrecord: I/O error. flush cache: scsi sendcmd: no error
CDB:  35 00 00 00 00 00 00 00 00 00
status: 0x2 (CHECK CONDITION)
Sense Bytes: F0 00 05 80 00 00 27 0A 00 00 00 00 B5 00 00 00 00 00
Sense Key: 0x5 Illegal Request, Segment 0
Sense Code: 0xB5 Qual 0x00 (dummy data blocks added) Fru 0x0
Sense flags: Blk −2147483609 (valid)
cmd finished after 0.002s timeout 40s
```

It simply notifies that a track that is smaller than the minimum size has been expanded to 300 sectors.

BUGS

Cdrecord has even more options than **ls**.

There should be a recover option to make disks usable, that have been written during a power failure.

CREDITS

Bill Swartz (Bill_Swartz@twolf.com)
For helping me with the TEAC driver support

Aaron Newsome (aaron.d.newsome@wdc.com)
For letting me develop Sony support on his drive

Eric Youngdale (eric@andante.jic.com)
For supplying mkisofs

Gadi Oxman (gadio@netvision.net.il)
For tips on the ATAPI standard

Finn Arne Gangstad (finnag@guardian.no)
For the first FIFO implementation.

Dave Platt (dplatt@feghoot.ml.org)
For creating the experimental packet writing support, the first implementation of CD-RW blanking support, the first .wav file decoder and many nice discussions on cdrecord.

Chris P. Ross (cross@eng.us.uu.net)

For the first implementation of a BSDI SCSI transport.

Grant R. Guenther (grant@torque.net)
For creating the first parallel port transport implementation for Linux.

Kenneth D. Merry (ken@kdm.org)
for providing the CAM port for FreeBSD together with Michael Smith (msmith@freebsd.org)

Heiko Eiβfeldt (heiko@hexco.de)
for making libedc_ecc available (needed to write RAW data sectors).

MAILING LISTS

If you want to actively take part on the development of cdrecord, you may join the developer mailing list via this URL:

http://lists.berlios.de/mailman/listinfo/cdrecord-developers

AUTHOR

Jörg Schilling
Seestr. 110
D-13353 Berlin
Germany

Additional information can be found on:
http://cdrecord.berlios.de/private/cdrecord.html

If you have support questions, send them to:

cdrecord-support@berlios.de

If you have definitely found a bug, send a mail to:

cdrecord-developers@berlios.de
or **joerg.schilling@fokus.fraunhofer.de**

To subscribe, use:

http://lists.berlios.de/mailman/listinfo/cdrecord-developers
or **http://lists.berlios.de/mailman/listinfo/cdrecord-support**

INTERFACE STABILITY

The interfaces provided by **cdrecord** are designed for long term stability. As **cdrecord** depends on interfaces provided by the underlying operating system, the stability of the interfaces offered by **cdrecord** depends on the interface stability of the OS interfaces. Modified interfaces in the OS may enforce modified interfaces in **cdrecord**.