NAME

readcd − read or write data Compact Discs

SYNOPSIS

readcd dev=*device* [*options*]

DESCRIPTION

Readcd is used to read or write Compact Discs.

The *device* refers to *scsibus*/*target*/*lun* of the drive. 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 drive 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 corect 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 referres 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*,

To make **readcd** portable to all UNIX platforms, the syntax **dev=** *devicename*:*scsibus*,*target*, is preferred as is hides OS specific knowledge about device names from the user. A specific OS must 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**.

OPTIONS

If no options except the *dev=* option have been specified, **readcd** goes into intercative mode. Select a primary function and then follow the instructions.

−version
Print version information and exit.

−v Increment the level of general verbosity by one.

−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 drive. 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.

−w Switch to write mode. If this option is not present, **readcd** reads from the specified device.

sectors=*range*
Specify a sector range that should be read. The range is specified by the starting sector number, a minus sign and the ending sector number.

EXAMPLES

For all examples below, it will be assumed that the drive is connected to the primary SCSI bus of the machine. The SCSI target id is set to 2.

To read the complete media from a CD-ROM writing the data to the file *cdimage.raw*:

readcd dev=2,0 f=cdimage.raw

To read sectors from range 150 ... 10000 from a CD-ROM writing the data to the file *cdimage.raw*:

readcd dev=2,0 sectors=150-10000 f=cdimage.raw

To write the data from the file *cdimage.raw* (e.g. a filesystem image from **mkisofs**) to a DVD-RAM, call:

readcd dev=2,0 -w f=cdimage.raw

FILES

SEE ALSO

cdrecord(1), **mkisofs**(1), **scg**(7), **fbk**(7).

NOTES

If you don't want to allow users to become root on your system, **readcd** may safely be installed suid root. This allows all users or a group of users with no root privileges to use **readcd. Readcd** in this case will only allow access to CD-ROM type drives- To give all user access to use **readcd,** enter:

```
chown root /usr/local/bin/readcd
chmod 4711 /usr/local/bin/readcd
```

To give a restricted group of users access to **readcd** enter:

```
chown root /usr/local/bin/readcd
chgrp cdburners /usr/local/bin/readcd
chmod 4710 /usr/local/bin/readcd
```

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.

When using **readcd** with the broken **Linux SCSI generic driver.** You should note that **readcd** uses a hack, that tries to emulate the functionality of the scg driver. Unfortunately, the sg driver on **Linux** has several severe bugs:

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

DIAGNOSTICS

A typical error message for a SCSI command looks like:

```
readcd: 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)

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 that 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.

BUGS

CREDITS

MAILING LISTS

If you want to actively take part on the development of cdrecord, you may join the cdwriting mailing list by sending mail to:

other-cdwrite-request@lists.debian.org

and include the word *subscribe* in the body. The mail address of the list is:

cdwrite@lists.debian.org

AUTHOR

Jörg Schilling
Seestr. 110
D-13353 Berlin
Germany

Additional information can be found on:
http://www.fokus.gmd.de/usr/schilling/cdrecord.html

Mail bugs and suggestions to:

joerg@schily.isdn.cs.tu-berlin.de or **js@cs.tu-berlin.de** or **schilling@fokus.gmd.de**