NAME

scgskeleton − Skeleton program for SCSI transport using libscg

SYNOPSIS

scgskeleton dev=*device* [*options*]

DESCRIPTION

Scgskeleton is s skeleton program to demonstrate the interfaces and capabilities of the SCSI transport library **libscg**.

Device naming

For a list of possible device name parameters call **scgskeleton −scanbus** or **scgskeleton dev=help** and then use the right **dev=** parameter based on the device listing.

OPTIONS

−help Prints a short summary of the **p** options and exists.

−version

Print version information and exit.

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

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 **scgskeleton** 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 **scgskeleton** described below.

Using logical names for devices

If no *dev* option is present, **scgskeleton** 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 specifyer but no address notation, **scgskeleton** tries to scan the SCSI address space for CD-ROM drives. If exactly one is found, this is used by default.

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.

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.

−**silent**, −**s**

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

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

−**V** Increment the verbose level with respect of SCSI command transport by one. This helps to debug problems during the process, that occur in the CD-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. −**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 devices on a system. The numbers printed out as labels are computed by: **bus * 100 + target**

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, **scgskeleton** defaults to a transfer size of 256 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.

EXAMPLES

ENVIRONMENT

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.

SEE ALSO

cdrecord(1), **scg**(7), **rcmd**(3), **ssh**(1).

NOTES

DIAGNOSTICS

A typical error message for a SCSI command looks like:

```
scgskeleton: 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 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.

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

BUGS

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**