NAME

scgcheck - check and validate the ABI of libscg

SYNOPSIS

scgcheck [*options*]

DESCRIPTION

Scgcheck is used to check and verify the Application Binary Interface of libscg.

Device naming

Most users do not need to care about device naming at all, as in **-auto** mode, **scgcheck** implements **auto target** support and automagically finds a test drive in case that exactly one CD-ROM type drive is available in the system.

OPTIONS

-version

Print version information and exit.

-auto

Instead of asking to confirm each test before running it, **scgcheck** tries to do a fully automated test.

dev=*target*

Set the SCSI target for the device, 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 device. 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 device 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*,*lun* 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*,*lun*.

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 one 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 **scgcheck** portable to all UNIX platforms, the syntax **dev=** *devicename*:*scsibus*,*target*,*lun* 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 **scgcheck** described below.

Autotarget Mode

If no **dev=** option is present, or if it only contains a transport specifyer but no address notation, **scgcheck** 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.

scgopts=*list*

A comma separated list of SCSI options that are handled by libscg. The implemented options may be updated independently from applications. Currently, one option: **ignore-resid** is supported to work around a Linux kernel bug.

-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.

f=*file*

Specify the log file to be used instead of *check.log*.

SEE ALSO

cdrecord(1), **readcd**(1), **mkisofs**(1),

NOTES

When using **scgcheck** with the broken **Linux SCSI generic driver.** You should note that **scgcheck** 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. **Scgcheck** for that reason cannot report failing SCSI commands in some situations.

- It cannot get real DMA count of transfer. **Scgcheck** cannot tell you if there is an DMA residual count.

- It cannot get number of bytes valid in auto sense data. **Scgcheck** 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:

```
scgcheck: 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. The sense data is decoded from tables in *scsierrs.c* in case the type of device is known.

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 realy needed to complete.

BUGS

None currently known.

Mail bugs and suggestions to **schilytools@mlists.in-berlin.de** or open a ticket at **https://codeberg.org/schilytools/schilytools/issues**.

The mailing list archive may be found at:

https://mlists.in-berlin.de/mailman/listinfo/schilytools-mlists.in-berlin.de.

AUTHOR

scgcheck was initially written by Joerg Schilling and is now maintained by the schilytools project authors.

SOURCE DOWNLOAD

The source code for **scgcheck** is included in the **schilytools** project and may be retrieved from the **schilytools** project at Codeberg at

https://codeberg.org/schilytools/schilytools.

The download directory is

https://codeberg.org/schilytools/schilytools/releases.