Rocky Enterprise Linux 9.2 Manual Pages on command 'sg_sanitize.8'

$ man sg_sanitize.8

SG_SANITIZE(8) SG3_UTILS SG_SANITIZE(8)

NAME

sg_sanitize - remove all user data from disk with SCSI SANITIZE command

SYNOPSIS

sg_sanitize [--ause] [--block] [--count=OC] [--crypto] [--dry-run]

[--desc] [--early] [--fail] [--help] [--invert] [--ipl=LEN] [--over?

write] [--pattern=PF] [--quick] [--test=TE] [--timeout=SECS] [--ver?

bose] [--version] [--wait] [--zero] [--znr] DEVICE

DESCRIPTION

This utility invokes the SCSI SANITIZE command. This command was first

introduced in the SBC-3 revision 27 draft. The purpose of the sanitize

operation is to alter the information in the cache and on the medium of

a logical unit (e.g. a disk) so that the recovery of user data is not

possible. If that user data cannot be erased, or is in the process of

being erased, then the sanitize operation prevents access to that user

data.

Once a SCSI SANITIZE command has successfully started, then user data

from that disk is no longer available. Even if the disk is power cy?

cled, the sanitize operation will continue after power is re-instated until it is complete.

This utility requires either the --block, --crypto, --fail or --over? write option. With the --block, --crypto or --overwrite option the user is given 15 seconds to reconsider whether they wish to erase all the data on a disk, unless the --quick option is given in which case the sanitize operation starts immediately. The disk's INQUIRY response strings are printed out just in case the wrong DEVICE has been given.

If the --early option is given then this utility will exit soon after starting the SANITIZE command with the IMMED bit set. The user can mon? itor the progress of the sanitize operation with the "sg_requests --num=9999 --progress" which sends a REQUEST SENSE command every 30 seconds. Otherwise if the --wait option is given then this utility will wait until the SANITIZE command completes (or fails) and that can be many hours.

If the --wait option is not given then the SANITIZE command is started with the IMMED bit set. If neither the --early nor the --wait options are given then this utility sends a REQUEST SENSE command after every 60 seconds until there are no more progress indications in which case this utility exits silently. If additionally the --verbose option is given the exit will be marked by a short message that the sanitize seems to have succeeded.

OPTIONS

Arguments to long options are mandatory for short options as well. The options are arranged in alphabetical order based on the long option name.

-A, --ause

sets the AUSE bit in the cdb. AUSE is an acronym for "allow un? restricted sanitize exit". The default action is to leave the AUSE bit cleared.

-B, --block

perform a "block erase" sanitize operation.

-c, --count=OC

where OC is the "overwrite count" associated with the "over? write" sanitize operation. OC can be a value between 1 and 31 and 1 is the default.

-C, --crypto

perform a "cryptographic erase" sanitize operation. Note that this erase is often very quick as it simply overwrites an inter? nal cryptographic key with a new value. Those keys are not ac? cessible to users and encrypt all data written then decrypt all data read from the media. The primary reason for doing that is to make this operation fast. This operation can not be reversed.

-d, --desc

sets the DESC field in the REQUEST SENSE command used for polling. By default this field is set to zero. A REQUEST SENSE polling loop is used after the SANITIZE command is issued (as? suming that neither the --early nor the --wait option have been given) to check on the progress of this command as it can take some time.

-D, --dry-run

this option will parse the command line, do all the preparation but bypass the actual SANITIZE command.

-e, --early

the default action of this utility is to poll the disk every 60 seconds to fetch the progress indication until the sanitize is finished. When this option is given this utility will exit "early" as soon as the SANITIZE command with the IMMED bit set to 1 has been acknowledged. This option and --wait cannot both be given.

-F, --fail

perform an "exit failure mode" sanitize operation. Typically re? quires the preceding SANITIZE command to have set the AUSE bit.

-h, --help

print out the usage information then exit.

-i, --ipl=LEN

set the initialization pattern length to LEN bytes. By default it is set to the length of the pattern file (PF) or 4 if the --zero option is given. Only active when the --overwrite option is also given. It is the number of bytes from the PF file that will be used as the initialization pattern (if the --zero option is not given). The minimum size is 1 byte and the maximum is the logical block size of the DEVICE (and not to exceed 65535). If LEN exceeds the PF file size then the initialization pattern is padded with zeros.

-I, --invert

set the INVERT bit in the overwrite service action parameter list. This only affects the "overwrite" sanitize operation. The default is a clear INVERT bit. When the INVERT bit is set then the initialization pattern is inverted between consecutive over? write passes.

-O, --overwrite

perform an "overwrite" sanitize operation. When this option is given then the --pattern=PF or the --zero option is required.

-p, --pattern=PF

where PF is the filename of a file containing the initialization pattern required by an "overwrite" sanitize operation. The length of this file will be used as the length of the initial? ization pattern unless the --ipl=LEN option is given. The length of the initialization pattern must be from 1 to the logical block size of the DEVICE.

-Q, --quick

the default action (i.e. when the option is not given) is to give the user 15 seconds to reconsider doing a sanitize opera? tion on the DEVICE. When this option is given that step (i.e. the 15 second warning period) is skipped.

-T, --test=TE

set the TEST field in the overwrite service action parameter list. This only affects the "overwrite" sanitize operation. The

default is to place 0 in that field.

-t, --timeout=SECS

where SECS is the number of seconds used for the timeout on the SANITIZE command.

-v, --verbose

increase the level of verbosity, (i.e. debug output).

-V, --version

print the version string and then exit.

-w, --wait

the default action (i.e. without this option and the --early option) is to start the SANITIZE command with the IMMED bit set then poll for the progress indication with the REQUEST SENSE command until the sanitize operation is complete (or fails). When this option is given (and the --early option is not given) then the SANITIZE command is started with the IMMED bit clear. For a large disk this might take hours. [A cryptographic erase operation could potentially be very quick.]

-z, --zero

with an "overwrite" sanitize operation this option causes the initialization pattern to be zero (4 zeros are used as the initialization pattern). Cannot be used with the --pattern=PF option. If this option is given twice (e.g. '-zz') then 0xff is used as the initialization byte.

-Z, --znr

sets ZNR bit (zoned no reset) in cdb. Introduced in the SBC-4 revision 7 draft.

NOTES

The SCSI SANITIZE command is closely related to the ATA SANITIZE command, both are relatively new with the ATA command being the first one defined. The SCSI to ATA Translation (SAT) definition for the SCSI SANITIZE command appeared in the SAT-3 revision 4 draft.

When a SAT layer is used to a (S)ATA disk then for OVERWRITE the initialization pattern must be 4 bytes long. So this means either the

--zero option may be given, or a pattern file (with the --pattern=PF option) that is 4 bytes long or set to that length with the --ipl=LEN option.

The SCSI SANITIZE command is related to the SCSI FORMAT UNIT command. It is likely that a block erase sanitize operation would take a similar amount of time as a format on the same disk (e.g. 9 hours for a 2 Ter? abyte disk). The primary goal of a format is the configuration of the disk at the end of a format (e.g. different logical block size or pro? tection information added). Removal of user data is only a side effect of a format. With the SCSI SANITIZE command, removal of user data is the primary goal. If a sanitize operation is interrupted (e.g. the disk is power cycled) then after power up any remaining user data will not be available and the sanitize operation will continue. When a for? mat is interrupted (e.g. the disk is power cycled) the drafts say very little about the state of the disk. In practice some of the original user data may remain and the format may need to be restarted.

Finding out whether a disk (SCSI or ATA) supports SANITIZE can be a challenge. If the user really needs to find out and no other informa? tion is available then try 'sg_sanitize --fail -vvv <device>' and ob? serve the sense data returned may be the safest approach. Using the --fail variant of this utility should have no effect unless it follows an already failed sanitize operation. If the SCSI REPORT SUPPORTED OP? ERATION CODES command (see sg_opcodes) is supported then using it would be a better approach for finding if sanitize is supported.

If using the dd command to check the before and after data of a partic? ular block (i.e. check the erase actually worked) it is a good idea to use the 'iflag=direct' operand. Otherwise the first read might be cached and returned when the same LBA is read a little later. Obviously this utility should only be used to sanitize data on a disk whose mounted file systems (if any) have been unmounted prior to the erase!

EXAMPLES

These examples use Linux device names. For suitable device names in other supported Operating Systems see the sg3_utils(8) man page.

As a precaution if this utility is called with no options then apart from printing a usage message, nothing happens:

```
sg_sanitize /dev/sdm
```

To do a "block erase" sanitize the --block option is required. The user will be given a 15 second period to reconsider, the SCSI SANITIZE command will be started with the IMMED bit set, then this utility will poll for a progress indication with a REQUEST SENSE command until the sanitize operation is finished:

```
sg_sanitize --block /dev/sdm
```

To start a "block erase" sanitize and return from this utility once it is started (but not yet completed) use the --early option:

```
sg_sanitize --block --early /dev/sdm
```

If the 15 second reconsideration time is not required add the --quick option:

```
sg_sanitize --block --quick --early /dev/sdm
```

To do an "overwrite" sanitize a pattern file may be given:

```
sg_sanitize --overwrite --pattern=rand.img /dev/sdm
```

If the length of that "rand.img" is 512 bytes (a typically logical block size) then to use only the first 17 bytes (repeatedly) in the "overwrite" sanitize operation:

```
sg_sanitize --overwrite --pattern=rand.img --ipl=17 /dev/sdm
```

To overwrite with zeros use:

```
sg_sanitize --overwrite --zero /dev/sdm
```

EXIT STATUS

The exit status of sg_sanitize is 0 when it is successful. Otherwise see the sg3_utils(8) man page. Unless the --wait option is given, the exit status may not reflect the success of otherwise of the format.

The Unix convention is that "no news is good news" but that can be a bit unnerving after an operation like sanitize, especially if it fin? ishes quickly (i.e. before the first progress poll is sent). Giving the --verbose option once should supply enough additional output to settle those nerves.

AUTHORS

Written by Douglas Gilbert.

REPORTING BUGS

Report bugs to <dgilbert at interlog dot com>.

COPYRIGHT

SEE ALSO

sg_requests(8), sg_format(8)