



copy). Default is 128 for block sizes less than 2048 bytes, otherwise the default is 32. So for bs=512 the reads and writes will each convey 64 KiB of data by default (less if near the end of the transfer or memory restrictions). When cd/dvd drives are accessed, the block size is typically 2048 bytes and bpt defaults to 32 which again implies 64 KiB transfers.

bs=BS where BS must be the block size of the physical device. Note that this differs from dd(1) which permits BS to be an integral multiple. Default is 512 which is usually correct for disks but incorrect for cdroms (which normally have 2048 byte blocks). For this utility the maximum size of each individual IO operation is BS \* BPT bytes.

cdbsz=6 | 10 | 12 | 16

size of SCSI READ and/or WRITE commands issued on sg device names. Default is 10 byte SCSI command blocks (unless calculations indicate that a 4 byte block number may be exceeded, in which case it defaults to 16 byte SCSI commands).

count=COUNT

copy COUNT blocks from IFILE to OFILE. Default is the minimum (of IFILE and OFILE) number of blocks that sg devices report from SCSI READ CAPACITY commands or that block devices (or their partitions) report. Normal files are not probed for their size. If skip=SKIP or seek=SEEK are given and the count is derived (i.e. not explicitly given) then the derived count is scaled back so that the copy will not overrun the device. If the file name is a block device partition and COUNT is not given then the size of the partition rather than the size of the whole device is used. If COUNT is not given and cannot be derived then an error message is issued and no copy takes place.

dio=0 | 1

permits direct IO to be selected on the write-side (i.e. on OFILE). Only allowed when the read-side (i.e. IFILE) is a sg device. When 1 there may be a "zero copy" copy (i.e. mmap-ed transfer on the read into the user space and direct IO from there on the write, potentially two DMAs and no data copying from the CPU). Default is 0. The same action as 'dio=1' is also available with 'oflag=dio'.

ibs=BS if given must be the same as BS given to 'bs=' option.

if=IFILE

read from IFILE instead of stdin. If IFILE is '-' then stdin is read. Starts read?

ing at the beginning of IFILE unless SKIP is given.

iflag=FLAGS

where FLAGS is a comma separated list of one or more flags outlined below. These flags are associated with IFILE and are ignored when IFILE is stdin.

obs=BS if given must be the same as BS given to 'bs=' option.

of=OFFILE

write to OFFILE instead of stdout. If OFFILE is '-' then writes to stdout. If OFFILE is /dev/null then no actual writes are performed. If OFFILE is '.' (period) then it is treated the same way as /dev/null (this is a shorthand notation). If OFFILE exists then it is `_not_` truncated; it is overwritten from the start of OFFILE unless 'oflag=append' or SEEK is given.

oflag=FLAGS

where FLAGS is a comma separated list of one or more flags outlined below. These flags are associated with OFFILE and are ignored when OFFILE is /dev/null, '.' (period), or stdout.

seek=SEEK

start writing SEEK bs-sized blocks from the start of OFFILE. Default is block 0 (i.e. start of file).

skip=SKIP

start reading SKIP bs-sized blocks from the start of IFILE. Default is block 0 (i.e. start of file).

sync=0 | 1

when 1, does SYNCHRONIZE CACHE command on OFFILE at the end of the transfer. Only active when OFFILE is a sg device file name.

time=0 | 1

when 1, times transfer and does throughput calculation, outputting the results (to stderr) at completion. When 0 (default) doesn't perform timing.

verbose=VERB

as VERB increases so does the amount of debug output sent to stderr. Default value is zero which yields the minimum amount of debug output. A value of 1 reports extra information that is not repetitive. A value 2 reports cdb's and responses for SCSI commands that are not repetitive (i.e. other than READ and WRITE). Error processing is not considered repetitive. Values of 3 and 4 yield output for all SCSI

commands (and Unix read() and write() calls) so there can be a lot of output.

-d, --dry-run

does all the command line parsing and preparation but bypasses the actual copy or read. That preparation may include opening IFILE or OFILE to determine their lengths. This option may be useful for testing the syntax of complex command line invocations in advance of executing them.

-h, --help

outputs usage message and exits.

-v, --verbose

when used once, this is equivalent to verbose=1. When used twice (e.g. "-vv") this is equivalent to verbose=2, etc.

-V, --version

outputs version number information and exits.

## FLAGS

Here is a list of flags and their meanings:

append causes the O\_APPEND flag to be added to the open of OFILE. For normal files this will lead to data appended to the end of any existing data. Cannot be used together with the seek=SEEK option as they conflict. The default action of this utility is to overwrite any existing data from the beginning of the file or, if SEEK is given, starting at block SEEK. Note that attempting to 'append' to a device file (e.g. a disk) will usually be ignored or may cause an error to be reported.

dio is only active with oflag (i.e. 'oflag=dio'). Its action is described in the 'dio=1' option description above.

direct causes the O\_DIRECT flag to be added to the open of IFILE and/or OFILE. This flag requires some memory alignment on IO. Hence user memory buffers are aligned to the page size. Has no effect on sg, normal or raw files.

dpo set the DPO bit (disable page out) in SCSI READ and WRITE commands. Not supported for 6 byte cdb variants of READ and WRITE. Indicates that data is unlikely to be required to stay in device (e.g. disk) cache. May speed media copy and/or cause a media copy to have less impact on other device users.

dsync causes the O\_SYNC flag to be added to the open of IFILE and/or OFILE. The "d" is prepended to lower confusion with the 'sync=0|1' option which has another action (i.e. a synchronisation to media at the end of the transfer).

`excl` causes the `O_EXCL` flag to be added to the open of `IFILE` and/or `OFILE`.

`fua` causes the `FUA` (force unit access) bit to be set in `SCSI READ` and/or `WRITE` commands. This only has effect with `sg` devices. The 6 byte variants of the `SCSI READ` and `WRITE` commands do not support the `FUA` bit. Only active for `sg` device file names.

`null` has no affect, just a placeholder.

## RETIRED OPTIONS

Here are some retired options that are still present:

`fua=0 | 1 | 2 | 3`

force unit access bit. When 3, `fua` is set on both `IFILE` and `OFILE`; when 2, `fua` is set on `IFILE`; when 1, `fua` is set on `OFILE`; when 0 (default), `fua` is cleared on both. See the '`fua`' flag.

## NOTES

A raw device must be bound to a block device prior to using `sgm_dd`. See `raw(8)` for more information about binding raw devices. To be safe, the `sg` device mapping to `SCSI` block devices should be checked with the `lsscsi` utility before use.

Raw device partition information can often be found with `fdisk(8)` [the "`-ul`" argument is useful in this respect].

Various numeric arguments (e.g. `SKIP`) may include multiplicative suffixes or be given in hexadecimal. See the "NUMERIC ARGUMENTS" section in the `sg3_utils(8)` man page.

The `count`, `skip` and `seek` parameters can take 64 bit values (i.e. very big numbers). Other values are limited to what can fit in a signed 32 bit number.

Data usually gets to the user space in a 2 stage process: first the `SCSI` adapter DMA's into kernel buffers and then the `sg` driver copies this data into user memory (write operations reverse this sequence). With memory mapped transfers a kernel buffer reserved by `sg` is memory mapped (see the `mmap(2)` system call) into the user space. When this is done the second (redundant) copy from kernel buffers to user space is not needed. Hence the transfer is faster and requires less "grunt" from the CPU.

All informative, warning and error output is sent to `stderr` so that `dd`'s output file can be `stdout` and remain unpolluted. If no options are given, then the usage message is output and nothing else happens.

For `sg` devices this utility issues `SCSI READ` and `WRITE (SBC)` commands which are appropriate for disks and reading from `CD/DVD/BD` drives. Those commands are not formatted cor?

rectly for tape devices so `sgm_dd` should not be used on tape devices.

This utility stops the copy if any error is encountered. For more advanced "copy on error" logic see the `sg_dd` utility (and its 'coe' flag).

## EXAMPLES

See the examples given in the man page for `sg_dd(8)`.

## SIGNALS

The signal handling has been borrowed from `dd`: `SIGINT`, `SIGQUIT` and `SIGPIPE` output the number of remaining blocks to be transferred and the records in + out counts; then they have their default action. `SIGUSR1` causes the same information to be output yet the copy continues. All output caused by signals is sent to `stderr`.

## EXIT STATUS

The exit status of `sgm_dd` is 0 when it is successful. Otherwise see the `sg3_utils(8)` man page. Since this utility works at a higher level than individual commands, and there are 'coe' and 'retries' flags, individual SCSI command failures do not necessarily cause the process to exit.

## AUTHORS

Written by Douglas Gilbert and Peter Allworth.

## REPORTING BUGS

Report bugs to <[dgilbert@interlog.com](mailto:dgilbert@interlog.com)>.

## COPYRIGHT

Copyright ? 2000-2019 Douglas Gilbert

This software is distributed under the GPL version 2. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

## SEE ALSO

The simplest variant of this utility is called `sg_dd`. A POSIX threads version of this utility called `sgp_dd` is in the `sg3_utils` package. The `lmbench` package contains `lmddd` which is also interesting. `dd(1)`, `ddpt(ddpt)`, `raw(8)`