MyWebUniversity
python
Rocky Linux
Manual Pages
FPDF Library
PDF generator
FPDF
Full credit is given to the above companies including the OS that this PDF file was generated!

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

$ man sg_write_x.8

SG_WRITE_X(8)   SG3_UTILS   SG_WRITE_X(8)

NAME

sg_write_x - SCSI WRITE normal/ATOMIC/SAME/SCATTERED/STREAM, ORWRITE

commands

SYNOPSIS

sg_write_x [--16] [--32] [--app-tag=AT] [--atomic=AB] [--bmop=OP,PGP]

[--bs=BS] [--combined=DOF] [--dld=DLD] [--dpo] [--dry-run] [--fua]

[--generation=EOG,NOG]   [--grpnum=GN]   [--help]   --in=IF

[--lba=LBA[,LBA...]]   [--normal]   [--num=NUM[,NUM...]]   [--off?

set=OFF[,DLEN]]   [--or]   [--quiet]   [--ref-tag=RT]   [--same=NDOB]

[--scat-file=SF] [--scat-raw] [--scattered=RD] [--stream=ID] [--strict]

[--tag-mask=TM] [--timeout=TO] [--unmap=U_A] [--verbose] [--version]

[--wrprotect=WPR] DEVICE

Synopsis per supported command:

sg_write_x --normal --in=IF [--16] [--32] [--app-tag=AT] [--bs=BS]

[--dld=DLD] [--dpo] [--fua] [--grpnum=GN] [--lba=LBA]   [--num=NUM]

[--offset=OFF[,DLEN]]   [--ref-tag=RT]   [--strict]   [--tag-mask=TM]

[--timeout=TO] [--wrprotect=WPR] DEVICE

sg_write_x --or --in=IF [--16] [--32] [--bmop=OP,PGP] [--bs=BS] [--dpo]

[--fua] [--generation=EOG,NOG] [--grpnum=GN] [--lba=LBA] [--num=NUM]

[--offset=OFF[,DLEN]] [--strict] [--timeout=TO] [--wrprotect=OPR] DE?

VICE

sg_write_x --atomic=AB --in=IF [--16] [--32] [--app-tag=AT] [--bs=BS]

[--dpo] [--fua] [--grpnum=GN] [--lba=LBA] [--num=NUM] [--off?

set=OFF[,DLEN]] [--ref-tag=RT] [--strict] [--timeout=TO] [--wrpro?

tect=WPR] DEVICE

sg_write_x --same=NDOB [--16] [--32] [--app-tag=AT] [--bs=BS] [--dpo]

[--fua] [--grpnum=GN] [--in=IF] [--lba=LBA] [--num=NUM] [--off?

set=OFF[,DLEN]] [--ref-tag=RT] [--strict] [--timeout=TO] [--unmap=U_A]

[--wrprotect=WPR] DEVICE

sg_write_x --scattered=RD --in=IF [--16] [--32] [--app-tag=AT]

[--bs=BS] [--dld=DLD] [--dpo] [--fua] [--grpnum=GN]

[--lba=LBA[,LBA...]] [--num=NUM[,NUM...]] [--offset=OFF[,DLEN]]

[--ref-tag=RT] [--scat-file=SF] [--scat-raw] [--strict] [--tag-mask=TM]

[--timeout=TO] [--wrprotect=WPR] DEVICE

sg_write_x --stream=ID --in=IF [--16] [--32] [--app-tag=AT] [--bs=BS]

[--dpo] [--fua] [--grpnum=GN] [--lba=LBA] [--num=NUM] [--off?

set=OFF[,DLEN]] [--ref-tag=RT] [--strict] [--tag-mask=TM] [--time?

out=TO] [--wrprotect=WPR] DEVICE

DESCRIPTION

This utility will send one of six SCSI commands, all associated with

writing data to the given DEVICE. They are a "normal" WRITE, ORWRITE,

WRITE ATOMIC, WRITE SAME, WRITE SCATTERED or WRITE STREAM. This utility

supports the 16 and 32 byte variants of all six commands. Hence some

closely related commands are not supported (e.g. WRITE(10)). All 32

byte variants, apart from ORWRITE(32), require the DEVICE to be format?

ted with type 1, 2 or 3 Protection Information (PI), making all logical

blocks 8 bytes (or a multiple of 8 bytes) longer on the media.

The command line interface is a little crowded with over thirty op?

tions. Hence the SYNOPSIS, after listing all the (long) options, lists

those applicable to each supported command. For each command synopsis,

the option that selects the SCSI command is shown first followed by any
required options. If no command option is given then a "normal" WRITE
is assumed. Even though the --scat-file=SF option can be given for ev?
ery command, it is only shown for WRITE SCATTERED where it is most use?
ful. If the --scat-file=SF option is given then neither the
--lba=LBA[,LBA...] nor the --num=NUM[,NUM...] options should be given.
Only the first item of the --lba=LBA[,LBA...] and the
--num=NUM[,NUM...] options (or first pair (or quintet) from the
--scat-file=SF option) is used for all but the WRITE SCATTERED command.
All commands can take --dry-run and --verbose in addition to those
shown in the SYNOPSIS.

The logical block size in bytes can be given explicitly with the
--bs=BS option, as long as BS is greater than zero. It is typically a
power of two, 512 or greater. If the --bs=BS option is not given or BS
is zero then the SCSI READ CAPACITY command is used to find the logical
block size. First the READ CAPACITY(16) command is tried and if suc?
cessful the logical block size in the response is typically used as the
actual block size for this utility. The exception is when PROT_EN is
set in the response and the --wrprotect=WPR option is given and
non-zero; in which case 8 (bytes) is added to the logical block size to
yield the actual block size used by this utility. If READ CAPACITY(16)
fails then READ CAPACITY(10) is tried and if that works then the logi?
cal block size in the response is used as the actual block size.

The number of bytes this utility will attempt to read from the file
named by IF is the product of the actual block size and the num?
ber_of_blocks (NUM or the sum of NUM arguments). If less bytes are read
from the file IF and the --strict option is given then this utility ex?
its with an exit status of SG_LIB_FILE_ERROR. If less bytes are read
from the file IF and the --strict option is not given then bytes of
zero are substituted for the "missing" bytes and this utility contin?
ues.

Attempts to write multi megabyte data with a single command are likely
to fail for one of several reasons. First the operating system might

object to allocating a buffer that large. Next the SCSI pass-through usually limits data blocks to a few megabytes or less. Finally the storage device might have a limited amount of RAM to support a write operation such as atomic (as it may need to roll back). The storage de? vice can inform the application client of its limitations via the block limits VPD page (0xb0), with the maximum atomic transfer length field amongst others.

A degenerate LBA (Logical Block Address) range descriptor with no PI has an LBA and NUM of zero. A degenerate LBA range descriptor with PI additionally has its RT, AT and TM fields set to zero (note: that is not the default values for RT, AT and TM). They are degenerate in the sense that they are indistinguishable from a pad of zeros that follow the scatter list in the data-out buffer. SBC-4 makes clear that a de? generate LBA range descriptor is valid. This may become an issue if RD given in the --scattered=RD option has the value 0. In this case the logic may need to scan the user provided data to calculate the number of LBA range descriptors which is required by the WRITE SCATTERED cdb. In the absence of other information the logic will take a degenerate LBA range descriptor as a terminator of the scatter list.

The reference for these commands is SBC-4 (T10/BSR INCITS 506-2021) and draft SBC-5 INCITS 571 revision 1 dated 21 May 2021. All six SCSI com? mands are described in those documents. WRITE ATOMIC was added in SBC-4 revision 3; WRITE STREAM was added in SBC-4 revision 7; WRITE SCATTERED was added in SBC-4 revision 11 while the others are in the previous SBC-3 standard.

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.

-6, --16

send the 16 byte cdb variant of the selected SCSI command. If no command is selected then the (normal) SCSI WRITE(16) command is sent. If neither this option nor the --32 option is given then this option is assumed.

-3, --32

send the 32 byte cdb variant of the selected SCSI command. If no command is selected then the (normal) SCSI WRITE(32) command is sent. If neither this option nor the --16 option is given then then the --16 option is assumed. If both this option and the --16 option are given then this option takes precedence. Note that apart from ORWRITE(32) all other 32 byte cdb variants re? quire a DEVICE formatted with type 1, 2 or 3 protection informa? tion.

-a, --app-tag=AT

where AT is the "expected logical block application tag" field found in most of the 32 byte cdb variants (the exception is OR? WRITE(32)). AT is a 16 bit field which means the maximum value is 0xffff. The default value is 0xffff .

-A, --atomic=AB

selects the WRITE ATOMIC command and AB is placed in the Atomic Boundary field of its cdb. It is a 16 bit field so the maximum value is 0xffff. If unsure what value to set, try 0 which will attempt to write the whole data-out buffer in a single atomic operation.

-B, --bmop=OP,PGP

where OP and PGP are the values to be placed in ORWRITE(32)'s BMOP and 'Previous Generation Processing' fields respectively. BMOP is a 3 bit field (ranges from 0 to 7) and PGP is a 4 bit field (ranges from 0 to 15). Both fields default to 0.

-b, --bs=BS

where BS is the logical block size or the actual block size which will be slightly bigger. The default value is zero. If this option is not given or is given with a BS of zero then the SCSI READ CAPACITY(16) command is sent to DEVICE. If that fails then the READ CAPACITY(10) command is sent. The logical and ac? tual block size will be derived from the response of the READ

CAPACITY command.

This section assumes BS is greater than zero. If BS is less than 512 (bytes) or not a multiple of 8, a warning is issued and the utility continues unless the --strict option is also given. If BS is a power of two (e.g. 512) then the logical and actual block size is set to BS (e.g. 512). If BS is not a power of two (e.g. 520) then the logical block size is set to the closest power of two less than BS (e.g. 512) and the actual block size is set to BS (e.g. 520).

If the logical and actual block size are different then a later check will reduce the actual block size back to the logical block size unless --wrprotect=WPR is greater than zero.

-c, --combined=DOF

This option only applies to WRITE SCATTERED and assumes the whole data-out buffer can be read from IF given by the --in=IF option. The whole data-out buffer is the parameter list header, followed by zero or more LBA range descriptors, optionally fol? lowed by some pad bytes and then the data to be written to the media. If the --lba=LBA[,LBA...], --num=NUM[,NUM...] or --scat-file=SF options are also given then an error is gener? ated. The DOF argument should be the value suitable for the 'Logical Block Data Offset' field in the WRITE SCATTERED cdb. This is the offset in the data-out buffer where the data to write to the media commences. The unit of that field is the ac? tual block size which is the logical block size plus a multiple of 8, if protection information (PI) is being sent. When WPR (from --wrprotect=WPR) is greater than zero then PI is expected. SBC-4 revision 15 does not state it but it would appear that a DOF value of 0 is invalid. It is suggested that this option be used with the --strict option while experimenting as random or incorrect data fed in via the --in=IF option could write a lot of "interesting" data all over the DEVICE. If DOF is given as 0 the utility will scan the data in IF until RD LBA range descrip?

tors are found; or if RD is also 0 until a degenerate LBA range descriptor is found.

-D, --dld=DLD

where DLD is the duration limits descriptor spread across 3 bits in the SCSI WRITE(16) and the WRITE SCATTERED(16) cdbs. DLD is between 0 to 7 inclusive with a default of zero. The DLD0 field in WRITE(16) and WRITE SCATTERED(16) is set if (0x1 & DLD) is non-zero. The DLD1 field in both cdbs is set if (0x2 & DLD) is non-zero. The DLD2 field in both cdbs is set if (0x4 & DLD) is non-zero.

-d, --dpo

if this option is given then the DPO (disable page out) bit field in the cdb is set. The default is to clear this bit field. Applies to all commands supported by thus utility except WRITE SAME.

-x, --dry-run

this option exits (with a status of 0) just before it would oth? erwise send the selected SCSI write command. It may still send a SCSI READ CAPACITY command (16 byte variant and perhaps 10 byte variant as well) so the DEVICE is still required. It reads the data in and processes it if the --in=IF and/or the --scat-file=SF options are given. All command line processing and sanity checks (e.g. if the --strict option is given) will be performed and if there is an error then there will be a non zero exit status value.

If this option is given twice (e.g. -xx) then instead of per? forming the selected write SCSI command, the data-out buffer is written to a file called sg_write_x.bin . If it doesn't exist then that file is created in the current directory and is trun? cated if it previously did exist with longer contents. The data-out buffer is written in binary with some information about it written to stdout. For writes other than scattered the file? name and its length in bytes is output to stdout. For write

scattered additionally its number of LBA range descriptors and
its logical block data offset written to stdout.

-f, --fua

if this option is given then the FUA (force unit access) bit
field in the cdb is set. The default is to clear this bit field.
Applies to all commands supported by thus utility except WRITE
SAME.

-G, --generation=EOG,NOG

the arguments for this option are used by the ORWITE(32) command
only. EOG is placed in the "Expected ORWgeneration" field while
NOG is placed in the "New ORWgeneration" field. Both are 32 bits
long and default to zero.

-g, --grpnum=GN

sets the 'Group number' field to GN. Defaults to a value of
zero. GN should be a value between 0 and 63.

-h, --help

output the usage message then exit. Use multiple times for more
help. Currently '-h' to '-hhhh' provide different output.

-i, --in=IF

read data (in binary) from a file named IF in a single OS system
call (in Unix: read(2)). That data is placed in a continuous
buffer and then used as the data-out buffer for all SCSI write
commands apart from WRITE SCATTERED(16 or 32) which may include
other data in the data-out buffer. For WRITE SCATTERED (16 or
32) the data-out buffer is made up of 3 or 4 components in this
order: a parameter list header (32 zero bytes); zero or more LBA
range descriptors, optionally some pad bytes (zeros) and then
data to write to the media. For WRITE SCATTERED IF only provides
the data to write to the media unless --combined=DOF is given.
When the --combined=DOF option is given IF contains all compo?
nents of the WRITE SCATTERED data-out buffer in binary. The data
read from IF starts from byte offset OFF which defaults to zero
and no more than DLEN bytes are read from that point (i.e. from

the file byte offset OFF). If DLEN is zero or not given the rest
of the file IF is read. This option is mandatory apart from when
--same=1 is given (that sets the NDOB bit which stands for "No
Data Out Buffer"). In Unix based OSes, any number of zeros can
be produced by using the /dev/zero device file.
IF may be "-" which is taken as stdin. In this case the --off?
set=OFF,DLEN can be given with OFF set to 0 and LEN set to a
non-zero value, preferably a multiple of the actual block size.
The utility can also deduce how long the IF should be from NUM
(or the sum of them in the case of a scatter list).

-l, --lba=LBA[,LBA...]

where the argument is a single Logical Block Address (LBA) or a
comma separated list of LBAs each of which is the address of the
first block written by the selected write command. Only the
WRITE SCATTERED command can usefully take more than one LBA.
Whatever number of LBAs is given, there needs to be an equal
number of NUMs given to the --num=NUM[,NUM...] option. The first
given LBA joins with the first given NUM to form the first LBA
range descriptor (which T10 number from zero in SBC-4). The sec?
ond LBA joins with the second LBA to form the second LBA range
descriptor, etc. A more convenient way to define a large number
of LBA range descriptors is with the --scat-file=SF option. De?
faults to logical block 0 (which could be dangerous) while NUM
defaults to 0 which makes the combination harmless. LBA is as?
sumed to be in decimal unless prefixed with '0x' or has a trail?
ing 'h'.

-N, --normal

the choice of a "normal" WRITE (16 or 32) command can be made
explicitly with this option. In the absence of selecting any
other command (e.g. --atomic=AB), the choice of a "normal"
WRITE is the default.

-n, --num=NUM[,NUM...]

where the argument is a single NUMber of blocks (NUM) or a comma separated list of NUMs that pair with the corresponding entries in the --lba=LBA[,LBA...] option. If a NUM is given and is not provided by another method (e.g. by using the --scat-file=SF op? tion) then it defaults to the number of blocks derived from the size of the file named by IF (starting at byte offset OFF to the end or the file or DLEN). Apart from the --combined=DOF option, an LBA must be explicitly given (either with I--lba=LBA or via --scat-file=SF), if not NUM defaults to 0 as a safety measure.

-o, --offset=OFF[,DLEN]

where OFF is the byte offset within the file named IF to start reading from. The default value of OFF is zero which is the be? ginning of file named IF. DLEN is the maximum number of bytes to read, starting at byte offset OFF, from the file named IF. Less bytes will be read if an end of file occurs before DLEN is ex? hausted. If DLEN is zero or not given then reading from byte offset OFF to the end of the file named IF is assumed.

-O, --or

selects the ORWRITE command. ORWRITE(16) has similar fields to WRITE(16) apart from the WRPROTECT field being named ORPROTECT with slightly different semantics and the absence of the 3 DLD bit fields. ORWRITE(32) has four extra fields that are set with the --bmop=OP,PGP and --generation=EOG,NOG options. ORWRITE(32) is the only 32 byte cdb command in this utility that does not require a DEVICE formatted with type 1, 2 or 3 PI (although it will still work if it is formatted with PI).

-Q, --quiet

suppress some informational messages such as the ones associated with detected errors when this utility is about to exit. The exit status value is still returned to the operating system when this utility exits.

-r, --ref-tag=RT

where RT is the "expected initial logical block reference tag" field found in the 32 byte cdb variants of WRITE, WRITE ATOMIC, WRITE SAME and WRITE STREAM. The field is also found in the WRITE SCATTERED(32) LBA range descriptors. It is a 32 bit field which means the maximum value is 0xffffffff. The default value is 0xffffffff.

-S, --same=NDOB

selects the WRITE SAME command with the NDOB field set to NDOB which stands for No Data-Out Buffer. NDOB can take values 0 or 1 (i.e. it is a single bit field). When --same=1 all options asso? ciated with the data-out buffer are ignored.

-q, --scat-file=SF

where SF is the name of an auxiliary file containing the scatter list for the WRITE SCATTERED command. If the --scat-raw option is also given then SF is assumed to contain both the parameter list header (32 bytes of zeros) followed by zero or more LBA range descriptors which are also 32 bytes long each. These com? ponents are as defined by SBC-4 (i.e. in binary with integers in big endian format). If the --scat-raw option is not given then a file of ACSII hexadecimal is expected as described in the SCATTERED FILE ASCII FORMAT section below.

If this option is given with the --combined=DOF option then this utility will exit with a syntax error. SF must not be "-", a way of stopping the user trying to redirect stdin.

-R, --scat-raw

this option only effects the way that the file named SF from the --scat-file=SF option for WRITE SCATTERED is interpreted. By de? fault (i.e. without this option), SF is parsed as ASCII hexadec? imal with blank lines and line contents from and including '#' to the end of line ignored. Hence it can contain comments and other indications. When this option is given, the file named SF is interpreted as binary. As binary it is assumed to contain 32 bytes of zeros (the WRITE SCATTERED parameter list header) fol? lowed by zero or more LBA range descriptors (which are 32 bytes each). If the --strict option is given the reserved field in

those two items are checked with any non zero bytes causing an
error.

-S, --scattered=RD

selects the WRITE SCATTERED command with RD being the number of
LBA range descriptors that will be placed in the data-out buf?
fer. If RD is zero then the logic will try and determine the
number of range descriptors by other means (e.g. by parsing the
file named by SF, if there is one). The LBA range descriptors
differ between the 16 and 32 byte cdb variants of WRITE SCAT?
TERED. In the 16 byte cdb variant the 32 byte LBA range descrip?
tor is made up of an 8 byte LBA, followed by a 4 byte num?
ber_of_blocks followed by 20 bytes of zeros. In the 32 byte
variant the LBA and number_of_blocks are followed by a RT (4
bytes), an AT (2 bytes) and a TM (2 bytes) then 12 bytes of ze?
ros.

This paragraph applies when RD is greater than zero. If RD is
less than the number of LBA range descriptors built from command
line options, from the --scat-file=SF option or decoded from IF
(when the --combined=DOF option is given) then RD takes prece?
dence; so RD is placed in the "Number of LBA Range Descriptors"
field in the cdb. If RD is greater than the number of LBA range
descriptors found from the provided data and options, then an
error is generated.

-T, --stream=ID

selects the WRITE STREAM command with the STR_ID field set to
ID.   ID can take values from 0 to 0xffff (i.e. it is a 16 bit
field).

-s, --strict

when this option is present, more things (e.g. that reserved
fields contain zeros) and any irregularities will terminate the
utility with a message to stderr and an indicative exit status.
While experimenting with these commands, especially WRITE SCAT?
TERED, it is recommended to use this option.

-t, --tag-mask=TM

where TM is the "logical block application tag mask" field found in the 32 byte cdb variants of WRITE, WRITE ATOMIC, WRITE SAME and WRITE STREAM. The field is also found in the WRITE SCATTERED(32) LBA range descriptors. It is a 16 bit field which means the maximum value is 0xffff. The default value is 0xffff.

-I, --timeout=TO

where TO is the command timeout value in seconds. The default value is 120 seconds. If NUM is large on slow media then these WRITE commands may require considerably more time than 120 sec? onds to complete.

-u, --unmap=U_A

where U_A is OR-ed bit values used to set the UNMAP and ANCHOR bit fields in the WRITE SAME (16 or 32) cdb. If U_A is 1 then the UNMAP bit field is set; if U_A is 2 then the ANCHOR bit field is set; if U_A is 3 then both the UNMAP and ANCHOR bit fields are set. The default value for both bit fields is clear (0); setting U_A to 0 will also clear both bit fields.

-v, --verbose

increase the degree of verbosity (debug messages). These mes? sages are usually written to stderr.

-V, --version

output version string then exit.

-w, --wrprotect=WPR

sets the WRPROTECT field (3 bits) in all sg_write_x commands apart from ORWRITE which has a 3 bit ORPROTECT field (and the synopsis shows OPR to highlight the difference). In all cases WPR is placed in that 3 bit field. The default value is zero which does not send any PI in the data-out buffer. WPR should be a value between 0 and 7.

SCATTERED FILE ASCII FORMAT

All commands in this utility can take a --scat-file=SF and that option can be seen as a replacement for the --lba=LBA[,LBA...] and

--num=NUM[,NUM...] options. if both the --scat-file=SF and --scat-raw
options are given then the file named SF is expected to be binary and
contain the parameter list header (32 bytes of zeros for both the 16
and 32 byte variants) followed by zero or more LBA range descriptors,
each of 32 bytes each. This section describes what is expected in SF
when the --scat-raw option is not given.

The ASCII hexadecimal "scatter file" (named by SF) can contain com?
ments, empty lines and numbers. If multiple numbers appear on one line
they can be separated by spaces, tabs or a single comma. Numbers are
parsed as decimal unless prefixed by "0x" (or "0X") or have a suffix of
"h". Ox is the prefix of hexadecimal number is the C language while T10
uses the "h" suffix for the same purpose. Anything from and including a
"#" character to the end-of-line is ignored, so comments can be placed
there.

For the WRITE SCATTERED (16) command, its LBA range descriptors contain
two items per descriptor: an 8 byte LBA followed by a 4 byte num?
ber_of_blocks. The remaining 20 bytes of the descriptor are zeros. The
format accepted is relatively loose with each decoded value being
placed in an LBA and then a number_of_blocks until the end-of-file is
reached. The pattern starts with a LBA and if it doesn't finish with a
number_of_blocks (i.e. an odd number of values are parsed) an error
occurs. So the number of LBA range descriptors generated will be half
the number of values parsed in SF.

For the WRITE SCATTERED (32) command, its LBA range descriptors contain
five items per descriptor: an 8 byte LBA followed by a 4 byte num?
ber_of_blocks, then a 4 byte RT, a 2 byte AT, and a 2 byte TM. The last
three items are associated with protection information (PI). The ac?
cepted format in the SF file is more constrained than the 16 byte cdb
variant. The items for each LBA range descriptor must be found on one
line with adjacent items being comma separated. The first two items
(LBA and number_of_blocks) must be given, and if no more items are on
the line then RT, AT and TM are given their default values (all "ff"
bytes). Spaces and tabs may appear between items but commas are the

separators. Two commas with no value between them will cause the "miss?

ing" item to receive its default value.

NOTES

Various numeric arguments (e.g. LBA) may include multiplicative suf?

fixes or be given in hexadecimal. See the "NUMERIC ARGUMENTS" section

in the sg3_utils(8) man page.

In Linux, prior to lk 3.17, the sg driver did not support cdb sizes

greater than 16 bytes. Hence a device node like /dev/sg1 which is asso?

ciated with the sg driver would fail with this utility if the --32 op?

tion was given (or implied by other options). The bsg driver with de?

vice nodes like /dev/bsg/6:0:0:1 does support cdb sizes greater than 16

bytes since its introduction in lk 2.6.28 .

EXIT STATUS

The exit status of sg_write_x is 0 when it is successful. Otherwise see

the sg3_utils(8) man page.

EXAMPLES

One simple usage is to write 4 blocks of zeros from (and including) a

given LBA according to the rules of WRITE ATOMIC with an atomic bound?

ary of 0. Since no cdb size option is given, the 16 byte cdb will be

assumed (i.e. WRITE ATOMIC(16)):

```
  sg_write_x --atomic=0 --in=/dev/zero --lba=0x1234 --num=4 /dev/sdc
```

Since --bs=BS has not been given, then this utility will call the READ

CAPACITY(16) command on /dev/sdc to determine the number of bytes in a

logical block. If the READ CAPACITY(16) command fails then the READ CA?

PACITY(10) command is tried. Let us assume one of them works and that

the number of bytes in each logical block is 512 bytes. So 4 blocks of

zeros (each block containing 512 bytes) will be written from (and in?

cluding) LBA 0x1234 . Now to bypass the need for the READ CAPACITY com?

mand(s) the --bs=BS option can be used:

```
  sg_write_x  --atomic=0  --bs=512  --in=/dev/zero --lba=0x1234 --num=4
/dev/sdc
```

Since --bs= is given and its value (512) is a power of 2, then the ac?

tual block size is also 512. If instead 520 was given then the logical block size would be 512 (the highest power of 2 less than 520) and the actual block size would be 520 bytes. To send the 32 byte variant add --32 as in:

```
  sg_write_x  --atomic=0  --32  --bs=512  --in=/dev/zero   --lba=0x1234
--num=4 /dev/sdc
```

For examples using 'sg_write_x --same=NDOB' see the manpage for sg_write_same(8). The syntax is a little different but the semantics are the same.

To send a WRITE STREAM(32) with a STR_ID of 1 use the following:

```
  sg_write_x   --stream=1  --32  --bs=512  --in=/dev/zero  --lba=0x1234
--num=4 /dev/sdc
```

Next is a WRITE SCATTERED(16) command with the scatter list, split be? tween the --lba= and --num= options, on the command line:

```
  sg_write_x    --scattered=2   --lba=2,0x33   --num=4,1  -i  /dev/zero
/dev/sg1
```

Example of a WRITE SCATTERED(16) command with a degenerate LBA range descriptor (first element to --lba= and --num=):

```
  sg_write_x    --scattered=2   --lba=0,0x33   --num=0,1  -i  /dev/zero
/dev/sg1
```

Example of a WRITE SCATTERED(16) command with the scatter list in scat_file.txt

```
  sg_write_x  --scattered=3 -q scat_file.txt -i /dev/zero /dev/sg1
```

Next a WRITE SCATTERED(16) command with its scatter list and data in a single file. Note that the argument to --scattered= is 0 so the number of LBA range descriptors is calculated by analyzing the first two blocks of scat_data.bin (because the argument to --combined= is 2) :

```
  sg_write_x  --scattered=0 --combined=2 -i scat_data.bin /dev/sg1
```

When the -xx option is used, a WRITE SCATTERED command is not executed but instead the contents of the data-out buffer are written to a file called sg_write_x.bin . In the case of WRITE SCATTERED that binary file is suitable for supplying to a later invocation to do the actual write to media. For example:

```
  sg_write_x  --scattered=3 -q scat_file.txt -xx -i /dev/zero /dev/sg1
```

```
Wrote 8192 bytes to sg_write_x.bin, LB data offset: 1

Number of LBA range descriptors: 3

  sg_write_x  --scattered=0 --combined=1 -i sg_write_x.bin /dev/sg1
```

Notice when the sg_write_x.bin is written (and nothing is written to

the media), a summary of what has happened is sent to stdout. The value

shown for "LB data offset:" (1) should be given to the --combined= op?

tion when the write to media actually occurs (i.e. the second invoca?

tion shown directly above).

AUTHORS

Written by Douglas Gilbert.

REPORTING BUGS

Report bugs to <dgilbert at interlog dot com>.

COPYRIGHT

SEE ALSO

sg_readcap,sg_vpd,sg_write_same,sg_stream_ctl(sg3_utils)