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

$ man sg_xcopy.8

SG_XCOPY(8) SG3_UTILS SG_XCOPY(8)

NAME

sg_xcopy - copy data to and from files and devices using SCSI EXTENDED COPY (XCOPY)

SYNOPSIS

sg_xcopy [bs=BS] [conv=CONV] [count=COUNT] [ibs=BS] [if=IFILE] [iflag=FLAGS] [obs=BS] [of=OFILE] [oflag=FLAGS] [seek=SEEK] [skip=SKIP] [--help] [--version] [app=0|1] [bpt=BPT] [cat=0|1] [dc=0|1] [fco=0|1] [id_usage={hold|dis? card|disable}] [list_id=ID] [prio=PRIO] [time=0|1] [verbose=VERB] [--on_dst|--on_src] [--verbose]

DESCRIPTION

Copy data to and from any files. Specialized for "files" that are Linux SCSI devices that support the SCSI EXTENDED COPY (XCOPY) command. This utility has similar syntax and semantics to dd(1) but with no "conversions" is supported.

The first group in the synopsis above are "standard" Unix dd(1) oper? ands. The second group are extra options added by this utility. Both

groups are defined below in combined, alphabetical order.

By default the XCOPY command is sent to OFILE. This can be changed with the --on_src or iflag=xflag options which cause the XCOPY command to be sent to IFILE instead. Also see the section on ENVIRONMENT VARIABLES.

In the SPC-4 standard the T10 committee has expanded the XCOPY command so that it now has two variants: "LID1" (for a List Identifier length of 1 byte) and "LID4" (for a List Identifier length of 4 bytes). This utility supports the older, LID1 variant which is also found in SPC-3 and earlier. While the LID1 variant in SPC-4 is command level (binary) compatible with XCOPY as defined in SPC-3, some of the command naming has changed. This utility uses the older, SPC-3 XCOPY names.

The ddpt utility supports the same xcopy(LID1) functionality as this utility with the same options and flags. Additionally ddpt supports a subset of xcopy(LID4) functionality variously called "xcopy version 2, lite" or ODX. ODX is a market name and stands for Offloaded Data Xfer (i.e. transfer).

OPTIONS

app={0|1}

if 1 start the destination of the copy at the end of OFILE. This assumes that OFILE is a regular file. The default is 0 in which case the destination of the copy starts at the beginning of OFILE (possibly offset be SEEK). This option cannot be used with the seek=SEEK option.

bpt=BPT

each IO transaction will be made using BPT blocks (or less if near the end of the copy). Default is 128 for logical block sizes less that 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 restric? tions). When cd/dvd drives are accessed, the logical block size is typically 2048 bytes and bpt defaults to 32 which again im? plies 64 KiB transfers.

bs=BS where BS must be the logical block size of the physical device

(if either the input or output files are accessed via SCSI com?
mands). Note that this differs from dd(1) which permits BS to be
an integral multiple. Defaults to the device logical block size.

cat={0|1}

sets the SCSI EXTENDED COPY command segment descriptor CAT bit to 0 or 1 (default: 0). The CAT bit (in conjunction with the PAD bit) controls the handling of residual data. See section HAN? DLING OF RESIDUAL DATA for details.

conv=CONV

all CONV arguments are ignored.

count=COUNT

copy COUNT blocks from IFILE to OFILE. Default is the minimum (IFILE if dc=0 or OFILE if dc=1) number of blocks that SCSI de? vices report from SCSI READ CAPACITY commands or that block de? vices (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 de? vice. 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 (or count=-1) and cannot be derived then an error message is issued and no copy takes place.

dc={0|1}

sets the SCSI EXTENDED COPY command segment descriptor DC bit to 0 or 1 (default: 0). The DC bit controls whether COUNT refers to the source (dc=0) or the target (dc=1) descriptor.

fco={0|1}

sets the SCSI EXTENDED COPY command segment descriptor FCO bit to 0 or 1 (default: 0). The Fast Copy Only (FCO) bit set will result in the copy being done but a technique faster than SCSI READ and WRITE commands. If the copy cannot but done in a faster manner then a sense key of "Copy aborted" with and addi?

tional sense of "Fast copy not possible" is returned.

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

id_usage={hold|discard|disable}

sets the SCSI EXTENDED COPY command parameter list field called LIST ID USAGE to 0 if the argument is 'hold', to 2 if the argu? ment is 'discard', or to '3' if the argument is 'disable'. If the device has the ability to hold data (as indicated by "held data limit" being greater than zero) then id_usage de? faults to 'hold' otherwise it defaults to 'discard'.

if=IFILE

read from IFILE instead of stdin. If IFILE is '-' then stdin is read. Starts reading at the beginning of IFILE unless SKIP is given.

iflag=FLAGS

where FLAGS is a comma separated list of one or more flags out? lined below. These flags are associated with IFILE and are ig? nored when IFILE is stdin.

list_id=ID

sets the SCSI EXTENDED COPY command parameter list field called LIST IDENTIFIER to ID. ID should be a value between 0 and 255 (inclusive). ID usually defaults to 1 unless id_usage=disable in which case it defaults to 0.

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

of=OFILE

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

oflag=FLAGS

where FLAGS is a comma separated list of one or more flags out? lined below. These flags are associated with OFILE and are ig?

nored when OFILE is /dev/null, '.' (period), or stdout.

prio=PRIO

sets the SCSI EXTENDED COPY command parameter list field called PRIORITY to PRIO. The default value is 1.

seek=SEEK

start writing SEEK bs-sized blocks from the start of OFILE. De? fault is block 0 (i.e. start of file).

skip=SKIP

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

time={0|1}

when 1, times transfer and does throughput calculation, out? putting 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 cdbs and responses for SCSI commands that are not repetitive (i.e. other that 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.

-h, --help

outputs usage message and exits.

--on_dst

send the XCOPY command to the output file/device (i.e. OFILE). This is the default unless overridden by the --on_src or iflag=xflag options. Also see the section below on ENVIRONMENT VARIABLES.

--on_src

send the XCOPY command to the input file/device (i.e. IFILE).

-v, --verbose

equivalent to verbose=1. When used twice, equivalent to ver?
bose=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 regular files this will lead to data appended to the end of any existing data. Cannot be used together with the seek=SEEK op? tion 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 ig? nored or may cause an error to be reported.

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

flock after opening the associated file (i.e. IFILE and/or OFILE) an attempt is made to get an advisory exclusive lock with the flock() system call. The flock arguments are "FLOCK_EX | FLOCK_NB" which will cause the lock to be taken if available else a "temporarily unavailable" error is generated. An exit status of 90 is produced in the latter case and no copy is done.

null has no affect, just a placeholder.

pad sets the SCSI EXTENDED COPY command segment descriptor PAD bit. The PAD bit (in conjunction with the CAT bit) controls the han? dling of residual data.(See section HANDLING OF RESIDUAL DATA for details.

xcopy has no affect; for compatibility with ddpt.

HANDLING OF RESIDUAL DATA

The pad and cat bits control the handling of residual data. As the data can be specified either in terms of source or target logical block size and both might have different block sizes residual data is likely to happen in these cases. If both logical block sizes are identical these

bits have no effect as residual data will not occur.

If none of these bits are set, the EXTENDED COPY command will be aborted with additional sense 'UNEXPECTED INEXACT SEGMENT'.

If only the cat bit is set the residual data will be retained and made available for subsequent segment descriptors. Residual data will be discarded for the last segment descriptor.

If the pad bit is set for the source descriptor only, any residual data for both source or destination will be discarded.

If the pad bit is set for the target descriptor only any residual source data will be handled as if the cat bit is set, but any residual destination data will be padded to make a whole block transfer.

If the pad bit is set for both source and target any residual source data will be discarded, and any residual destination data will be padded.

ENVIRONMENT VARIABLES

If the command line invocation does not explicitly (and unambiguously) indicate whether the XCOPY SCSI command should be sent to IFILE (i.e. the source) or OFILE (i.e. the destination) then a check is made for the presence of the XCOPY_TO_SRC and XCOPY_TO_DST environment vari? ables. If either one exists (but not both) then it indicates where the SCSI XCOPY command will be sent. By default the XCOPY command is sent to OFILE.

RETIRED OPTIONS

Here are some retired options that are still present:

append=0 | 1

when set, equivalent to 'oflag=append'. When clear the action is to overwrite the existing file (if it exists); this is the de? fault. See the 'append' flag.

NOTES

Copying data behind an Operating System's back can cause problems. In the case of Linux, users should look at this link: https://linux-mm.org/Drop_Caches

This command sequence may be useful:

sync; echo 3 > /proc/sys/vm/drop_caches

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

The COUNT, SKIP and SEEK arguments can take 64 bit values (i.e. very big numbers). Other values are limited to what can fit in a signed 32 bit number.

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.

If a device supports xcopy operations then it should set the 3PC field (3PC stands for Third Party Copy) in its standard INQUIRY response. This utility will attempt a xcopy operation irrespective of the value in the 3PC field but if it is zero (cleared) one would expect the xcopy operation to fail.

The status of the SCSI EXTENDED COPY command can be queried with sg_copy_results(sg3_utils)

Currently only block-to-block transfers are implemented; IFILE and OFILE must refer to a SCSI block device.

No account is taken of partitions so, for example, /dev/sbc2, /dev/sdc, /dev/sg2, and /dev/bsg/3:0:0:1 would all refer to the same thing: the whole logical unit (i.e. the whole disk) starting at LBA 0. So any par? tition indication (e.g. /dev/sdc2) is ignored. The user should set SKIP, SEEK and COUNT with information obtained from a command like 'fdisk -l -u /dev/sdc' to account for partitions.

XCOPY (LID1) capability has been added to the ddpt utility which is in a package of the same name. The ddpt utility will run on other OSes (e.g. FreeBSD and Windows) while sg_xcopy only runs on Linux. Also ddpt permits the arguments to ibs= and ibs= to be different.

EXAMPLES

Copy 2M of data from the start of one device to another:

sg_xcopy if=/dev/sdo of=/dev/sdp count=2048 list_id=2 dc=1

sg_xcopy: if=/dev/sdo skip=0 of=/dev/sdp seek=0 count=1024

Start of loop, count=1024, bpt=65535, lba_in=0, lba_out=0

sg_xcopy: 1024 blocks, 1 command

Check the status of the EXTENDED COPY command:

sg_copy_results --status --list_id=2 /dev/sdp

Receive copy results (copy status):

Held data discarded: Yes

Copy manager status: Operation completed without errors

Segments processed: 1

Transfer count units: 0

Transfer count: 0

SIGNALS

The signal handling has been borrowed from dd: SIGINT, SIGQUIT and SIG?

PIPE 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 sg_xcopy is 0 when it is successful. Otherwise see

the sg3_utils(8) man page.

An additional exit status of 90 is generated if the flock flag is given

and some other process holds the advisory exclusive lock.

AUTHORS

Written by Hannes Reinecke and Douglas Gilbert.

REPORTING BUGS

Report bugs to <dgilbert at interlog dot com>.

COPYRIGHT

SEE ALSO

There is a web page discussing sg_dd at

https://sg.danny.cz/sg/sg_dd.html

A POSIX threads version of this utility called sgp_dd is in the sg3_utils package. Another version from that package is called sgm_dd and it uses memory mapped IO to speed transfers from sg devices.

The lmbench package contains lmdd which is also interesting. For moving data to and from tapes see dt which is found at https://www.scsi?faq.org/RMiller_Tools/index.html

To change mode parameters that effect a SCSI device's caching and error recovery see sdparm(sdparm)

See also dd(1), sg_copy_results(sg3_utils), ddrescue(GNU), ddpt,ddptctl(ddpt)