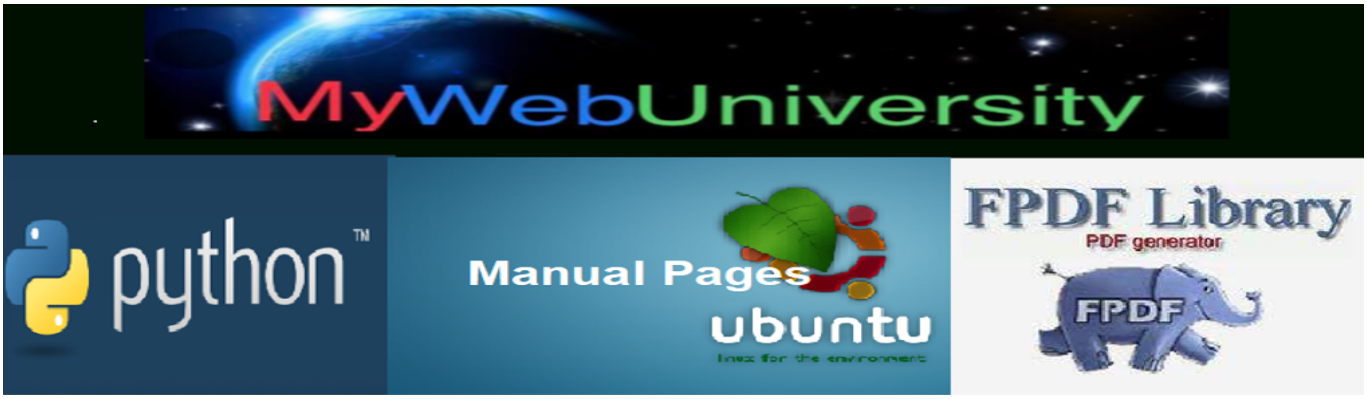

Rocky Enterprise Linux 9.2 Manual Pages on command 'sg_compare_and_write.8'

$ man sg_compare_and_write.8

COMPARE AND WRITE(8) SG3_UTILS COMPARE AND WRITE(8)

NAME

sg_compare_and_write - send the SCSI COMPARE AND WRITE command

SYNOPSIS

sg_compare_and_write [--dpo] [--fua] [--fua_nv] [--grpnum=GN] [--help] --in=IF [--inw=WF] --lba=LBA [--num=NUM] [--quiet] [--timeout=TO] [--verbose] [--version] [--wrprotect=WP] [--xferlen=LEN] DEVICE

DESCRIPTION

Send the SCSI COMPARE AND WRITE command to DEVICE. This utility fetches a compare buffer and a write buffer from either one or two files. If the --inw=WF option is given then the compare buffer is fetched from the file indicated by the --in=IF while the write buffer is fetched from the file indicated by the --inw=WF. If the --inw=WF option is not given then the concatenated compare and write buffers are fetched from the file indicated by the --in=IF option.

Those buffers are expected to each contain NUM blocks of data. The compare starts at logi? cal block address LBA on the DEVICE and if the comparison fails (i.e. the provided compare buffer does not equal the data at LBA on the DEVICE) then the COMPARE AND WRITE command finishes with a sense key of MISCOMPARE. In this case this utility will complete and set an exit status of 14 (which happens to be the sense key value of MISCOMPARE).

If the comparison succeeds then the provided write buffer is stored starting at LBA for NUM blocks on the DEVICE.

The actual number of bytes transferred in the data-out buffer of the COMPARE AND WRITE command may need to be given by the user with the --xferlen=LEN option. LEN defaults to (2

* NUM * 512) which is 1024 for the default NUM of 1. If the block size is other than 512 then the user will need to use --xferlen=LEN option. If protection information is given (indicated by a value of WP other than 0 (the default)) then for a NUM of 1 LEN should be 1040 . Note that the SCSI READ CAPACITY command is not performed by this utility (e.g. to find the block size).

The T10 definition of the SCSI COMPARE AND WRITE command requires that the DEVICE implement the compare and optional write as an uninterrupted series of actions. Depending on some other DEVICE settings a verify operation may occur prior to the compare.

When a mismatch occurs between the compare buffer and the blocks starting at LBA read from the DEVICE the sense buffer containing the MISCOMPARE sense key causes several messages to be sent to stderr (including the offset of the first byte mismatch). To suppress these messages use the --quiet option. With or without the --quiet option the exit status will be set to 14.

This command is defined in SBC-3 whose most recent revision is 36. SBC-3 and other SCSI documents can be found at http://www.t10.org .

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.

-d, --dpo

Set the DPO bit in the COMPARE AND WRITE CDB

-f, --fua

Set the FUA bit in the COMPARE AND WRITE CDB

-F, --fua_nv

Set the FUA_NV bit in the COMPARE AND WRITE CDB. This bit was removed in SBC-3 revision 35d and its position marked as "reserved".

-g, --grpnum=GN

where GN is the value to be placed in the group number field in the COMPARE AND WRITE CDB.

-h, --help

output the usage message then exit.

-i, --in=IF

read data (binary) from file named IF. This will either be the combined compare and write buffers (when the --inw=WF option is not given) or just the compare buffer

(when the --inw=WF option is given). If IF is '-' then stdin (e.g. a pipe) is read.

-C, --inc=IF

The same as the --in option.

-D, --inw=WF

read data (binary) from file named WF. This will the write buffer that will become the second half of the data-out buffer sent to the DEVICE associated with the COM? PARE AND WRITE command. Note that when this option is given then the --in=IF is ex? pected to hold the associated compare buffer.

-l, --lba=LBA

where LBA is the logical block address to start the COMPARE AND WRITE command. As? sumed to be in decimal unless prefixed with '0x' or has a trailing 'h'.

-n, --num=NUM

where NUM is the number of blocks, starting at LBA, to read and compare with the verify instance. And given a match, the NUM of blocks to write starting LBA. The default value for NUM is 1.

-q, --quiet

suppress the sense buffer messages associated with a MISCOMPARE sense key that would otherwise be sent to stderr. Still set the exit status to 14 which is the sense key value indicating a MISCOMPARE.

-t, --timeout=TO

where TO is the command timeout value in seconds. The default value is 60 seconds. If NUM is large (or zero) a WRITE SAME command may require considerably more time than 60 seconds to complete.

-v, --verbose

increase the degree of verbosity (debug messages).

-V, --version

output version string then exit.

-w, --wrprotect=WP

set the WRPROTECT field in the cdb to WP. The default value is 0 which implies no protection information is sent (along with the user data) by this utility.

-x, --xferlen=LEN

where LEN is the data out buffer length in byte. It defaults to (2 * NUM * 512) bytes. If the DEVICE block size is other than 512 bytes or WP is non-zero (implying

additional protection information) then this default will be incorrect; the use must supply the correct value for LEN

NOTES

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

EXAMPLES

Before overwriting the first two blocks of whatever (SCSI) storage device that is chosen, take a small backup. The logical block size is assumed to be 512 bytes. Take a copy (in backup01.bin) of the first two blocks::

```
# sg_dd if=/dev/sg1 bs=512 of=backup01.bin count=2
2+0 records in
2+0 records out
```

WARNING: if /dev/sg1 corresponds to a disk on your system that contains currently mounted file systems, do _not_ continue. If you can, unmount all file systems on that disk. If that is not possible, use another disk with no mounted file systems on it. In Linux the scsi_debug driver is a good candidate for experimentation.

Now fill the first block with 0xff bytes:

```
# sg_dd iflag=ff bs=512 of=/dev/sg1 count=1
1+0 records in
1+0 records out
```

and the second block with 0x0 bytes:

```
# sg_dd iflag=00 bs=512 seek=1 of=/dev/sg1 count=1
1+0 records in
1+0 records out
```

Now copy those two blocks into a file:

```
# sg_dd if=/dev/sg1 bs=512 of=ff00.bin count=2
2+0 records in
2+0 records out
```

Now we can do a compare and write command. It is told to compare the first block (i.e. LBA 0) with the first block in the given file (i.e. ff00.bin). If they are equal (they should be both full of 0xff bytes). Since the compare succeeds, it will write the second block in ff00.bin over LBA 0:

```
# sg_compare_and_write --in=ff00.bin --lba=0 --num=1 /dev/sg1
```

No news is good news. Now if we do that command again:

sg_compare_and_write --in=ff00.bin --lba=0 --num=1 /dev/sg1

Miscompare at byte offset: 0 [0x0]

sg_compare_and_write failed: Miscompare

This is expected. The first sg_compare_and_write ended up writing 0x0 bytes over LBA 0x0. The second sg_compare_and_write command compares LBA 0x0 with 0xff bytes and fails immedi? ately (i.e. at byte offset: 0). Now we will overwrite the first 3 bytes of ff00.bin with 0x0:

sg_dd bs=1 iflag=00 of=ff00.bin count=3

3+0 records in

3+0 records out

Notice the 'bs=1' operand. The dd utility (and thus sg_dd) is very useful for doing small binary edits on a file. Now if we do that sg_compare_and_write again, it still fails but with a small difference:

sg_compare_and_write --in=ff00.bin --lba=0 --num=1 /dev/sg1

Miscompare at byte offset: 3 [0x3]

sg_compare_and_write failed: Miscompare

So the bytes at offset 0, 1, and 2 compared equal but not the byte at offset 3. The SCSI COMPARE AND WRITE will stop on the first micompared byte.

EXIT STATUS

The exit status of sg_compare_and_write is 0 when it is successful. If the compare step fails then the exit status is 14. For other exit status values see the EXIT STATUS section in the sg3_utils(8) man page.

Earlier versions of this utility set an exit status of 98 when there was a MISCOMPARE.

AUTHORS

Written by Shahar Salzman. Maintained by Douglas Gilbert. Additions by Eric Seppanen.

REPORTING BUGS

Report bugs to shahar.salzman@kaminario.com or dgilbert@interlog.com

COPYRIGHT

ditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials pro? vided with the distribution.

* Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written per? mission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EX? PRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MER? CHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL Kami? nario Technologies LTD BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEG? LIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

SEE ALSO

sg_xcopy, sg_receive_copy_results(sg3_utils)