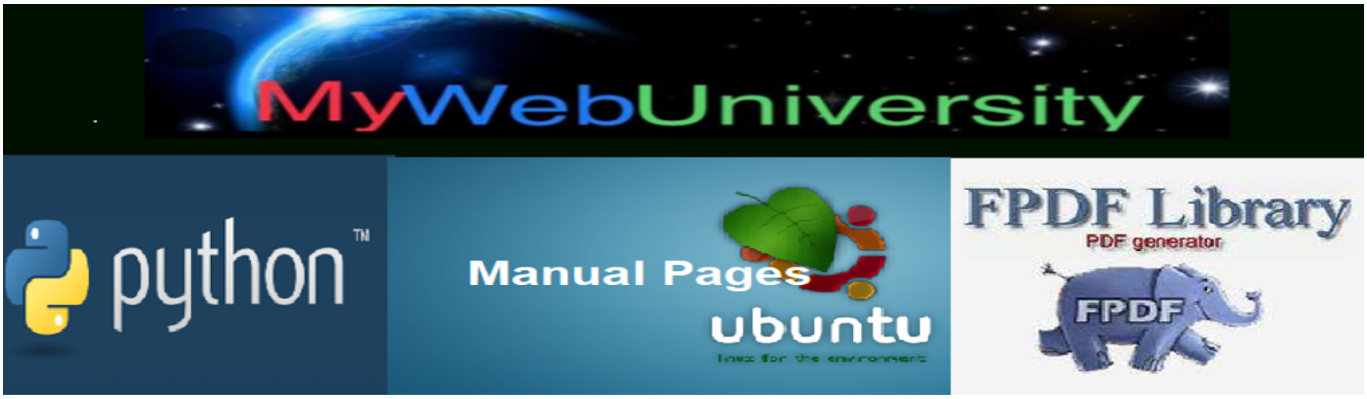




## *Rocky Enterprise Linux 9.2 Manual Pages on command 'sg_luns.8'*

***$ man sg_luns.8***

SG_LUNS(8) SG3_UTILS SG_LUNS(8)

NAME

sg_luns - send SCSI REPORT LUNS command or decode given LUN

SYNOPSIS

sg_luns [--decode] [--help] [--hex] [--linux] [--lu_cong] [--maxlen=LEN] [--quiet] [--raw]

[--readonly] [--select=SR] [--verbose] [--version] DEVICE

sg_luns --test=ALUN [--decode] [--hex] [--lu_cong] [--verbose]

DESCRIPTION

In the first form shown in the SYNOPSIS this utility sends the SCSI REPORT LUNS command to the DEVICE and outputs the response. The response should be a list of LUNs ("a LUN inven? tory") for the I_T nexus associated with the DEVICE. Roughly speaking that is all LUNs that share the target device that the REPORT LUNS command is sent through. This command is defined in the SPC-3 and SPC-4 SCSI standards and its support is mandatory. The most re? cent draft if SPC-6 revision 1.

When the --test=ALUN option is given (the second form in the SYNOPSIS), then the ALUN value is decoded as outlined in various SCSI Architecture Model (SAM) standards and recent drafts (e.g. SAM-6 revision 2, section 4.7) .

Where required below the first form shown in the SYNOPSIS is called "device mode" and the second form is called "test mode".

OPTIONS

Arguments to long options are mandatory for short options as well.

-d, --decode

decode LUNs into their component parts, as described in the LUN section of SAM-3,

SAM-4 and SAM-5.

[test mode] ALUN is decoded irrespective of whether this option is given or not. If this option is given once then the given ALUN is output in T10 preferred format (which is 8 pairs of hex digits, each separated by a space). If given twice then the given ALUN is output in an alternate T10 format made up of four quads of hex digits with each quad separated by a "-" (e.g. C101-0000-0000-0000).

-h, --help

output the usage message then exit.

-H, --hex

[device mode] when given once this utility will output the SCSI response (i.e. the data-out buffer) to the REPORT LUNS command in ASCII hex then exit. When given twice it causes --decode to output component fields in hex rather than decimal.

[test mode] when this option is given, then decoded component fields of ALUN are output in hex.

-l, --linux

this option is only available in Linux. After the T10 representation of each 64 bit LUN (in 16 hexadecimal digits), if this option is given then to the right, in square brackets, is the Linux LUN integer in decimal. If the --hex option is given twice (e.g. -HH) as well then the Linux LUN integer is output in hexadecimal.

-L, --lu_cong

this option is only considered with --decode. When given once then the list of LUNs is decoded as if the LU_CONG bit was set in each LU's corresponding INQUIRY re? sponse. When given twice the list of LUNs is decoded as if the LU_CONG bit was clear in each LU's corresponding INQUIRY response. When this option is not given and --decode is given then an INQUIRY is sent to the DEVICE and the setting of its LU_CONG bit is used to decode the list of LUNs.

[test mode] decode ALUN as if the LU_CONG bit is set in its corresponding standard INQUIRY response. In other words treat ALUN as if it is a conglomerate LUN. If not given (or given twice) then decode ALUN as if the LU_CONG bit is clear.

-m, --maxlen=LEN

where LEN is the (maximum) response length in bytes. It is placed in the cdb's "al? location length" field. If not given (or LEN is zero) then 8192 is used. The maxi? mum allowed value of LEN is 1048576.

-q, --quiet

output only the ASCII hex rendering of each report LUN, one per line. Without the --quiet option, there is header information printed before the LUN listing.

-r, --raw

output the SCSI response (i.e. the data-out buffer) in binary (to stdout).

-R, --readonly

open the DEVICE read-only (e.g. in Unix with the O_RDONLY flag). The default is to open it read-write.

-s, --select=SR

SR is placed in the SELECT REPORT field of the SCSI REPORT LUNS command. The de? fault value is 0. Hexadecimal values may be given with a leading "0x" or a trailing "h". For detailed information see the REPORT LUNS command in SPC (most recent is SPC-4 revision 37 in section 6.33). To simplify, for the I_T nexus associated with the DEVICE, the meanings of the SR values defined to date for SPC-4 are:

0 : most luns excluding well known logical unit numbers

1 : well known logical unit numbers

2 : all luns accessible to this I_T nexus

0x10 : only accessible administrative luns

0x11 : administrative luns plus non-conglomerate luns (see SPC-4)

0x12 : if DEVICE is an administrative LU, then report its lun plus its subsidiary luns

For SR values 0x10 and 0x11, the DEVICE must be either LUN 0 or the REPORT LUNS well known logical unit. Values between 0xf8 and 0xff (inclusive) are vendor specific, other values are reserved. This utility will accept any value between 0 and 255 (0xff) for SR .

-t, --test=ALUN

ALUN is assumed to be a hexadecimal number in ASCII hex or the letter 'L' followed by a decimal number (see below). The hexadecimal number can be up to 64 bits in size (i.e. 16 hexadecimal digits) and is padded to the right if less than 16 hexa? decimal digits are given (e.g. --test=0122003a represents T10 LUN: 01 22 00 3a 00 00 00 00). ALUN may be prefixed by '0x' or '0X' (e.g. the previous example could have been --test=0x0122003a). ALUN may also be given with spaces, tabs, or a '-' between each byte (or other grouping (e.g. c101-0000-0000-0000)). However in the case of space or tab separators the ALUN would need to be surrounded by single or

double quotes.

In the leading 'L' case the, following decimal number (hex if preceded by '0x') is assumed to be a Linux "word flipped" LUN which is converted into a T10 LUN repre? sentation and printed. In both cases the number is interpreted as a LUN and decoded as if the --decode option had been given. Also when ALUN is a hexadecimal number it can have a trailing 'L' in which case the corresponding Linux "word flipped" LUN value is output. The LUN is decoded in all cases.

The action when used with --decode is explained under that option.

-v, --verbose

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

-V, --version

print the version string and then exit.

NOTES

The SCSI REPORT LUNS command is important for Logical Unit (LU) discovery. After a target device is discovered (usually via some transport specific mechanism) and after sending an INQUIRY command (to determine the LU_CONG setting), a REPORT LUNS command should either be sent to LUN 0 (which is Peripheral device addressing method with bus_id=0 and tar? get/lun=0) or to the REPORT LUNS well known LUN (i.e. 0xc101000000000000). SAM-5 requires that one of these responds with an inventory of LUNS that are contained in this target de? vice.

In test mode, if the --hex option is given once then in the decoded output, some of the component fields are printed in hex with leading zeros. The leading zeros are to indicate the size of the component field. For example: in the Peripheral device addressing method (16 bits overall), the bus ID is 6 bits wide and the target/LUN field is 8 bits wide; so both are shown with two hex digits (e.g. bus_id=0x02, target=0x3a).

EXAMPLES

Typically by the time user space programs get to run, SCSI LUs have been discovered. In Linux the lsscsi utility lists the LUs that are currently present. The LUN of a device (LU) is the fourth element in the tuple at the beginning of each line. Below we see a tar? get (or "I_T Nexus": "6:0:0") has two LUNS: 1 and 49409. If 49409 is converted into T10 LUN format it is 0xc101000000000000 which is the REPORT LUNS well known LUN.

```
 # lsscsi -g
 [6:0:0:1]    disk    Linux    scsi_debug       0004  /dev/sdb   /dev/sg1
```

```
[6:0:0:2]    disk    Linux    scsi_debug      0004  /dev/sdc   /dev/sg2
[6:0:0:49409]wlun    Linux    scsi_debug      0004  -          /dev/sg3
```

We could send a REPORT LUNS command (with SR 0x0, 0x1 or 0x2) to any of those file device nodes and get the same result. Below we use /dev/sg1 :

```
# sg_luns /dev/sg1
Lun list length = 16 which imples 2 lun entry
Report luns [select_report=0x0]:
    0001000000000000
    0002000000000000
```

That is a bit noisy so cut down the clutter with --quiet:

```
# sg_luns -q /dev/sg1
0001000000000000
0002000000000000
```

Now decode that LUN into its component parts:

```
# sg_luns -d -q /dev/sg1
0001000000000000
      Peripheral device addressing: lun=1
0002000000000000
      Peripheral device addressing: lun=2
```

Now use --select=1 to find out if there are any well known LUNs:

```
# sg_luns -q -s 1 /dev/sg1
c101000000000000
```

So how many LUNs do we have all together (associated with the current I_T Nexus):

```
# sg_luns -q -s 2 /dev/sg1
0001000000000000
0002000000000000
c101000000000000
# sg_luns -q -s 2 -d /dev/sg1
0001000000000000
      Peripheral device addressing: lun=1
0002000000000000
      Peripheral device addressing: lun=1
c101000000000000
```

REPORT LUNS well known logical unit

The following example uses the --linux option and is not available in other operating sys? tems. The extra number in square brackets is the Linux version of T10 LUN shown at the start of the line.

```
# sg_luns -q -s 2 -l /dev/sg1
0001000000000000    [1]
0002000000000000    [2]
c101000000000000    [49409]
```

Now we use the --test= option to decode LUNS input on the command line (rather than send a REPORT LUNS command and act on the response):

```
# sg_luns --test=0002000000000000
Decoded LUN:
  Peripheral device addressing: lun=2
# sg_luns --test="c1 01"
Decoded LUN:
  REPORT LUNS well known logical unit
# sg_luns -t 0x023a004b -H
Decoded LUN:
  Peripheral device addressing: bus_id=0x02, target=0x3a
  >>Second level addressing:
    Peripheral device addressing: lun=0x4b
```

The next example is Linux specific as we try to find out what the Linux LUN 49409 trans? lates to in the T10 world:

```
# sg_luns --test=L49409
64 bit LUN in T10 preferred (hex) format:  c1 01 00 00 00 00 00 00
Decoded LUN:
  REPORT LUNS well known logical unit
```

And the mapping between T10 and Linux LUN representations can be done the other way:

```
# sg_luns -t c101L
Linux 'word flipped' integer LUN representation: 49409
Decoded LUN:
  REPORT LUNS well known logical unit
```

EXIT STATUS

The exit status of sg_luns is 0 when it is successful. Otherwise see the sg3_utils(8) man page.

## AUTHORS

Written by Douglas Gilbert.

## REPORTING BUGS

Report bugs to <dgilbert at interlog dot com>.

## COPYRIGHT



## SEE ALSO

sg_inq(8)