Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'lsscsi.8' command

$ man lsscsi.8

lsscsi(8) LSSCSI lsscsi(8)

NAME

lsscsi - list SCSI devices (or hosts), list NVMe devices

SYNOPSIS

lsscsi [--brief] [--classic] [--controllers] [--device] [--generic]

[--help] [--hosts] [--kname] [--list] [--long] [--long-unit] [--lunhex]

[--no-nvme] [--pdt] [--protection] [--protmode] [--scsi_id] [--size]

[--sysfsroot=PATH] [--sz-lbs] [--transport] [--unit] [--verbose]

[--version] [--wwn] [H:C:T:L]

DESCRIPTION

Uses information in sysfs (Linux kernel series 2.6 and later) to list

SCSI devices (or hosts) currently attached to the system. Many non-SCSI

storage devices (but not all) used the SCSI subsystem in Linux. In lss?

csi version 0.30 support was added to list NVMe devices.

In single line per device (LU or NVMe namespace) mode, the default,

each line starts with a 4 element tuple surrounded by square brackets.

For SCSI devices the first element 'H' is the host number, the second

element 'C' is the controller number, the third element 'T' is the tar?

get number and the final element is the Logical Unit Number (LUN). All

four are integers. For NVMe namespaces see two paragraphs down. When

the --hosts option is given for SCSI devices the tuple is reduced to

one element: the host number.

If a H:C:T:L tuple is given as an argument on the command line then it

acts as a filter and only devices that match it are listed. The colons don't have to be present, and '-', '*', '?' or missing components at the end are interpreted as wildcards. The default is '*:*:*:*' which means to match devices (i.e. Logical Units). Any filter string using '*' of '?' should be surrounded by single or double quotes to stop shell expansions. If '-' is used as a wildcard then the whole filter tuple should be prefixed by '-- ' to tell this utility there are no more options on the command line to be interpreted. A leading '[' and trailing ']' are permitted (e.g. '[1:0:0]' matches all LUNs on 1:0:0). May also be used to filter --hosts in which case only the H is active and may be either a number or in the form "host<n>" where <n> is a host number.

For NVMe devices and controllers almost all of the previous paragraph applies. The main difference is that "N" appears in the 'H' (first) po? sition. The 'C' position for NVMe is the controller's Linux generated "char" device minor number which is the first number that appears in a typical NVMe controller name, for example: "/dev/nvme2". The 'T' posi? tion for NVMe is the "CNTLID" value. The final 'L' position is the NVMe namespace identifier which is typically a sequential value starting at 1. The leading explicit "N" for NVMe devices is converted internally into a large value (32,767) that should not interfere with any Linux generated SCSI host number; it also means that the numeric sort used to show hosts (controllers) and devices (LUs or logical units) will always place NVMe devices and controllers after those that use the SCSI sub? system. To filter using a H:C:T:L argument for NVMe controllers, "hostN", "hostN:<num>", "N" or "N:<num>" may be used; when no "<num>" is given, only NVMe controllers will be listed (i.e. it lists no SCSI hosts (HBAs)).

By default in this utility device node names (e.g. "/dev/sda" or "/dev/root_disk") are obtained by noting the major and minor numbers for the listed device obtained from sysfs (e.g. the contents of "/sys/block/sda/dev") and then looking for a match in the "/dev" direc? tory. This "match by major and minor" will allow devices that have been

given a different name by udev (for example) to be correctly reported by this utility.

In some situations it may be useful to see the device node name that Linux would produce by default, so the --kname option is provided. An example of where this may be useful is kernel error logs which tend to report disk error messages using the disk's default kernel name.

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. Hyphenated long options can also take underscore, and vice versa (e.g. --scsi_id or --scsi-id are acceptable).

-b, --brief

reduces one line per device output to the tuple and the primary device name. This may simplify scripts that process the output of this utility. With the --generic option it will show on each line the tuple (from which the bsg pass-through device name can be deduced), the primary device name (which the block subsystem uses) and the sg device name (also a pass-through).

When the --pdt option is used together with this option the SCSI Peripheral Device Type (PDT) is displayed in hex (with a leading "0x") between the tuple and the primary device name. For NVMe namespaces "0x0" is displayed (for a disk or direct access de? vice).

-c, --classic

The output is similar to that obtained from 'cat /proc/scsi/scsi'

-C, --controllers

Lists NVMe controllers and SCSI hosts. This is a synonym for the --hosts option.

-d, --device

After outputting the (probable) SCSI device name the device node major and minor numbers are shown in brackets (e.g. "/dev/sda[8:0]").

-g, --generic

Output the SCSI generic device file name. Note that if the sg driver is a module it may need to be loaded otherwise '-' may appear.

NVMe does not have generic (char) devices in the same sense as SCSI. Instead NVMe Admin, NVM (i.e. block type commands such as Read and Write) and MI (Management Interface (e.g. to an enclo? sure)) commands are all sent to the containing controller whose device name is shown when the --hosts option is used.

To unclutter the single line per device mode the --brief option combined with this option should help.

-h, --help

Output the usage message and exit.

-H, --hosts

List the SCSI hosts and NVMe controllers currently attached to the system. If this option is not given (and the --controllers option is not given) then SCSI devices (logical units (LUs)) followed by NVMe devices (namespaces) are listed.

-k, --kname

Use Linux default algorithm for naming devices (e.g. block major 8, minor 0 is "/dev/sda") rather than the "match by major and minor" in the "/dev" directory as discussed above.

-L, --list

Output additional information in <attribute_name>=<value> pairs, one pair per line preceded by two spaces. This option has the same effect as '-lll'.

-l, --long

Output additional information for each SCSI device (host). Can be used multiple times for more output in which case the shorter option form is more convenient (e.g. '-lll'). When used three times (i.e. '-lll') outputs SCSI device (host) attributes one per line; preceded by two spaces; in the form "<attri? bute_name>=<value>".

-U, --long-unit

Output logical unit name in full, if available. It replaces the normal vendor, product and revision strings given in the single logical unit per line mode. If no logical unit name is found "none" is printed. If the logical unit name is long (e.g. a UUID) then following fields are pushed further to the right as required. This option is functionally equivalent to the '-uuu' option.

If the option is used twice (e.g. '-UU') then EUI, NAA, UUID and T10 vendor ID formats are prefixed by "eui.", "naa.", "uuid." and "t10." respectively. Note that SCSI name format used by iSCSI should already be prefixed by 'iqn.'. Using the '--unit' option 4 or more times (e.g. '-uuuu') will have the same action as '-UU'.

--long_unit is also an acceptable form when invoking this op? tion.

-x, --lunhex

when this option is used once the LUN in the tuple (at the start of each device line) is shown in "T10" format which is up to 16 hexadecimal digits. It is prefixed by "0x" to distinguish the LUN from the decimal value shown in the absence of this option. Also hierarchal LUNs are shown with a "_" character separating the levels. For example the two level LUN: 0x0355006600000000 will appear as 0x0355_0066. If this option is given twice (e.g. using the short form: '-xx') then the full 16 hexadecimal digits are shown for each LUN, prefixed by "0x".

For NVMe, the namespace identifier (nsid) is shown in the "L" position. The nsid is a 32 bit unsigned quantities with 0x0 and 0xffffffff reserved. Without this option, the nsid is shown in decimal. When this option is used once the nsid is output in hex with a lead 0x and with up to 3 leading zeros. When this option is used twice the nsid is output in hex with up to 7 leading ze? ros.

-N, --no-nvme

this option excludes NVMe devices and controllers for the out? put. This option may be needed to stop NVMe device output inter? fering with specific format output like that produced when the --classic option is used.

To only show NVMe devices, use 'lsscsi N', to only show NVMe controllers, use 'lsscsi -H N'.

-D, --pdt

this option displays the SCSI Peripheral Device Type (PDT) in hex preceded by "0x". For NVME namespaces "0x0' is displayed which corresponds to a disk ("Direct Access Device" or SSD). In single line output this hex PDT replaces the device type abbre? viation (e.g. "0x0 " replaces "disk ") and appears after the tuple.

-p, --protection

Output target (DIF) and initiator (DIX) protection types.

-P, --protmode

Output effective protection information mode for each disk de? vice.

-i, --scsi_id

outputs the udev derived matching id found in /dev/disk/by-id/scsi* . This is only for disk (and disk like) devices. If no match is found then "dm-uuid-mpath*" and "usb*" are searched in the same directory. If there is still no match then the /sys/class/block/<disk>/holders directory is searched. The matching id is printed following the device name (e.g. /dev/sdc) and if there is no match "-" is output. Note that only disk (like) devices are matched by this option; so, for example, a SCSI enclosure will have an identifier of '-'.

Prior to revision 164 the first match in /dev/disk/by-id/scsi* was printed. A change was added at that point to check for iden? tifiers in a specific order as some are considered "stronger" than others. See the ORDER OF SCSI IDENTIFIERS section below.

-s, --size

Print disk capacity in human readable form. When given once, normal base 10 SI units are used as a prefix for 'B' which is bytes (aka octets). For example MB, GB and TB stand for 10^6, 10^9 and 10^12 bytes respectively. When given twice, IEC 80000-3 prefixes for 'B' are used; for example MiB, GiB and TiB stand for 2^20, 2^30 and 2^40 bytes respectively. The output is rounded to 3 or less significant figures in order to fit on a single line. It will also output the size of RBC devices, CD/DVD media and host managed ZBC disks. Host aware ZBC disks have their "peripheral device type" set to 0 (the same as normal disks) so their size is output.

If given three times (short form is the more convenient: '-sss') then the disk capacity as a logical block count is given. This is an exact figure in decimal reported by the storage device at discovery. Discovery is typically just after boot time, or when it was last attached if the storage device is removable.

To unclutter the single line per device mode the --brief option combined with this option should help.

-y, --sysfsroot=PATH

assumes sysfs is mounted at PATH instead of the default '/sys' . If this option is given PATH should be an absolute path (i.e. start with '/').

-S, --sz-lbs

Print disk capacity as a number of logical blocks (which is the same as '-sss'). When used twice a comma is added followed by the logical block size in bytes. It should be a number like 512 or 4096.

If the logical block size cannot be found (e.g. because the ver? sion of Linux predates the /sys/block/<dev_name>/queue direc? tory) then the number of 512 byte blocks followed comma and then '512' is output irrespective of what the true logical block size of the device is. This special case action occurs whether this

option is given one or more times.

To unclutter the single line per device mode the --brief option combined with this option should help.

-t, --transport

Output transport information. This will be target related infor? mation or, if --hosts is given, initiator related information. When used without --list, a name or identifier (or both) are output on a single line, usually prefixed by the type of trans? port. For devices this information replaces the normal vendor, product and revision strings. When the --list option is also given then additionally multiple lines of attribute_name=value pairs are output, each indented by two spaces. See the section on transports below.

-u, --unit

Output logical unit name, if available. If this option is given once or twice, then the 30 character field where the vendor, product and revision strings are usually placed is expanded to 32 characters and replaced by the logical unit name. If no logi? cal unit name is found "none" is printed. The first found of the NAA, EUI-64 or SCSI name string is output unless a SCSI name string is found and the associated target port indicates the iSCSI protocol, in which case the SCSI name string is preferred. Finally if there is no match on the above and a T10 Vendor ID descriptor is found then it is used.

If the name cannot fit in the 32 character field then it is truncated to the right and a trailing '_' character is used to alert the reader to the truncation. The 32 character width is chosen since that is large enough to hold 16 byte NAA or EUI-64 identifiers. However SCSI name strings as used by iSCSI can be larger than that.

If this option is used twice then this field is also 32 charac? ter wide. If the logical unit name cannot fit then it will be truncated to the left and a leading '_' character is used to

alert the reader to the truncation.

If this option is used three times the whole logical unit name is output, followed by several spaces.

In order for this option to work, it needs a Linux kernel from and including 3.15 . It accesses the sysfs vpd_pg83 file for the device in question. Old SCSI and ATA (SATA) equipment may not provide this information. If it is provided by ATA (SATA) then it will be the WWN.

-v, --verbose

outputs directory names where information is found. Use multiple times for more output.

-V, --version

outputs version information then exits. If used once outputs to stderr; if used twice outputs to stdout and shortens the date to yyyymmdd numeric format. The first number in the version string is the release number.

-w, --wwn

outputs the WWN for disks instead of manufacturer, model and re? vision (or instead of transport information). The World Wide Name (WWN) is typically 64 bits long (16 hex digits) but could be up to 128 bits long. To indicate the WWN is hexadecimal, it is prefixed by "0x". The ATA/SATA WWN is referred to as LU name in SCSI jargon; hence this option is more or less superseded by the --unit and --long-unit options.

TRANSPORTS

This utility lists SCSI devices which are known as logical units (LU) in the SCSI Architecture Model (ref: SAM-5 at https://www.t10.org) or hosts when the --hosts option is given. A host is called an initiator in SAM-5. A SCSI command travels out via an initiator, across some transport to a target and then onwards to a logical unit. A target de? vice may contain several logical units. A target device has one or more ports that can be viewed as transport end points. Each FC and SAS disk is a single target that has two ports and contains one logical unit. If

both target ports on a FC or SAS disk are connected and visible to a machine, then lsscsi will show two entries. Initiators (i.e. hosts) also have one or more ports and some HBAs in Linux have a host entry per initiator port while others have a host entry per initiator device.

When the --transport option is given for devices (i.e. --hosts not given) then most of the information produced by lsscsi is associated with the target, or more precisely: the target port, through which SCSI commands pass that access a logical unit.

Typically this utility provides one line of output per "device" or host. Significantly more information can be obtained by adding the --list option. When used together with the --transport option, after the summary line, multiple lines of transport specific information in the form "<attribute_name>=<value>" are output, each indented by two spaces. Using a filter argument will reduce the volume of output if a lot of devices or hosts are present.

The transports that are currently recognized are: IEEE 1394, ATA, FC, iSCSI, SAS, SATA, SPI, SRP and USB.

For IEEE 1394 (a.k.a. Firewire and "SBP" when storage is involved), the EUI-64 based target port name is output when --transport is given, in the absence of the --hosts option. When the --hosts option is given then the EUI-64 initiator port name is output. Output on the summary line specific to the IEEE 1394 transport is prefixed by "sbp:".

To detect ATA and SATA devices a crude check is performed on the driver name (after the checks for other transports are exhausted). Based on the driver name either the ATA or SATA transport type is chosen. Output on the summary line is either "ata:" or "sata:". A search is made for an associated vpd_pg83 file in sysfs, if found it may contain the de? vice's WWN which is output if present. The WWN will not appear in Linux kernels before 3.15 and with old PATA and SATA devices. Most device and hosts flagged as "ata:" will use the parallel ATA transport (PATA). For SATA devices that are attached via a SAS expander, see the SAS para? graph below.

For Fibre Channel (FC) the port name and port identifier are output

when --transport is given. In the absence of the --hosts option these ids will be for the target port associated with the device (logical unit) being listed. When the --hosts option is given then the ids are for the initiator port used by the host. Output on the summary line specific to the FC transport is prefixed by "fc:". If FCoE (over Eth? ernet) is detected the prefix is changed to "fcoe:".

For iSCSI the target port name is output when --transport is given, in the absence of the --hosts option. This is made up of the iSCSI name and the target portal group tag. Since the iSCSI name starts with "iqn" no further prefix is used. When the --hosts option is given then only "iscsi:" is output on the summary line.

For Serial Attached SCSI the SAS address of the target port (or initia? tor port if --hosts option is also given) is output. This will be a naa-5 address. For SAS HBAs and SAS targets (such as SAS disks and tape drives) the SAS address will be world wide unique. For SATA disks at? tached to a SAS expander, the expander provides the SAS address by adding a non zero value to its (i.e. the expander's) SAS address (e.g. expander_sas_address + phy_id + 1). SATA disks directly attached to SAS HBAs seem to have an indeterminate SAS address. Output on the summary line specific to the SAS transport is prefixed by "sas:".

For SATA devices, see the paragraph above on ATA devices. As noted in the previous paragraph, SATA devices attached to SAS expanders will display a manufactured SAS transport address (manufactured by the ex? pander) rather than the SATA device's WWN.

For the SCSI Parallel Interface (SPI) the target port identifier (usu? ally a number between 0 and 15 inclusive) is output when --transport is given, in the absence of the --hosts option. When the --hosts option is given then only "spi:" is output on the summary line.

For the PCIe transport (a.k.a. PCI Express) there at two possible stor? age types: NVMe and SOP/PQI (SCSI over PCIe). There are very few exam? ples of the latter currently so this utility concentrates on NVMe. NVMe uses its own command set and not SCSI but has many things in common. Rather than re-invent everything currently in use that SCSI has accumu?

lated over nearly 40 years, NVMe is beginning to use some parts of SCSI. A recent example is the SES-3 standard for enclosure management which has been adopted by NVMe. In SCSI a SES device is a logical unit with a peripheral device type (PDT) of 0xd (for enclosure) so it will appear when the lsscsi utility is invoked without any options. In NVMe is seems that an enclosure with appear as attached to the management interface (MI) of a NVMe controller. This means it should appear when "lsscsi --hosts" is invoked. It is unclear whether such a NVMe con? troller can have any storage namespaces associated with it. The sg_ses utility (in the sg3_utils package) can then be given that NVMe con? troller's device name (e.g. /dev/nmve1).

When the --transport option is given, after "pcie" the NVMe con? troller's subsystem vendor id and device id are output, separated by a colon (e.g. "pcie 0x8086:0x390a").

For the SCSI RDMA Protocol (SRP) the IB (InfiniBand) port's GUID is given. As an example, it has a form like this: 0002:c903:00fa:abcd .

When a USB transport is detected, the summary line will contain "usb:" followed by a USB device name. The USB device name has the form "<b>-<p1>[.<p2>[.<p3>]]:<c>.<i>" where <b> is the USB bus number, <p1> is the port on the host. <p2> is a port on a host connected hub, if present. If needed <p3> is a USB hub port closer to the USB storage device. <c> refers to the configuration number while <i> is the inter? face number. There is a separate SCSI host for each USB (SCSI) target. A USB SCSI target may contain multiple logical units. Thus the same "usb: <device_name>" string appears for a USB SCSI host and all logical units that belong to the USB SCSI target associated with that USB SCSI host.

LUNS

For historical reasons and as used by several other Unix based Operat? ing Systems, Linux uses a tuple of integers to describe (a path to) a SCSI device (also know as a Logical Unit (LU)). The last element of that tuple is the so-called Logical Unit Number (LUN). And originally in SCSI a LUN was an integer, at first 3 bits long, then 8 then 16

bits. SCSI LUNs today (SAM-5 section 4.7) are 64 bits but SCSI stan? dards now consider a LUN to be an array of 8 bytes.

Up until 2013, Linux mapped SCSI LUNs to a 32 bit integer by taking the first 4 bytes of the SCSI LUN and ignoring the last 4 bytes. Linux treated the first two bytes of the SCSI LUN as a unit (a word) and it became the least significant 16 bits in the Linux LUN integer. The next two bytes of the SCSI LUN became the upper 16 bits in the Linux LUN in? teger. The rationale for this was to keep commonly used LUNs small Linux LUN integers. The most common LUN (by far) in SCSI LUN (hex) no? tation is 00 00 00 00 00 00 00 00 and this becomes the Linux LUN inte? ger 0. The next most common LUN is 00 01 00 00 00 00 00 00 and this be? comes the Linux LUN integer 1.

In 2013 it is proposed to increase Linux LUNs to a 64 bit integer by extending the mapping outlined above. In this case all information that is possible to represent in a SCSI LUN is mapped a Linux LUN (64 bit) integer. And the mapping can be reversed without losing information.

This version of the utility supports both 32 and 64 bit Linux LUN inte? gers. By default the LUN shown at the end of the tuple commencing each line is a Linux LUN as a decimal integer. When the --lunhex option is given then the LUN is in SCSI LUN format with the 8 bytes run together, with the output in hexadecimal and prefixed by '0x'. The LUN is decoded according to SAM-5's description and trailing zeros (i.e. digits to the right) are not shown. So LUN 0 (i.e. 00 00 00 00 00 00 00 00) is shown as 0x0000 and LUN 65 (i.e. 00 41 00 00 00 00 00 00) is shown as 0x0041.

If the --lunhex option is given twice then the full 64 bits (i.e. 16 hexadecimal digits) are shown.

If the --lunhex option is not given on the command line then the envi? ronment variable LSSCSI_LUNHEX_OPT is checked. If LSSCSI_LUNHEX_OPT is present then its associated value becomes the number of times the --lunhex is set internally. So, for example, 'LSSCSI_LUNHEX_OPT=2 lss? csi' and 'lsscsi -xx' are equivalent.

ORDER OF SCSI IDENTIFIERS

This section only applies to the --scsi_id option and only for disk

like devices (e.g. not tapes nor enclosures). There are potentially several SCSI identifiers and from revision 164 they are checked in the following order and only the first found is output.

The SCSI identifier preference order is:

3 : NAA based (Network address Authority)

2 : EUI-64 based

8 : SCSI name string (e.g. iSCSI: iqn.1998-01.com.zzware.iscsi:name1)

S : serial number from SCSI VPD page 0x80

1 : T10 Vendor Identifier

0 : Vendor Specific

Those numbers prefixing each entry are the SCSI 'Designator Types' found in the definition of the Device Identification VPD page (0x83) in SPC/-3, SPC/-4 and SPC/-5. The 'S' of course if not a number and it refers to a different VPD page: the Serial Number VPD page (0x80).

There is a more general --wwn option that should apply to almost all devices. The term "WWN" (world-wide name) comes from the ATA and NVMe standards and corresponds to the "Logical Unit (LU) name" in SCSI. The LU name in SCSI tends to change by transport. For SAS the LU name is the LU's NAA identifier.

Plus there the --unit and the --long-unit options that may be helpful in uniquely identifying storage devices.

EXAMPLES

Information about this utility including examples can also be found at: https://sg.danny.cz/scsi/lsscsi.html .

NOTES

Information for this command is derived from the sysfs file system, which is assumed to be mounted at /sys unless specified otherwise by the user. SCSI (pseudo) devices that have been detected by the SCSI mid level will be listed even if the required upper level drivers (i.e. sd, sr, st, osst or ch) have not been loaded. If the appropriate upper level driver has not been loaded then the device file name will appear as '-' rather than something like '/dev/st0'. Note that some devices (e.g. scanners and medium changers) do not have a primary upper level

driver and can only be accessed via a SCSI generic (sg) device name.

Generic SCSI devices can also be accessed via the bsg driver in Linux.

By default, the bsg driver's device node names are of the form

'/dev/bsg/H:C:T:L'. So, for example, the SCSI device shown by this

utility on a line starting with the tuple '6:0:1:2' could be accessed

via the bsg driver with the '/dev/bsg/6:0:1:2' device node name.

lsscsi version 0.21 or later is required to correctly display SCSI de?

vices in Linux kernel 2.6.26 (and possibly later) when the CON?

FIG_SYSFS_DEPRECATED_V2 kernel option is not defined.

AUTHOR

Written by Doug Gilbert

REPORTING BUGS

Report bugs to <dgilbert at interlog dot com>.

COPYRIGHT

SEE ALSO

lspci lsusb lsblk