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

$ man sg_persist.8

SG_PERSIST(8)   SG3_UTILS   SG_PERSIST(8)

NAME

    sg_persist - use SCSI PERSISTENT RESERVE command to access registra?

    tions and reservations

SYNOPSIS

    sg_persist [OPTIONS] DEVICE

    sg_persist [OPTIONS] --device=DEVICE

    sg_persist --help | --version

DESCRIPTION

    This utility allows Persistent reservations and registrations to be

    queried and changed. Persistent reservations and registrations are

    queried by sub-commands (called "service actions" in SPC-4) of the SCSI

    PERSISTENT RESERVE IN (PRIN) command. Persistent reservations and reg?

    istrations are changed by sub-commands of the SCSI PERSISTENT RESERVE

    OUT (PROUT) command.

    There is a two stage process to obtain a persistent reservation. First

    an application (an I_T nexus in standard's jargon) must register a

    reservation key. If that is accepted (and it should be unless some

    other I_T nexus has registered that key) then the application can try

    and reserve the device. The reserve operation must specify the reser?

    vation key and a "type" (see the --prout-type=TYPE option).

    It is relatively safe to query the state of Persistent reservations and

    registrations. With no options this utility defaults to the READ KEYS

sub-command of the PRIN command. Other PRIN sub-commands are READ RESERVATION, REPORT CAPABILITIES and READ FULL STATUS. Before trying to change Persistent reservations and registrations users should be aware of what they are doing. The relevant sections of the SCSI Primary Commands document (i.e. SPC-5 whose most recent draft is revision 18 dated 4 January 2018) are sections 5.14 (titled "Reserva‐tions"), 6.16 (for the PRIN command) and 6.17 (for the PROUT command).

To safeguard against accidental use, the --out option must be given when a PROUT sub-command (e.g. --register) is used.

The older SCSI RESERVE and RELEASE commands (both 6 and 10 byte vari‐ants) are not supported by this utility. In SPC-3, RESERVE and RELEASE are deprecated, replaced by Persistent Reservations. RESERVE and RE‐LEASE have been removed from SPC-4 and Annex B is provided showing how to convert to persistent reservation commands. See a utility called 'scsires' for support of the SCSI RESERVE and RELEASE commands.

The DEVICE is required by all variants of this utility apart from --help. The DEVICE can be given either as an argument (typically but not necessarily the last one) or via the --device=DEVICE option.

SPC-4 does not use the term "sub-command". It uses the term "service action" for this and for part of a field's name in the parameter block associated with the PROUT command (i.e. "service action reservation key"). To lessen the potential confusion the term "sub-command" has been introduced.

OPTIONS

Arguments to long options are mandatory for short options as well. The following options are sorted in alphabetical order, based on their long option name.

-l, --alloc-length=LEN

specify the allocation length of the PRIN command. LEN is a hex value. By default this value is set to the size of the data-in buffer (8192). This parameter is of use for verification that response to PRIN commands with various allocation lengths is per section 4.3.5.6 of SPC-4 revision 18. Valid LEN values are

0-8192.

-C, --clear

    Clear is a sub-command of the PROUT command. It releases the

    persistent reservation (if any) and clears all registrations

    from the device. It is required to supply a reservation key that

    is registered for this I_T_L nexus (identified by

    --param-rk=RK).

-d, --device=DEVICE

    DEVICE to send SCSI commands to. The DEVICE can either be pro?

    vided via this option or via a freestanding argument. For exam?

    ple, these two: 'sg_persist --device=/dev/sg2' and 'sg_persist

    /dev/sg2' are equivalent.

-h, --help

    output a usage message showing main options. Use twice (e.g.

    '-hh') for the other option and more help.

-H, --hex

    the response to a valid PRIN sub-command will be output in hexa?

    decimal. By default (i.e. without this option) if the PRIN

    sub-command is recognised then the response will be decoded as

    per SPC-4. May be used more than once for more hex and less

    text.

-i, --in

    specify that a SCSI PERSISTENT RESERVE IN command is required.

    This is the default.

-m, --maxlen=LEN

    LEN is used as the ALLOCATION LENGTH field of the PRIN command.

    LEN is by default a decimal value. To give a hex value use a

    '0x' or '0X' prefix, or use a 'h' (or 'H') suffix. Can also take

    multipliers, see --maxlen=LEN option in the sg3_utils manual

    page.

    This option is the same as --alloc-length=LEN option apart from

    the representation of LEN. The option defaults to decimal while

    --alloc-length=LEN only takes hex.

-n, --no-inquiry

the default action is to do a standard SCSI INQUIRY command and
output make, product and revision strings plus the peripheral
device type prior to executing a PRIN or PROUT command. With
this option the INQUIRY command is skipped.

-o, --out

specify that a SCSI PERSISTENT RESERVE OUT command is required.

-Y, --param-alltgpt

set the 'all target ports' (ALL_TG_PT) flag in the parameter
block of the PROUT command. Only relevant for 'register' and
'register and ignore existing key' sub-commands.

-Z, --param-aptpl

set the 'activate persist through power loss' (APTPL) flag in
the parameter block of the PROUT command. Relevant for 'regis‐
ter', 'register and ignore existing key' and 'register and move'
sub-commands.

-K, --param-rk=RK

specify the reservation key found in the parameter block of the
PROUT command. RK is assumed to be hex (up to 8 bytes long). De‐
fault value is 0. This option is needed by most PROUT sub-com‐
mands.

-S, --param-sark=SARK

specify the service action reservation key found in the parame‐
ter block of the PROUT command. SARK is assumed to be hex (up to
8 bytes long).   Default value is 0. This option is needed by
some PROUT sub-commands.

-P, --preempt

Preempt is a sub-command of the PROUT command. Preempts the ex‐
isting persistent reservation (identified by --param-sark=SARK)
with the registration key that is registered for this I_T_L
nexus (identified by --param-rk=RK). If a new reservation is es‐
tablished as a result of the preemption then the supplied
--prout-type=TYPE is used as the type for this new reservation.

-A, --preempt-abort

> Preempt and Abort is a sub-command of the PROUT command. Pre? empts the existing persistent reservation (identified by --param-sark=SARK) with the registration key that is registered for this I_T_L nexus (identified by --param-rk=RK). If a new reservation is established as a result of the preemption then the supplied --prout-type=TYPE is used as the type for this new reservation. ACA and other pending tasks are aborted.

-T, --prout-type=TYPE

> specify the PROUT command's 'type' argument. Required by the 'register-move', 'reserve', 'release' and 'preempt (and abort)' sub-commands. Valid TYPE values: 1-> write exclusive, 3-> exclu? sive access, 5-> write exclusive - registrants only, 6-> exclu? sive access - registrants only, 7-> write exclusive - all regis? trants, 8-> exclusive access - all registrants. Default value is 0 (which is an invalid type). Each "persistent reservation type" is explained in more detail in a subsection of that name in the read reservation section of the PRIN command (section 6.15.3.3 of SPC-4 revision 37).

-s, --read-full-status

> Read Full Status is a sub-command of the PRIN command. For each registration with the given SCSI device, it lists the reserva? tion key and associated information. TransportIDs, if supplied in the response, are decoded.

-k, --read-keys

> Read Keys is a sub-command of the PRIN command. Lists all the reservation keys registered (i.e. registrations) with the given SCSI device. This is the default sub-command for the SCSI PRIN command.

-y, --readonly

> Open DEVICE read-only. May be useful with PRIN commands if there are unwanted side effects with the default read-write open. When given twice is interpreted as forcing a read-write open thus

overriding the SG_PERSIST_IN_RDONLY environment variable if present. See the ENVIRONMENT VARIABLES section for more.

-r, --read-reservation

Read Reservation is a sub-command of the PRIN command. List in? formation about the current holder of the reservation on the DE? VICE. If there is no current reservation this will be noted. In? formation about the current holder of the reservation includes its reservation key, scope and type.

-s, --read-status

same as --read-full-status.

-G, --register

Register is a sub-command of the PROUT command. It has 3 differ? ent actions depending on associated parameters. a) add a new registration with '--param-rk=0' and '--param-sark=<new_rk>'; b) Change an existing registration with '--param-rk=<old_rk>' and '--param-sark=<new_rk>'; or c) Delete an existing registration with '--param-rk=<old_rk>' and '--param-sark=0'.

-I, --register-ignore

Register and Ignore Existing Key is a sub-command of the PROUT command. Similar to --register except that when changing a reservation key the old key is not specified. The '--param-sark=<new_rk>' option should also be given.

-M, --register-move

register (another initiator) and move (the reservation held by the current initiator to that other initiator) is a sub-command of the PROUT command. It requires the transportID of the other initiator. [The standard uses the term I_T nexus but the point to stress is that there are two initiators (the one sending this command and another one) but only one logical unit.] The --prout-type=TYPE and --param-rk=RK options need to match that of the existing reservation while --param-sark=SARK option spec? ifies the reservation key of the new (i.e. destination) regis? tration.

-Q, --relative-target-port=RTPI

    relative target port identifier that reservation is to be moved

    to by PROUT 'register and move' sub-command. RTPI is assumed to

    be hex in the range 0 to ffff inclusive. Defaults to 0 .

-L, --release

    Release is a sub-command of the PROUT command. It releases the

    current persistent reservation. The   --prout-type=TYPE   and

    --param-rk=RK options, matching the reservation, must also be

    specified.

-z, --replace-lost

    Replace Lost Reservation is a sub-command of the PROUT command.

    It "begins a recovery process for the lost persistent reserva?

    tion that is managed by application clients". It also stops the

    device server terminating commands due to a lost persistent

    reservation. Options should be be '--param-rk=0' (or not given),

    '--param-sark=<new_rk>' and --prout-type=TYPE.

-c, --report-capabilities

    Report Capabilities is a sub-command of the PRIN command. It

    lists information about the aspects of persistent reservations

    that the DEVICE supports.

-R, --reserve

    Reserve is a sub-command of the PROUT command. It creates a new

    persistent reservation (if permitted). The --prout-type=TYPE and

    --param-rk=RK options must also be specified.

-X, --transport-id=TIDS

    The TIDS argument can take one of several forms. It can be a

    comma (or single space) separated list of ASCII hex bytes repre?

    senting a single TransportID as defined in SPC-4. They are usu?

    ally 24 bytes long apart from in iSCSI. The TIDS argument may be

    a   transport specific form (e.g. "sas,5000c50005b32001" is

    clearer than an equivalent to the hex byte form:

    "6,0,0,0,5,0,c5,0,5,b3,20,1"). The TIDS argument may be "-" in

    which case one or more TransportIDs can be read from stdin. The

TIDS argument may be of the form "file=<name>" in which case one
or more TransportIDs can be read from a file called <name>. See
the "TRANSPORT IDs" section below for more information.

-U, --unreg

optional when the PROUT register and move sub-command is in‐
voked. If given it will unregister the current initiator (I_T
nexus) after the other initiator has been registered and the
reservation moved to it. When not given the initiator (I_T
nexus) that sent the PROUT command remains registered.

-v, --verbose

print out cdb of issued commands prior to execution. If used
twice prints out the parameter block associated with the PROUT
command prior to its execution as well. If used thrice decodes
given transportID(s) as well. To see the response to a PRIN com‐
mand in low level form use the --hex option.

-V, --version

print out version string. Ignore all other parameters.

-?   output usage message. Ignore all other parameters.

TRANSPORT IDs

TransportIDs are used in persistent reservations to identify initia‐
tors. The format of a TransportID differs depending on the type of
transport being used. Their format is described in SPC-4 (in draft re‐
vision 37 see section 7.6.4).

A TransportID is required for the PROUT 'register and move' sub-command
and the PROUT 'register' sub-command can have zero, one or more Trans‐
portIDs.

When the --transport-id=TIDS option is given then the TIDS argument may
be a comma (or single space) separated list of ASCII hex bytes that
represent a single TransportID as defined in SPC-4. Alternatively the
TIDS argument may be a transport specific string starting with either
"fcp,", "spi,", "sbp,", "srp,", "iqn", "sas," or "sop,". The "iqn" form
is an iSCSI qualified name. Apart from "iqn" the other transport spe‐
cific leadin string may be given in upper case (e.g. "FCP,").

The "fcp," form should be followed by 16 ASCII hex digits that repre?
sent an initiator's N_PORT_NAME (e.g. "fcp,10000000C9F3A571"). The
"spi," form should be followed by   "<scsi_address>,<relative_tar?
get_port_identifier>" (both decimal numbers). The "sbp," form should be
followed by 16 ASCII hex digits that represent an initiator's EUI-64
name. The "srp," form should be followed by 32 ASCII hex digits that
represent an initiator port identifier. The "sas," form should be fol?
lowed by 16 ASCII hex digits that represent an initiator's port SAS ad?
dress (e.g. "sas,5000c50005b32001"). The "sop," form takes a hex number
that represents a routing id.

There are two iSCSI qualified name forms. The shorter form contains the
iSCSI name of the initiator port (e.g. "iqn.5886.com.acme.diskar?
rays-sn-a8675309"). The longer form adds the initiator session id (ISID
in hex) separated by ",i,0x". For example "iqn.5886.com.acme.diskar?
rays-sn-a8675309,i,0x1234567890ab".   On the command line to stop punc?
tuation in an iSCSI name being (mis)-interpreted by the shell, putting
the option argument containing the iSCSI name in double quotes is ad?
vised. iSCSI names are encoded in UTF-8 so if non (7 bit) ASCII charac?
ters appear in the iSCSI name on the command line, there will be diffi?
culties if they are not encoded in UTF-8. The locale can be changed
temporarily by prefixing the command line invocation of sg_persist with
"LANG=en_US.utf-8" for example.

Alternatively the TIDS argument may specify a file (or pipe) from which
one or more TransportIDs may be read. If the TIDS argument is "-" then
stdin (standard input) is read. If the TIDS argument is of the form
"file=<name>" then a file called <name> is read. A valid SPC-4 Trans?
portID is built from the transport specific string outlined in the pre?
vious paragraphs. The parsing of the data read is relatively simple.
Empty lines are ignored. Everything from and including a "#" on a line
is ignored. Leading spaces and tabs are ignored. There can be one
transportID per line. The transportID can either be a comma, space or
tab separated list of ASCII hex bytes that represent a TransportID as
defined in SPC-4. Padding with zero bytes to a minimum length of 24

bytes is performed if necessary. The transportID may also be transport

specific string type discussed above.

In SPC-3 the SPEC_I_PT bit set to one and TransportIDs were allowed for

the PROUT register and ignore existing key sub-command. In SPC-4 that

is disallowed yielding a CHECK CONDITION status with and ILLEGAL RE?

QUEST sense key and an additional sense code set to INVALID FIELD IN

PARAMETER LIST.

NOTES

In the 2.4 series of Linux kernels the DEVICE must be a SCSI generic

(sg) device. In the 2.6 series any SCSI device name (e.g. /dev/sdc,

/dev/st1m or /dev/sg3) can be specified. For example "sg_persist

--read-keys /dev/sdb" will work in the 2.6 series kernels.

The only scope for PROUT commands supported in the current draft of

SPC-4 is "LU_SCOPE". Hence there seems to be no point in offering an

option to set scope to another value.

Most errors with the PROUT sub-commands (e.g. missing or mismatched

--prout-type=TYPE) will result in a RESERVATION CONFLICT status. This

can be a bit confusing when you know there is only one (active) initia?

tor: the "conflict" is with the SPC standard, not another initiator.

Some recent disks accept some PRIN and PROUT sub-commands when the me?

dia is stopped. One exception was setting the APTPL flag (with the

--param-aptpl option) during a key register operation, it complained if

the disk one stopped. The error indicated it wanted the disk spun up

and when that happened, the registration was successful.

EXAMPLES

These examples use Linux device names. For suitable device names in

other supported Operating Systems see the sg3_utils(8) man page.

Due to the various option defaults the simplest example executes the

'read keys' sub-command of the PRIN command:

  sg_persist /dev/sdb

This is the same as the following (long-winded) command:

  sg_persist --in --read-keys --device=/dev/sdb

To read the current reservation either the '--read-reservation' form or

the shorter '-r' can be used:

   sg_persist -r /dev/sdb

To register the new reservation key 0x123abc the following could be

used:

   sg_persist --out --register --param-sark=123abc /dev/sdb

Given the above registration succeeds, to reserve the DEVICE (with type

'write exclusive') the following could be used:

   sg_persist --out --reserve --param-rk=123abc

       --prout-type=1 /dev/sdb

To release the reservation the following can be given (note that the

--param-rk and --prout-type arguments must match those of the reserva?

tion):

   sg_persist --out --release --param-rk=123abc

       --prout-type=1 /dev/sdb

Finally to unregister a reservation key (and not effect other registra?

tions which is what '--clear' would do) the command is a little sur?

prising:

   sg_persist --out --register --param-rk=123abc /dev/sdb

Now have a close look at the difference between the register and unreg?

ister examples above.

An example file that is suitably formatted to pass transportIDs via a

'--transport-id=file=transport_ids.txt' option can be found in the ex?

amples sub-directory of the sg3_utils package. There is also a simple

test script called sg_persist_tst.sh in the same directory.

The above sequence of commands was tested successfully on a Seagate

Savvio 10K.3 disk and a 1200 SSD both of which have SAS interfaces.

EXIT STATUS

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

the sg3_utils(8) man page.

ENVIRONMENT VARIABLES

Currently there is one recognised environment variable: SG_PER?

SIST_IN_RDONLY. If present and only if a PRIN command has been se?

lected then the given DEVICE is opened read-only (e.g. in Unix that is

with the O_RDONLY flag). See the --readonly option.

AUTHOR

Written by Douglas Gilbert

REPORTING BUGS

Report bugs to <dgilbert at interlog dot com>.

COPYRIGHT

Copyright ? 2004-2018 Douglas Gilbert

This software is distributed under the GPL version 2. There is NO war?

ranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PUR?

POSE.

SEE ALSO

sg3_utils(sg3_utils), scsires(internet)

sg3_utils-1.43   June 2018   SG_PERSIST(8)