NAME

sg_wr_mode − write (modify) SCSI mode page

SYNOPSIS

sg_wr_mode [−−*contents=H,H...*] [−−*dbd*] [−−*force*] [−−*help*] [−−*len=10|6*] [−−*mask=M,M...*] [−−*page=PG_H[,SPG_H]*] [−−*rtd*] [−−*save*] [−−*six*] [−−*verbose*] [−−*version*] *DEVICE*

DESCRIPTION

Writes a modified mode page to *DEVICE*. Uses the SCSI MODE SENSE (6 or 10 byte variant) command to fetch the existing mode data which includes a mode page (or subpage). It then combines that with the contents, potentially masked, and writes the modified mode page with the SCSI MODE SELECT (6 or 10 byte variant) command. This utility does not modify the block descriptor(s); if any block descriptors are fetched by the MODE SENSE command then the same block descriptors are written back with the following MODE SELECT command.

If the −−*rtd* option is given then most other options apart from −−*save*, −−*len*=10|6 and −−*six* are ignored. In this case only a MODE SELECT command is sent to the *DEVICE* with the RTD bit (Revert To Defaults) set. This bit was added to this command in SPC−5 revision 11, so older devices may not support it. The Extended Inquiry VPD page has the RTD_SUP bit to indicate whether the *DEVICE* supports the RTD bit in the MODE SELECT(6 and 10) commands. When the −−*rtd* option is given the rest of this section can be ignored.

If a contents argument is not given then the various components (i.e. header, block descriptor(s) and mode page) of the "current" values of the existing mode page are printed out. In this case the mode page is not altered on the device.

If the contents are specified, and a mask is not specified, then the contents must match the existing mode page in various aspects unless the −−*force* option is given. These include length, mode page code and subpage code if applicable. If all is well then the contents string is written to *DEVICE* as the new mode page.

If both contents and mask strings are specified then only bit positions in the contents corresponding to set bits in the mask are taken while the existing mode page supplies bit positions corresponding to clear bits. When a mask is given then the mask and/or the contents may be shorter than the existing mode page. If the mask is shorter than the contents then the remaining bytes are taken from the contents. If the contents are shorter than the existing mode page then the remaining bytes are taken from the existing mod page.

The force option allows the contents string to be written as the new mode page without any prior checks on the existing mode page. This should only be required for vendor specific mode pages. The existing mode data is ignored apart from the block descriptors which can be suppressed with the −−*dbd* option if need be.

Changing individual fields in a mode page is probably more easily done with the sdparm utility. Fields can be identified by acronym or by a numerical descriptor.

OPTIONS

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

−c, **−−contents**=*H,H...*

where *H,H...* is a string of comma separated hex numbers each of which should resolve to a byte value (i.e. 0 to ff inclusive). A (single) space separated string of hex numbers is also allowed but the list needs to be in quotes. This is the new contents of the mode page to be written to *DEVICE*, potentially filtered by the mask string.

−c, **−−contents**=-

reads contents string from stdin. The hex numbers in the string may be comma, space, tab or linefeed (newline) separated. If a line contains "#" then the remaining characters on that line are ignored. Otherwise each non separator character should resolve to a byte value (i.e. 0 to ff inclusive). This forms the new contents of the mode page to be written to *DEVICE*, potentially filtered by the mask string.

−d, **−−dbd**

disable block descriptors (DBD flag in cdb). Some device types include block descriptors in the mode data returned by a MODE SENSE command. If so the same block descriptors are written by

the MODE SELECT command. This option instructs the MODE SENSE command not to return any block descriptors. This would be a sensible default for this utility apart from the fact that not all SCSI devices support the DBD bit in the cdb.

-f, **--force**
force the contents string to be taken as the new mode page, or at least doesn't do checks on the existing mode page. Note that *DEVICE* may still reject the new contents for the mode page. Cannot be given with the *--mask=M,M...* option.

-h, **--help**
output the usage message then exit.

-l, **--len**=10 | 6
length of the SCSI commands (cdb) sent to *DEVICE*. The default is 10 so 10 byte MODE SENSE and MODE SELECT commands are issued. Some old devices don't support the 10 byte variants hence this option.

-m, **--mask**=*M,M...*
where *M,M...* is a string of comma separated hex numbers each of which should resolve to a byte value (i.e. 0 to ff inclusive). A (single) space separated string of hex numbers is also allowed but the list needs to be in quotes. The mask chooses (bit by bit) whether the new mode page comes from the contents (mask bit set) or from the existing mode page (mask bit clear). If the mask string is shorter than the contents string then the remaining bytes are taken from the contents string. If the contents string is shorter than the existing mode page then the remaining bytes are taken from the existing mode page (i.e. they are left unaltered).

-p, **--page**=*PG_H*
where *PG_H* is the page code value to fetch and modify. The page code is in hex and should be between 0 and 3e inclusive. Notice that page code 3f to fetch all mode pages is disallowed.

-p, **--page**=*PG_H,SPG_H*
where *PG_H* is the page code value and *SPG_H* is the subpage code value to fetch and modify. Both values are in hex. The subpage code should be between 0 and fe inclusive. Notice that subpage code ff to fetch all mode subpages (for a given mode page or all mode pages in the case of 3f,ff) is disallowed.

-R, **--rtd**
when this option is given most other actions are bypassed and a MODE SELECT(6 or 10) command is sent to the *DEVICE* with the RTD bit set. This will cause all current values (and saved values if the *--save* option is also given) of all mode pages to be reverted to their default values.

-s, **--save**
changes the "saved" mode page when MODE SELECT is successful. By default (i.e. when *--save* is not used) only the "current" mode page values are changed when MODE SELECT is successful. In this case the new mode page will stay in effect until the device is reset (e.g. power cycled). When it restarts the "saved" values for the mode page will be re-instated. So to make changes permanent use the *--save* option.
When used with the *--rtd* option then both the current and saved values in each mode page are reverted to their default values. In the absence of *--save* option only the current values in each mode page are reverted to their default values.

-6, **--six**
this option will cause the 6 byte variants of MODE SENSE and MODE SELECT commands to be used. The default is to use the 10 byte options. This option is equivalent to using the *--len=6* option.

-v, **--verbose**
increase the level of verbosity, (i.e. debug output).

−V, **−−version**
print the version string and then exit.

NOTES

This utility does not check whether the contents string is trying to modify parts of the mode page which are changeable. The device should do that and if some part is not changeable then it should report: "Invalid field in parameter list".

Some mode pages are not saveable. If so an attempt to use the *−−save* option should cause an error to be reported from the device: "Illegal field in cdb".

The device is required to do various checks before it accepts a new mode page. If these checks fail then the mode page is not altered and either a "parameter list length error" or an "invalid field in parameter list" error is returned by the device in the sense data.

The recommended way to modify a mode page is to read it with a MODE SENSE, modify some part of it then write it back to the device with a MODE SELECT command. For example, reading an existing mode page can be accomplished with 'sg_modes −p=1a −r /dev/sdb > mp_1a.txt' (the power condition mode page). The mp_1a.txt file can be edited and then used as the contents string to this utility (e.g. 'sg_wr_mode −p 1a −s −c − /dev/sdb < mp_1a.txt').

Two fields differ between what is read from the device with MODE SENSE and what is written to the device with MODE SELECT: the mode data length is reserved (i.e. zero(es)) in a MODE SELECT command while the PS bit ((sub)page byte 0 bit 7) in each mode (sub)page is reserved (zero) in a MODE SELECT command. The PS bit given in the contents string is zeroed unless the *−−force* option is selected.

EXAMPLES

This utility can be used together with the sg_modes utility. To re−instate the default mode page values (i.e. the mode page values chosen by the manufacturer of the device) as both the current and saved mode page values the following sequence could be used:

```
$ sg_modes −−control=2 −−page=1a −r /dev/sda > t
$ sg_wr_mode −−page=1a −−contents=− −−save /dev/sda < t
```

Next is an example of using a mask to modify the "idle condition counter" of the "power condition" mode page (0x1a) from 0x28 to 0x37. Note that the change is not saved so the "idle condition counter" will revert to 0x28 after the next power cycle. The output from sg_modes is abridged.

```
$ sg_modes −−page=1a /dev/hdc
>> Power condition (mmc), page_control: current
00     1a 0a 00 03 00 00 00 28  00 00 01 2c

$ sg_wr_mode −p 1a −c 0,0,0,0,0,0,0,37 −m 0,0,0,0,0,0,0,ff /dev/hdc

$ sg_modes −p 1a /dev/hdc
>> Power condition (mmc), page_control: current
00     1a 0a 00 03 00 00 00 37  00 00 01 2c
```

EXIT STATUS

The exit status of sg_wr_mode 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

sdparm(sdparm), sg_modes(sg3_utils), sginfo(sg3_utils)