

CLS SES Firmware Tools Manual

1 CLS SES Firmware Tools Introduction

The CLS SES Firmware Tools contain utilities that can be used to perform SES firmware upgrade, CLI commands, diagnostic related logs dump.

The CLS SES Firmware tools contain the following utilities:

- In-band SES firmware upgrade tool (cls_upg_tool)
- CLI Over SES command line tool (cls_cli_tool)
- Firmware diagnostic logs dump tool (cls_diag_tool)
- Portable in-band SES firmware upgrade script (fw_upgrade.py)

2 CLS SES Firmware Tools Functionality

CLS SES Firmware tools provides the following functions:

- Perform in-band firmware upgrade via SES page
- Perform SES firmware command via in-band SES page interface
- Dump SES firmware diagnostic logs for trouble shooting

2.1 cls_upg_tool

cls_upg_tool is a Linux-based tool for upgrading CLS JBOD SES firmware. It was made specifically for Linux as it relies on the SCSI Generic Driver (sg), and all data is transferred through SES pages.

Operation Steps:

1. Connect the host server and one canister in the enclosure which you want to upgrade firmware;
2. Find out the SES device name (like “/dev/sgX”).
3. Execute “./cls_upg_tool -d /dev/sgX -f XXX.bin”, then you can upgrade the sg device with specify image.
4. For more operations, please use “./cls_upg_tool -h” for help;

Example:

```
[root@localhost ses_upg_tool]# ./cls_upg_tool -d /dev/sg3 -f eros_2.0.4.0_CombinedImage.bin
```

Download [other] image to JBOD Begin...

Upgrade current image progress ### [0%] ###

Upgrade current image progress ### [8%] ###

Upgrade current image progress ### [16%] ###

```
Upgrade current image progress ### [24%] ###
Upgrade current image progress ### [33%] ###
Upgrade current image progress ### [41%] ###
Upgrade current image progress ### [49%] ###
Upgrade current image progress ### [58%] ###
Upgrade current image progress ### [66%] ###
Upgrade current image progress ### [74%] ###
Upgrade current image progress ### [83%] ###
Upgrade current image progress ### [91%] ###
Upgrade current image progress ### [99%] ###
Upgrade current image progress ### [100%] ###
Download other image to JBOD Successfully!
```

2.2 cls_cli_tool

cls_cli_tool is a Linux-based tool for performing firmware CLI commands, all these CLI commands can be performed by console. It was made specifically for Linux as it relies on the SCSI Generic Driver (sg), and all data is transferred through SES pages.

Note: detail CLI commands, please refer to CLS SES firmware interface spec.

Operation Steps:

1. Connect the host server and one canister in the enclosure which you want to use cli commands;
2. Find out the SES device name (“/dev/sgX”).
3. Execute “./cls_cli_tool -d /dev/sgX”; then you can log in the cli command shell and do cli command just as you do the operation with UART terminal;
4. Execute “quit” when you want to exist the CLI command shell;
5. For more operations, please use “./cls_cli_tool -h” for help;

Example:

```
[root@localhost ses_upg_tool]# sg_scan -i
/dev/sg0: scsi0 channel=0 id=32 lun=0
      DP      BP12G+      1.00 [rmb=0 cmdq=0 pqual=0 pdev=0xd]
/dev/sg1: scsi0 channel=2 id=0 lun=0
      DELL PERC H310 2.12 [rmb=0 cmdq=1 pqual=0 pdev=0x0]
/dev/sg2: scsi5 channel=0 id=0 lun=0 [em]
      TSSTcorp DVD-ROM SN-108DN D150 [rmb=1 cmdq=0 pqual=0 pdev=0x5]
/dev/sg3: scsi7 channel=0 id=7 lun=0
      CLS 1ES0034-XX 0001 [rmb=0 cmdq=1 pqual=0 pdev=0xd]
[root@localhost ses_upg_tool]# ./cls_cli_tool -d /dev/sg3
[INFO] fetch next segment...
enter cli over ses shell

ESCE B $ qinfo
===== FW Revision 2.0.4.0 (@ IMG0/DATA0 on ECC) [Sep 21 2015] (500e0eca022412fe.8056.44/44) =====
#id name : v (crc .img_len.ver) [offset .begin -end .w.part_len ] (misc)
-----
#00 BOOT : v (efda3b51.00017c00.020006) [0x00000000.0xbf000000-0xbf017fff.w.0x0000c000]
```

```
#01 DAT0 : v (a3fa4cbf.0000fc40.020004) [0x00020000.0xbf020000-0xbf02ffff.r.0x00008000] (Using)
#02 DAT1 : v (a3fa4cbf.0000fc40.020004) [0x00320000.0xbf320000-0xbf32ffff.w.0x00008000]
#03 BCFG : v (f0050704.00000008.020006) [0x00300000.0xbf300000-0xbf317fff.w.0x0000c000] (AIF=IMG0/DATA0)
#05 IMG0 : v (5eb48469.001e0b60.020004) [0x00040000.0xbf040000-0xbf2cffff.r.0x00148000] (Running)
#06 IMG1 : v (5eb48469.001e0b60.020004) [0x00340000.0xbf340000-0xbf5cffff.w.0x00148000]
#08 CPLD : v (c4b26d91.0006952f.020004) [0x005e0000.0xbf5e0000-0xbf6dffff.w.0x00080000]
```

```
ESCE B $ quit
exit cli over ses shell
```

2.3 cls_diag_tool

cls_diag_tool is a Linux-based tool for dumping JBOD diagnostic logs for trouble shooting. It was made specifically for Linux as it relies on the SCSI Generic Driver (sg), and all data is transferred through SES pages.

Operation Steps:

1. Connect the host server and one canister in the enclosure which you want to gather diagnostic log;
2. Find out the SES device name (“/dev/sgX”).
3. Execute “./cls_diag_tool -n -d /dev/sgX”, the diagnostic log will save in current directory named as “diag_dump_[slot id]_[timestamp].dat”.
4. For more operations, please use “./cls_diag_tool -h” for help;

Example:

```
[root@localhost ses_upg_tool]# ./cls_diag_tool -n -d /dev/sg3
[INFO] fetch next segment...
CLS JBOD self diagnostic...
execute cli cmd <about> ok !
[INFO] fetch next segment...
[INFO] fetch next segment...
[INFO] fetch next segment...
execute cli cmd <checklist get> ok !
[INFO] fetch next segment...
execute cli cmd <temp get> ok !
[INFO] fetch next segment...
execute cli cmd <vpd get -d 0 0 256> ok !
[INFO] fetch next segment...
execute cli cmd <vpd get -d 0 256 256> ok !
[INFO] fetch next segment...
execute cli cmd <vpd get -d 0 512 256> ok !
[INFO] fetch next segment...
execute cli cmd <vpd get -d 1 0 256> ok !
[INFO] fetch next segment...
execute cli cmd <vpd get -d 1 256 256> ok !
```

```

[INFO] fetch next segment...
execute cli cmd <vpd get -d 1 512 256> ok !
execute cli cmd <mode get> ok !
[INFO] fetch next segment...
execute cli cmd <fru get> ok !
[INFO] fetch next segment...
[INFO] fetch next segment...
[INFO] fetch next segment...
[INFO] fetch next segment...
execute cli cmd <drv get> ok !
[INFO] fetch next segment...
[INFO] fetch next segment...
[INFO] fetch next segment...
[INFO] fetch next segment...
[INFO] fetch next segment...
execute cli cmd <port get> ok !
[INFO] fetch next segment...
execute cli cmd <power get> ok !
[INFO] fetch next segment...
[INFO] fetch next segment...
[INFO] fetch next segment...
[INFO] fetch next segment...
execute cli cmd <led get> ok !
[INFO] fetch next segment...
[INFO] fetch next segment...
execute cli cmd <threshold get> ok !
execute cli cmd <vpd get -e 1 0 256> ok !
execute cli cmd <vpd get -e 1 256 256> ok !
Exe Shell cmd: sg_inq /dev/sg3
Exe Shell cmd: sg_ses -p0 /dev/sg3
Exe Shell cmd: sg_ses -p1 /dev/sg3
Exe Shell cmd: sg_ses -p2 /dev/sg3
Exe Shell cmd: sg_ses -p5 /dev/sg3
Exe Shell cmd: sg_ses -p7 /dev/sg3
Exe Shell cmd: sg_ses -p10 /dev/sg3
execute cli cmd <log clear> ok !
execute cli cmd <log clear> ok !
Reset JBOD ...
Please wait 30 seconds ...
.....
.....
Reset JBOD successfully!
Dump log finished!

```

After the test, the diagnostic log will save in current directory.

```
[root@localhost ses_upg_tool]# ls
cls_cli_tool      diag_dump_B_2015-11-27-15h02h55s.dat  Eros_V2.0.4.0_pkg.zip
cls_diag_tool     eros_2.0.4.0_CombinedImage.bin          fw_upgrade_rev_0.1.py
cls_upg_tool      Eros_V2.0.4.0_FlipFlop_pkg.zip         ses_tools_64
```

Note: the dumped log file is human readable, users can open it by using file editor tools directly.

2.4 fw_upgrade_rev_0.1.py

fw_upgrade_rev_0.1.py is a Python script that can be used to download SES firmware image for CLS JBOD product. This script invokes sg_utils command sg_senddiag to perform the firmware upgrade via in-band SES pages. It is verified in Linux environment.

Operation Steps:

1. Connect the host server and one canister in the enclosure which you want to upgrade firmware;
2. Find out the SES device name(names like “/dev/sgX”).
3. Execute “./fw_upgrade.py /dev/sgX XXX.bin”, then you can upgrade the sg device with specify image.

Example:

```
[root@localhost ses_upg_tool]# ./fw_upgrade_rev_0.1.py /dev/sg4 eros_2.0.4.0_CombinedImage.bin
Send diagnostic cmd: 1d 10 00 08 09 00
Send diagnostic cmd: 1d 10 00 08 09 00
.....
Send diagnostic cmd: 1d 10 00 08 09 00
Send diagnostic cmd: 1d 10 00 08 09 00
Send diagnostic cmd: 1d 10 00 08 09 00
Send diagnostic cmd: 1d 10 00 08 09 00
Send diagnostic cmd: 1d 10 00 08 09 00
Send diagnostic cmd: 1d 10 00 08 09 00
Send diagnostic cmd: 1d 10 00 08 09 00
Send diagnostic cmd: 1d 10 00 08 09 00
Send diagnostic cmd: 1d 10 00 05 2c 00
toggle
Send diagnostic cmd: 1d 10 00 00 06 00
reset
Send diagnostic cmd: 1d 10 00 00 06 00
[root@localhost ses_upg_tool]#
```