Sun Datacenter InfiniBand Switch 648

Command Reference

Part No. 820-7741-12
May 2010, Revision A

Contents

Using This Documentation ix

Command Reference 1

Understanding Switch-Specific Commands 1

`activate` Command (switch-specific) 3

`checkfans` Command 4

`checklinks` Command 5

`checkpowers` Command 7

`checkpwrfault` Command 8

`checkswitches` Command 9

`checkvoltages` Command 10

`clearboardstat` Command 11

`deactivate` Command (switch-specific) 12

`disableboard` Command 13

`disablepsu` Command 14

`disablestby` Command 14

`disableswitchport` Command 15

`enableboard` Command 16

`enablehotinsert` Command 17

`enablepsu` Command 18

`enablestby` Command 19

`enableswitchport` Command 19

env_test Command 20
findport Command 21
getbaseguid Command 22
getboardstat Command 23
getfwversion Command 25
getportcounters Command 26
getportstatus Command 27
getpsufwver Command 28
getserialnumbers Command 29
getsymerr Command 30
i2ctest Command 31
m9diag Command 31
mcmversion Command 33
psustatus Command 33
resetswitch Command 34
selectdebugport Command 35
setlinkspeed Command 36
showlogs Command 37
showpresent Command 38
showtemps Command 39
showvoltages Command 40
Understanding CLIA Commands 42
activate Command (CLIA) 44
deactivate Command (CLIA) 45
exit Command (CLIA) 46
fru Command 47
frucontrol Command 48
fruinfo Command 49

`getfruledstate` Command 51
`getipmbstate` Command 52
`getlanconfig` Command 54
LAN Configuration Parameters 56
`help` Command (CLIA) 57
`ipmc` Command 58
`localaddress` Command 60
`poll` Command 60
`quit` Command 61
`sel` Command 62
`sensordata` Command 63
`session` Command 65
`setextracted` Command 66
`setfruledstate` Command 67
`setipmbstate` Command 68
`setlanconfig` Command 69
`setlocked` Command 70
`shelf` Command 71
`shelfaddress` Command 75
`shmstatus` Command 76
`showunhealthy` Command 77
`switchover` Command 78
`terminate` Command 78
`user [-v]` Command 79
`user add` Command 81
`user channel` Command 82
`user delete` Command 83
`user enable` Command 84

`user name` Command 85
`user passwd` Command 86
`version` Command (CLIA) 87
Understanding InfiniBand Software Commands 88
`ib_read_bw` Command 91
`ib_read_lat` Command 94
`ib_send_bw` Command 97
`ib_send_lat` Command 100
`ib_write_bw` Command 102
`ib_write_lat` Command 105
`ibaddr` Command 108
`ibcheckerrors` Command 110
`ibchecknet` Command 112
`ibchecknode` Command 113
`ibcheckport` Command 115
`ibcheckportstate` Command 116
`ibcheckportwidth` Command 118
`ibcheckstate` Command 119
`ibcheckwidth` Command 121
`ibclearcounters` Command 123
`ibclearerrors` Command 124
`ibdatacounters` Command 126
`ibdatacounts` Command 127
`ibdiagnet` Command 129
`ibdiagpath` Command 133
`ibhosts` Command 137
`ibnetdiscover` Command 139
`ibnodes` Command 142

`ibping` Command 143
`ibportstate` Command 146
`ibroute` Command 148
`ibrouters` Command 150
`ibstat` Command 152
`ibstatus` Command 154
`ibswitches` Command 155
`ibsysstat` Command 157
`ibtracert` Command 159
`ibv_devices` Command 161
`ibv_devinfo` Command 162
`ibv_rc_pingpong` Command 163
`ibv_srq_pingpong` Command 166
`ibv_uc_pingpong` Command 168
`ibv_ud_pingpong` Command 171
`ofed_info` Command 173
`opensm` Command 174
`opensmd` Daemon 180
`osmtest` Command 184
`perfquery` Command 187
`saquery` Command 190
`sminfo` Command 193
`smpdump` Command 194
`smpquery` Command 196
`vendstat` Command 198

Index 201

Using This Documentation

This command reference provides detailed information that describes the various commands used to administrate the Sun Datacenter InfiniBand Switch 648 from Oracle® in an InfiniBand fabric. This document is written for technicians, system administrators, authorized service providers, and users who have advanced experience with InfiniBand fabric hardware.

- "Related Documentation" on page ix
- "Documentation, Support, and Training" on page x
- "Documentation Feedback" on page x

Related Documentation

The documents listed as online are available at:

(http://docs.sun.com/app/docs/prod/ib.switch.648)

Application	Title	Part Number	Format	Location
Getting started	*Sun Datacenter InfiniBand Switch 648 Getting Started Guide*	820-7745	Printed PDF	Shipping kit Online
Last-minute information	*Sun Datacenter InfiniBand Switch 648 Product Notes*	820-7743	PDF	Online
Preparation and installation	*Sun Datacenter InfiniBand Switch 648 Installation Guide*	820-7738	PDF HTML	Online
Administration	*Sun Datacenter InfiniBand Switch 648 Administration Guide*	820-7739	PDF HTML	Online
Service	*Sun Datacenter InfiniBand Switch 648 Service Manual*	820-7740	PDF HTML	Online

Application	Title	Part Number	Format	Location
Command reference	*Sun Datacenter InfiniBand Switch 648 Command Reference*	820-7741	PDF HTML	Online
Compliance	*Sun Datacenter InfiniBand Switch 648 Safety and Compliance Guide*	820-7744	PDF	Online
ILOM information	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Supplement for the Sun Datacenter InfiniBand Switch 648*	821-0896	PDF HTML	Online

Documentation, Support, and Training

These web sites provide additional resources:

- Documentation (http://docs.sun.com)
- Support (http://www.sun.com/support)
- Training (http://www.sun.com/training)

Documentation Feedback

Submit comments about this document by clicking the Feedback[+] link at (http://docs.sun.com). Include the title and part number of your document with your feedback:

Sun Datacenter InfiniBand Switch 648 Command Reference, part number 820-7741-12.

Command Reference

This document provides detailed information regarding the commands that are used to administrate the Sun Datacenter InfiniBand Switch 648.

- "Understanding Switch-Specific Commands" on page 1
- "Understanding CLIA Commands" on page 42
- "Understanding InfiniBand Software Commands" on page 89

Related Information

- *Switch Installation*
- *Switch Administration*
- *Switch Remote Management*
- *Switch Service*

Understanding Switch-Specific Commands

The switch-specific commands are a simplified way to direct the Pigeon Point Shelf Manager, internal to the switch, to perform more complex tasks. Only the `root` user of the CMC can run the switch-specific commands. The format of the switch-specific commands is as follows:

`#` *command* [*component*] [*identifier*]

where:

- *command* is the command being issued.
- *component* is an abbreviated string representing a component.

- *identifier* is the numeric identifier of that component.

Command Syntax	Links
`activate lc|fc` *slot*	"`activate` Command (switch-specific)" on page 3
`checkfans`	"`checkfans` Command" on page 4
`checklinks`	"`checklinks` Command" on page 5
`checkpowers lc|fc` *slot*	"`checkpowers` Command" on page 7
`checkpwrfault`	"`checkpwrfault` Command" on page 8
`checkswitches`	"`checkswitches` Command" on page 9
`checkvoltages`	"`checkvoltages` Command" on page 10
`clearboardstat lc|fc` *slot*	"`clearboardstat` Command" on page 11
`deactivate lc|fc` *slot*	"`deactivate` Command (switch-specific)" on page 12
`disableboard lc|fc` *slot*	"`disableboard` Command" on page 13
`disablepsu` *slot*	"`disablepsu` Command" on page 14
`disablestby lc|fc` *slot*	"`disablestby` Command" on page 14
`disableswitchport lc|fc` *slot switch_chip port*	"`disableswitchport` Command" on page 15
`enableboard lc|fc` *slot*	"`enableboard` Command" on page 16
`enablehotinsert lc|fc` *slot*	"`enablehotinsert` Command" on page 17
`enablepsu` *slot*	"`enablepsu` Command" on page 18
`enablestby lc|fc` *slot*	"`enablestby` Command" on page 19
`enableswitchport lc|fc` *slot switch_chip port*	"`enableswitchport` Command" on page 19
`env_test`	"`env_test` Command" on page 20
`findport -g` *guid port*	"`findport` Command" on page 21
`getbaseguid lc|fc` *slot*	"`getbaseguid` Command" on page 22
`getboardstat lc|fc` *slot*	"`getboardstat` Command" on page 23
`getfwversion lc|fc` *slot*	"`getfwversion` Command" on page 25
`getportcounters lc|fc` *slot switch_chip port*	"`getportcounters` Command" on page 26
`getportstatus lc|fc` *slot switch_chip port*	"`getportstatus` Command" on page 27
`getpsufwver` *slot*	"`getpsufwver` Command" on page 28
`getserialnumbers`	"`getserialnumbers` Command" on page 29
`getsymerr lc|fc` *slot switch_chip port*	"`getsymerr` Command" on page 30

Command Syntax	Links
`i2ctest`	"`i2ctest` Command" on page 31
`m9diag`	"`m9diag` Command" on page 31
`mcmversion`	"`mcmversion` Command" on page 33
`psustatus` *slot*	"`psustatus` Command" on page 33
`resetswitch lc|fc` *slot switch_chip state*	"`resetswitch` Command" on page 34
`selectdebugport lc|fc` *slot*	"`selectdebugport` Command" on page 35
`setlinkspeed lc|fc` *slot switch_chip port speed*	"`setlinkspeed` Command" on page 36
`showlogs`	"`showlogs` Command" on page 37
`showpresent`	"`showpresent` Command" on page 38
`showtemps`	"`showtemps` Command" on page 39
`showvoltages`	"`showvoltages` Command" on page 40

Related Information

- "Understanding CLIA Commands" on page 42
- "Understanding InfiniBand Software Commands" on page 89

`activate` Command (switch-specific)

Activates a component. Issued on the CMC.

Syntax

`activate lc|fc` *slot*

where *slot* is the number of the line card (`0–8`) or fabric card (`0–8`).

Description

This switch-specific command brings a line card or fabric card from standby state to a fully powered-on state. Always activate all fabric cards before activating line cards.

Note – Activating a fabric card or line card can take up to 2 minutes for all of the switch chips to boot. During the switch chip boot process, the card might be identified as being in an active (M4) state.

Example

The following example shows how to activate line card 8 with the `activate` command.

```
# activate lc 8
#
```

Related Information

- "`deactivate` Command (switch-specific)" on page 12
- "`activate` Command (CLIA)" on page 44

`checkfans` Command

Displays fan status. Issued on the CMC.

Syntax

`checkfans`

Description

This switch-specific command displays fan status for fabric cards 0 through 8. Each fabric card has four fans, 0 through 3. The output contains fan speeds or a warning message of fan failure.

Example

The following example shows how to check fabric card fan status with the `checkfans` command.

```
# checkfans
Checking M9 fans...
FC 0 Fan 0 RPM = 21210.000000
FC 0 Fan 1 RPM = 21782.000000
Warning : FC 0 Fan 2 stopped
FC 0 Fan 3 RPM = 21496.000000

FC 1 Fan 0 RPM = 22068.000000
FC 1 Fan 1 RPM = 21782.000000
FC 1 Fan 2 RPM = 21496.000000
FC 1 Fan 3 RPM = 21782.000000
.
.
.
#
```

In the example, fan 2 of fabric card 0 has stopped.

Note – The output in the example is just a portion of the full output.

Related Information

- "`showpresent` Command" on page 38
- "`sensordata` Command" on page 64
- "`showunhealthy` Command" on page 78

`checklinks` Command

Checks which links are down. Issued on the CMC.

Syntax

```
checklinks [-c][-e]
```

Description

This switch-specific command verifies that links are up for active line cards and fabric cards. Output is either a simple return of `OK`, or a comprehensive list of ports and respective switch chips on the component that are down.

Options

The following table describes the options to the `checklinks` command and their purposes:

Option	Purpose
`-c`	Output includes ports to line card connectors, stating which connector (for example 3A) on the respective line card is down.
`-e`	Attempt to enable links that are down. When this option is used, re-run the `checklinks` command to verify that links have been enabled.

Example

The following example shows how to check which links are down with the `checklinks` command.

```
# checklinks -c
LC 0 Active, checking links.....OK
LC 1 Active, checking links.....OK
LC 2 Active, checking links.....OK
LC 3 Active, checking links.....
Port 16 on I4 01 LC 3 is down          (Cable 3 A )
Port 17 on I4 01 LC 3 is down          (Cable 3 A )
Port 18 on I4 01 LC 3 is down          (Cable 3 A )
Port 22 on I4 03 LC 3 is down          (Cable 6 A )
Port 23 on I4 03 LC 3 is down          (Cable 6 A )
Port 24 on I4 03 LC 3 is down          (Cable 6 A )
.
.
.
#
```

In the example, switch chips 1 and 3 on line card 3 have downed links. These links route to cables at connector 3A and 6A on line card 3.

Note – The output in the example is just a portion of the full output.

Related Information

- "`setlinkspeed` Command" on page 36

`checkpowers` Command

Initiates power pass-fail test. Issued on the CMC.

Syntax

`checkpowers lc|fc` *slot*

where *slot* is the number of the line card (`0–8`) or fabric card (`0–8`).

Description

This switch-specific command performs a pass-fail test for the internal power levels of a line card or fabric card.

Note – The component must be deactivated before it is checked.

Example

The following example shows how to check the internal power of line card 8 with the `checkpowers` command.

```
# checkpowers lc 8
Checking power OK signals on lc 8
Power check for LC 8 PASSED
#
```

Note – You must deactivate a line card of fabric card before using this command.

Related Information

- "checkpwrfault Command" on page 8
- "checkvoltages Command" on page 10
- "psustatus Command" on page 33
- "showvoltages Command" on page 40

checkpwrfault Command

Verifies internal power conditions all cards. Issued on the CMC.

Syntax

```
checkpwrfault
```

Description

This switch-specific command verifies internal power conditions are nominal for line cards and fabric cards. Output is a simplified `OK`.

Example

The following example shows how to verify internal power conditions for all fabric and line cards with the `checkpwrfault` command.

```
# checkpwrfault
Checking power fault in M9...
FC 0 Power fault sensor = 0x00      OK
FC 1 Power fault sensor = 0x00      OK
FC 2 Power fault sensor = 0x00      OK
.
.
.
#
```

Note – The output in the example is just a portion of the full output.

Related Information

- "checkpowers Command" on page 7
- "checkvoltages Command" on page 10
- "psustatus Command" on page 33
- "showvoltages Command" on page 40

checkswitches Command

Verifies switch chip boot status. Issued on the CMC.

Syntax

```
checkswitches
```

Description

This switch-specific command verifies switch chip boot status and the presence of fatal errors. Output is a simplified OK. Should an switch chip fail, the output indicates so.

Note – Wait at least 1 minute after activating a line card or fabric card before using the checkswitches command to verify the status of that component. If the checkswitches command is issued within 1 minute of activating a line card or fabric card, the command might report the respective switches as failed.

Example

The following example shows how to check switch chip boot status with the checkswitches command.

```
# checkswitches
Checking booted switches in M9...
FC 0 Active, checking switches ....OK
FC 1 Active, checking switches ....OK
FC 2 Active, checking switches ....OK
FC 3 Active, checking switches ....
I4 no 1 on FC 3 did not boot correct. Boot syndrome = 0x00000000
I4 no 2 on FC 3 did not boot correct. Boot syndrome = 0x00000000
```

```
FC 4 Active, checking switches ....OK
.
.
.
#
```

In the example, switch chips 1 and 2 on fabric card 3 did not boot correctly.

Note – The output in the example is just a portion of the full output.

Related Information

- "`resetswitch` Command" on page 34

`checkvoltages` Command

Verifies voltages for all cards. Issued on the CMC.

Syntax

`checkvoltages`

Description

This switch-specific command performs a pass-fail test to verify that voltages are at the nominal values for line cards and fabric cards. Output is a simplified `OK`.

Example

The following example shows how to check voltages are at nominal values for fabric cards and line cards with the `checkvoltages` command.

```
# checkvoltages
Reading M9 voltages...
Checking FC 0 ...
FC 0 OK
Checking FC 1 ...
FC 1 OK
Checking FC 2 ...
```

```
FC 2 OK
.
.
.
#
```

Note – The output in the example is just a portion of the full output.

Related Information

- "`checkpowers` Command" on page 7
- "`checkpwrfault` Command" on page 8
- "`env_test` Command" on page 20
- "`m9diag` Command" on page 31
- "`psustatus` Command" on page 33
- "`showvoltages` Command" on page 40

`clearboardstat` Command

Resets board statistics. Issued on the CMC.

Syntax

`clearboardstat lc|fc` *slot*

where *slot* is the number of the line card (`0–8`) or fabric card (`0–8`).

Description

This switch-specific command resets the statistics that are displayed with the `getboardstat` command.

Example

The following example shows how to reset the statistics for fabric card 0 with the `clearboardstat` command.

```
# clearboardstat fc 0
#
```

Related Information

- "`getboardstat` Command" on page 23
- "`ibclearerrors` Command" on page 125

`deactivate` Command (switch-specific)

Deactivates a card. Issued on the CMC.

Syntax

`deactivate lc|fc` *slot*

where *slot* is the number of the line card (`0–8`) or fabric card (`0–8`).

Description

This switch-specific command brings a line card or fabric card from a fully powered on state to a standby state. In the standby state, the IPMB bus is still active. You must deactivate a line card or fabric card if you want to run the `checkpowers` command for that component.

Example

The following example shows how to activate line card 8 with the `deactivate` command.

```
# deactivate lc 8
Deactivating LC 8
Pigeon Point Shelf Manager Command Line Interpreter
```

```
  Command issued via IPMB, status = 0 (0x0)
  Command executed successfully
#
```

Related Information

- "`activate` Command (switch-specific)" on page 3
- "`deactivate` Command (CLIA)" on page 45

`disableboard` Command

Disables a card. Issued on the CMC.

Syntax

`disableboard lc|fc` *slot*

where *slot* is the number of the line card (`0–8`) or fabric card (`0–8`).

Description

This switch-specific command brings a line card or fabric card from a standby state to a fully powered-off state. You can disable a component from a fully powered state. If you disable a component, you are asked if you want to do so. To avoid this message, first deactivate the component and then disable it.

Example

The following example shows how to disable line card 8, which is still active, with the `disableboard` command.

```
# disableboard lc 8
lc 8 is active do you want to continue removing this board (y/n)? y
lc 8 is M1 state, turning of stby
STBY for lc 8 is off
#
```

Related Information

- "`enableboard` Command" on page 16

- "disablestby Command" on page 14

disablepsu Command

Disables a power supply. Issued on the CMC.

Syntax

`disablepsu` *slot*

where *slot* is the number of the power supply (`0–3`).

Description

This switch-specific command brings a power supply from a fully powered on state to a standby state.

Example

The following example shows how to bring power supply 0 to a standby state with the `disablepsu` command.

```
# disablepsu 0
PSU 0, 12 V is off
#
```

Related Information

- "enablepsu Command" on page 18
- "psustatus Command" on page 33

disablestby Command

Disables standby power for a card. Issued on the CMC.

Syntax

`disablestby lc|fc` *slot*

where *slot* is the number of the line card (`0–8`) or fabric card (`0–8`).

Description

This switch-specific command disables the standby power on a line card or fabric card.

Example

The following example shows how to disable the standby power for line card 8 with the `disablestby` command.

```
# disablestby lc 8
. . . . . . . . . . STBY for lc 8 is off
#
```

Related Information

- "`enablestby` Command" on page 19

`disableswitchport` Command

Disables a switch chip port. Issued on the CMC.

Syntax

`disableswitchport lc|fc` *slot switch_chip port*

where:

- *slot* is number of the line card (`0–8`) or fabric card (`0–8`).
- *switch_chip* is the number of the chip on the line card (`0–3`) or fabric card (`0–1`).
- *port* is the number of the port (`1–36`).

Description

This switch-specific command disables a port on an switch chip for a specific line card or fabric card.

Example

The following example shows how to disable port 6 on switch chip 1 of fabric card 0 with the `disableswitchport` command.

```
# disableswitchport fc 0 1 6
Disabling port 6 on switch 1 on FC 0
#
```

Related Information

- "`enableswitchport` Command" on page 19
- "`ibportstate` Command" on page 147

`enableboard` Command

Enables a card. Issued on the CMC.

Syntax

`enableboard lc|fc` *slot*

where *slot* is the number of the line card (`0–8`) or fabric card (`0–8`).

Description

This switch-specific command brings a line card or fabric card from a fully powered off state to a standby state. The IPMB bus becomes active.

Example

The following example shows how to enable line card 8 with the `enableboard` command.

```
# enableboard lc 8
lc 8 is now enabled
#
```

Related Information

- "`disableboard` Command" on page 13
- "`activate` Command (switch-specific)" on page 3

`enablehotinsert` Command

Enables hot insertion of a card. Issued on the CMC.

Syntax

`enablehotinsert lc|fc` *slot*

where *slot* is the number of the line card (`0–8`) or fabric card (`0–8`).

Description

This switch-specific command readies the IPMB bus and standby power for a line card or fabric card slot before insertion, so that the bus and power are automatically enabled upon line card or fabric card insertion into that slot. This command is similar to the `enableboard` command or the `enableipmb` and `enablestby` commands, but is executed before the card is inserted.

Example

The following example shows how to ready line card slot 8 for hot-insertion with the `enablehotinsert` command.

```
# enablehotinsert lc 8
lc 8 is now enabled
#
```

Related Information

- "activate Command (switch-specific)" on page 3
- "enableboard Command" on page 16
- "enablestby Command" on page 19

enablepsu Command

Enables a power supply. Issued on the CMC.

Syntax

`enablepsu` *slot*

where *slot* is the number of the power supply (`0–3`).

Description

This switch-specific command brings a power supply from a standby state to a fully powered on state.

Example

The following example shows how to bring power supply 0 to a fully powered on state with the `enablepsu` command.

```
# enablepsu 0
Turning on 12V ...
PSU 0, 12 V on
#
```

Related Information

- "disablepsu Command" on page 14
- "psustatus Command" on page 33

`enablestby` Command

Enables standby power for a card. Issued on the CMC.

Syntax

`enablestby lc|fc` *slot*

where *slot* is the number of the line card (`0–8`) or fabric card (`0–8`).

Description

This switch-specific command enables the standby power on a line card or fabric card.

Example

The following example shows how to enable standby power for line card 8 with the `enablestby` command.

```
# enablestby lc 8
STBY for lc 8 OK
#
```

Related Information

- "`disablestby` Command" on page 14
- "`enableboard` Command" on page 16
- "`enablehotinsert` Command" on page 17

`enableswitchport` Command

Enables a switch chip port. Issued on the CMC.

Syntax

`enableswitchport lc|fc` *slot switch_chip port*

where:

- *slot* is number of the line card (0–8) or fabric card (0–8).
- *switch_chip* is the number of the chip on the line card (0–3) or fabric card (0–1).
- *port* is the number of the port (1–36).

Description

This switch-specific command enables a port on an I4 switch chip for a specific line card or fabric card.

Example

The following example shows how to enable port 6 on switch chip 1 of fabric card 1 with the `enableswitchport` command.

```
# enableswitchport fc 0 1 6
Enabling port 6 on switch 1 on FC 0
#
```

Related Information

- "`disableswitchport` Command" on page 15
- "`ibportstate` Command" on page 147

`env_test` Command

Performs CMC environmental test. Issued on the CMC.

Syntax

```
env_test
```

Description

This switch-specific command conducts a voltage and temperature test of the active CMC. The command output identifies nominal voltage, measured voltage, and temperature.

Example

The following example shows how to perform the CMC environment test with the `env_test` command.

```
# env_test
M9 CMC Environment test started:
Measured 3.3V STBY = 3.30 V
Measured 12V = 12.00 V
Measured 1.2V STBY = 1.21 V
Measured 1.2V = 1.22 V
Measured 2.5V = 2.51 V
Measured VBAT = 3.05 V
Temperature at ADM1026 = 27 ºC
M9 CMC Environment test PASSED
#
```

Related Information

- "`m9diag` Command" on page 31
- "`showvoltages` Command" on page 40
- "`showtemps` Command" on page 39
- "`sensordata` Command" on page 64

`findport` Command

Displays the location of a GUID. Issued on the CMC.

Syntax

`findport -g` *guid port*

where:

- *guid* is the global unit identifier.
- *port* is the number of the port (`1–36`).

Description

This switch-specific command displays the location of a switch chip within the switch. The output of the `findport` command also states if the port of the switch chip routes to a cable connection. The global unit identifier (GUID) for a node is provided in hexadecimal form, with the leading `0x` truncated. The `-g` option signifies a GUID is provided and is required.

The `findport` command also reports if the *port* is tied to an InfiniBand cable CXP connector, such as for a line card. The connector is identified by *group_number letter-P_number*, where:

- *group_number* is the connector group number (`0–11`).
- *letter* is the letter representing the upper row (`A`) or lower row (`B`).
- *P_number* is the number identifying the respective P and the location of the Link LED:
  - `1` – P1, the right Link LED
  - `2` – P2, the center Link LED
  - `3` – P3, the left Link LED

Example

The following example shows how to display the location of the node with GUID `0x0021283a83ae11d2`, port `19` within the switch with the `findport` command.

```
# findport -g 0021283a83ae11d2 19
Port 19 on switch 3 on LC 1 cable 9 B-3
#
```

In the example output, the node with GUID 0021283a83ae11d2 is I4 chip `3` in line card `1`. The link is to CXP cable connector `9B`. The `-3` means P3 carries the link, and link status is indicated with the left Link LED.

Related Information

- "`getbaseguid` Command" on page 22

`getbaseguid` Command

Displays base GUID. Issued on the CMC.

Syntax

`getbaseguid lc|fc` *slot*

where *slot* is the number of the line card (`0–8`) or fabric card (`0–8`).

Description

This switch-specific command displays the base global unit identifier (GUID) for a line card or fabric card. The GUID is a 64-bit value.

Example

The following example shows how to display the base GUID for fabric card 0 with the `getbaseguid` command.

```
# getbaseguid fc 0
Base GUID = 0x00bad0fc0033f100
#
```

Related Information

- "`findport` Command" on page 21
- "`ibnodes` Command" on page 143
- "`opensm` Command" on page 175

`getboardstat` Command

Displays board statistics. Issued on the CMC.

Syntax

`getboardstat lc|fc` *slot*

where *slot* is the number of the line card (`0–8`) or fabric card (`0–8`).

Description

This switch-specific command displays statistics that help you determine the reliability of a line card or fabric card. The output of this command contains the number of power faults, number of board boots, and numbers over temperature faults for the line card or fabric card.

Additionally, the boot syndrome and fatal error registers, the number of faults and boots, and the switch up and init flags for each switch chip are also displayed.

Example

The following example shows how to display the statistics for fabric card 0 with the `getboardstat` command.

```
# getboardstat fc 0
---------------------- FC 0 INFO ----------------------
current_power_fault:   0x000000
total_power_fault:     0x000000
Number of board boots: 0x0000
LM75_0 temp protect:   0x00
LM75_1 temp protect:   0x00
LM75_2 temp protect:   0x00
LM75_3 temp protect:   0x00
LM75_4 temp protect:   0x00
LM75_5 temp protect:   0x00
LM75_6 temp protect:   0x00
LM75_7 temp protect:   0x00
---------------------- I4 0 ----------------------
Current Boot Syndrome Register:  0x00000000
Current Fatal Error Register:    0x00000000
Previous Boot Syndrome Register: 0x00000000
Previous Fatal Error Register:   0x00000000
Number of faults:                0x0000
Number of boots:                 0x0003
Switch up flag:                  0x00
Switch init flag:                0x01
---------------------- I4 1 ----------------------
Current Boot Syndrome Register:  0x00000000
Current Fatal Error Register:    0x00000000
Previous Boot Syndrome Register: 0x00000000
Previous Fatal Error Register:   0x00000000
Number of faults:                0x0000
Number of boots:                 0x0003
Switch up flag:                  0x00
Switch init flag:                0x01
#
```

Related Information

- "saquery Command" on page 191

getfwversion Command

Displays firmware versions. Issued on the CMC.

Syntax

```
getfwversion lc|fc slot
```

where *slot* is the number of the line card (0–8) or fabric card (0–8).

Description

This switch-specific command displays firmware information for line cards and fabric cards, including switch chip firmware versions. Output identifies firmware type and version number.

Note – If any two switch chips have differing firmware versions, a message indicates as so.

Example

The following example shows how to display the firmware versions for fabric card 0 with the getfwversion command.

```
# getfwversion fc 0
FW versions for FC 0
  H8 version          : 0.1.4
  FC FPGA version     : 1.0.1
  I4 FW image version : 7.2.300
  INI file version    : 1
#
```

Related Information

- "getpsufwver Command" on page 28

- "mcmversion Command" on page 33
- "version Command (CLIA)" on page 88

getportcounters Command

Displays port counters. Issued on the CMC.

Syntax

`getportcounters lc|fc` *slot switch_chip port* `[-c]`

where:

- *slot* is number of the line card (0–8) or fabric card (0–8).
- *switch_chip* is the number of the chip on the line card (0–3) or fabric card (0–1).
- *port* is the number of the port (1–36).

Description

This switch-specific command returns the value of all port counters for a particular *switch* and *port* on that switch for a line card or fabric card. The `-c` option will clear all counters, instead of displaying them.

Example

The following example shows how to display the counters for switch chip 1, port 5 on line card 8 with the `getportcounters` command.

```
# getportcounters lc 8 1 5
Port counters on switch 1 port 5 :
Symbol error counter                  : 0x0000
Link error recovery counter           : 0x00
Link down counter                     : 0x00
PortRcvErrors counter                 : 0x0000
PortRcvSwitchRelayErrors counter      : 0x0000
PortXmitDiscards counter              : 0x0000
PortXmitConstraintErrors counter      : 0x00
PortRcvConstraintErrors counter       : 0x00
LocalLinkIntegrityErrors counter      : 0x00
ExcessiveBufferOverrunErrors counter : 0x00
VL15Dropped counter                   : 0x0000
```

```
PortXmitData counter                    : 0x001b0000
PortRcvData counter                     : 0x001b7ae8
PortXmitPkts counter                    : 0x00007ae8
PortRcvPkts counter                     : 0x000061b5
#
```

Related Information

- "getportstatus Command" on page 27
- "ibcheckerrors Command" on page 111
- "ibdatacounts Command" on page 128

getportstatus Command

Displays port status. Issued on the CMC.

Syntax

`getportstatus lc|fc` *slot switch_chip port* `[-v]`

where:

- *slot* is number of the line card (0–8) or fabric card (0–8).
- *switch_chip* is the number of the chip on the line card (0–3) or fabric card (0–1).
- *port* is the number of the port (1–36).

Description

This switch-specific command returns the status of the specified *port* of a particular *switch* chip on a line card or fabric card. The `-v` option provides verbose output.

Example

The following example shows how to display the status of switch chip 1, port 5 on line card 8 with the `getportstatus` command.

```
# getportstatus lc 8 1 5
Port Info for switch 1 port 5 :
PortState:              0x04 Active
```

```
PortPhysicalState:      0x05 LinkUp
LinkSpeedActive:        0x04 10 Gbps
#
```

Related Information

- "getportcounters Command" on page 26
- "ibcheckport Command" on page 116
- "ibcheckportstate Command" on page 117
- "ibcheckportwidth Command" on page 119

getpsufwver Command

Displays power supply firmware versions. Issued on the CMC.

Syntax

`getpsufwver` *slot*

where *slot* is the number of the power supply (`0–3`).

Description

This switch-specific command displays firmware version information for power supplies. If a power supply has been taken to a standby state, it is reported as `off`.

Example

The following example shows how to display the firmware version of power supply 0 with the `getpsufwver` command.

```
# getpsufwver 0
PSU 0 FW version 2.5
#
```

Related Information

- "getfwversion Command" on page 25
- "mcmversion Command" on page 33

■ "version Command (CLIA)" on page 88

getserialnumbers Command

Displays line card and fabric card serial numbers. Issued on the CMC.

Syntax

```
getserialnumbers
```

Description

This switch-specific command returns the serial numbers of all installed fabric cards and line cards.

Example

The following example shows how to display the line card and fabric card serial numbers with the getserialnumbers command.

```
# getserialnumbers
FC 00 SN=54138090901FC0007 Rev 1
FC 01 SN=54138110907FC0035 Rev 1
FC 02 SN=54138090901FC0005 Rev 1
FC 03 SN=54138090901FC0016 Rev 1
FC 04 SN=54138090901FC0006 Rev 1
FC 05 SN=54138110906FC0011 Rev 1
FC 06 SN=54138090901FC0014 Rev 1
FC 07 SN=54138090901FC0017 Rev 1
FC 08 SN=54138090901FC0015 Rev 1
LC 00 SN=54138100907LC0017 Rev 1
LC 01 SN=54138100907LC0018 Rev 1
LC 02 SN=54138100907LC0027 Rev 1
LC 03 SN=54138100907LC0025 Rev 1
LC 04 SN=54138100907LC0019 Rev 1
LC 05 SN=54138100907LC0024 Rev 1
LC 06 SN=54138100907LC0028 Rev 1
LC 07 SN=54138100907LC0029 Rev 1
LC 08 SN=54138100911LC0046 Rev 2
#
```

Related Information

- "fruinfo Command" on page 50

getsymerr Command

Displays port symbol error counter. Issued on the CMC.

Syntax

`getsymerr lc|fc` *slot switch_chip port* `-c`

where:

- *slot* is number of the line card (0–8) or fabric card (0–8).
- *switch_chip* is the number of the chip on the line card (0–3) or fabric card (0–1).
- *port* is the number of the port (1–36).

Description

This switch-specific command shows the symbol error counter for the specified *port* for a particular *switch* chip on a line card or fabric card. The `-c` option clears the counter.

Example

The following example shows how to display the symbol error counter for switch chip 1, port 5 on line card 8 with the `getsymerr` command.

```
# getsymerr lc 8 1 5
Symbol error counter switch 1 port 5 : 0x0000
#
```

Related Information

- "ibcheckerrors Command" on page 111
- "ibclearerrors Command" on page 125

`i2ctest` Command

Displays status of the I^2C bus. Issued on the CMC.

Syntax

```
i2ctest
```

Description

This switch-specific command conducts a pass-fail test of the I^2C devices on the CMC. Output identifies device, bus, address, and result of test.

Example

The following example shows how to perform the I^2C test with the `i2ctest` command.

```
# i2ctest
M9 CMC I2C access test started:
I2C test of ADM1026 at bus 3 addr 2e -PASSED
I2C test of PCA9506 at bus 2 addr 21 -PASSED
I2C test of PCA9506 at bus 2 addr 24 -PASSED
I2C test of PCA9548 at bus 2 addr 71 -PASSED
I2C test of PCA9555 at bus 2 addr 20 -PASSED
I2C test of AT64C24 at bus 2 addr 57 -PASSED
M9 CMC I2C access test PASSED
#
```

Related Information

- "`m9diag` Command" on page 31

`m9diag` Command

Performs a quick CMC check. Issued on the CMC.

Syntax

```
m9diag
```

Description

This switch-specific command returns the results of the last POST, performs the `i2test` command, and performs the `env_test` command. The output is a quick overview of the CMC's status. If any test is `FAILED` or has an error, of if the measured voltages deviate more than 10% from the normal values, there is a problem with the CMC or the power supplied to the CMC.

Example

The following example shows how to perform a quick CMC check with the `m9diag` command.

```
# m9diag

Apr  1 22:32:44 shmm1500 user.warn kernel: POST memory PASSED
Apr  1 22:32:44 shmm1500 user.warn kernel: POST i2c PASSED
Apr  1 22:32:44 shmm1500 user.warn kernel: POST uart PASSED
Apr  1 22:32:44 shmm1500 user.warn kernel: POST ethernet PASSED
Apr  1 22:32:44 shmm1500 user.warn kernel: POST crc PASSED

M9 CMC I2C access test started:
I2C test of ADM1026 at bus 3 addr 2e -PASSED
I2C test of PCA9506 at bus 2 addr 21 -PASSED
I2C test of PCA9506 at bus 2 addr 24 -PASSED
I2C test of PCA9548 at bus 2 addr 71 -PASSED
I2C test of PCA9555 at bus 2 addr 20 -PASSED
I2C test of AT64C24 at bus 2 addr 57 -PASSED
M9 CMC I2C access test PASSED

M9 CMC Environment test started:
Measured 3.3V STBY = 3.30 V
Measured 12V = 12.00 V
Measured 1.2V STBY = 1.21 V
Measured 1.2V = 1.22 V
Measured 2.5V = 2.51 V
Measured VBAT = 3.05 V
Temperature at ADM1026 = 27 ºC
M9 CMC Environment test PASSED
#
```

Related Information

- "env_test Command" on page 20
- "i2ctest Command" on page 31

mcmversion Command

Displays the MCM version. Issued on the CMC.

Syntax

```
mcmversion
```

Description

This switch-specific command displays the version of the switch chassis manager (MCM).

Example

The following example shows how to display the version of the MCM with the mcmversion command.

```
# mcmversion
M9CM version 1.1.4
Build time: Oct  6 2009 09:18:56
#
```

Related Information

- "getfwversion Command" on page 25
- "getpsufwver Command" on page 28
- "version Command (CLIA)" on page 88

psustatus Command

Displays power supply status. Issued on the CMC.

Syntax

`psustatus` *slot*

where *slot* is the number of the power supply (`0–3`).

Description

This switch-specific command displays the on-off status for a power supply. If a power supply has been taken to a standby state, it is reported as `off`.

Example

The following example shows how to display the status of power supply 0 with the `psustatus` command.

```
# psustatus 0
PSU 0, 12 V on
#
```

Related Information

- "`checkpowers` Command" on page 7
- "`checkpwrfault` Command" on page 8
- "`checkvoltages` Command" on page 10
- "`disablepsu` Command" on page 14
- "`enablepsu` Command" on page 18

`resetswitch` Command

Resets a switch chip. Issued on the CMC.

Syntax

`resetswitch lc|fc` *slot* *switch_chip* *state*

where:

- *slot* is number of the line card (`0–8`) or fabric card (`0–8`).

- *switch_chip* is the number of the chip on the line card (0–4) or fabric card (0–1). Using a value of 255 affects all switch chips on the card.
- *state* is 0 to reset once, and 1 to hold in reset.

Description

This switch-specific command resets a switch chip on a line card or fabric card. Reset can be one time, or reset can be held indefinitely until the switch is reset again.

Example

The following example shows how to reset switch chip 0 on line card 8 with the `resetswitch` command.

```
# resetswitch lc 8 0 0
#
```

Related Information

- "`ibportstate` Command" on page 147

`selectdebugport` Command

Sets component to debug. Issued on the CMC.

Syntax

`selectdebugport lc|fc` *slot*

where *slot* is number of the line card (0–8) or fabric card (0–8).

Description

This switch-specific command enables you to directly communicate with the H8 chip of the specified line card or fabric card, circumventing the indirect control of the CMC software. The `selectdebugport` command sets which line card or fabric card is addressed through the debug port of the CMC. The debug port is located between the NET MGT and SER MGT connectors, and uses the serial communication parameters of 9600, 8, N, 1, and no flow control.

Note – Resetting a CMC or performing a CMC switchover resets the debug port.

Example

The following example shows how to set line card 8 for debugging through the debug port on the CMC with the `setdebugport` command.

```
# selectdebugport lc 8
#
```

`setlinkspeed` Command

Sets a switch chip port speed. Issued on the CMC.

Syntax

`setlinkspeed lc|fc` *slot switch_chip port speed*

where:

- *slot* is number of the line card (`0–8`) or fabric card (`0–8`).
- *switch_chip* is the number of the chip on the line card (`0–3`) or fabric card (`0–1`).
- *port* is the number of the port (`1–36`).
- *speed* is the bandwidth of the port `qdr`, `ddr`, or `sdr`.

Description

This switch-specific command sets the speed of a port on a particular switch chip in a line card or fabric card. The speed set is singular and not a limit. If too many symbol errors are encountered, try setting for a slower speed (`ddr` or `sdr`).

Example

The following example shows how to set the speed of port 3 on switch 0 of line card 8 to quad data rate (`qdr`) with the `setlinkspeed` command.

```
# setlinkspeed lc 8 0 3 qdr
#
```

Related Information

- "`checklinks` Command" on page 5
- "`ibportstate` Command" on page 147

`showlogs` Command

Displays switch log. Issued on the CMC.

Syntax

```
showlogs
```

Description

This switch-specific command displays a more user-friendly log of switch events, including timestamp, location, sensor, and type of event.

Example

The following example shows how to display the switch log with the `showlogs` command.

```
# showlogs
Oct 26 11:55:34 2009; from:(LC 2); sensor:(I4_BOOTED_STATUS); event:asserted:
0x03
Oct 26 11:55:52 2009; from:(FC 0); sensor:(0); event:asserted: HotSwap: FRU 0
M4->M7, Cause=0x4
Oct 26 11:56:04 2009; from:(FC 0); sensor:(0); event:asserted: HotSwap: FRU 0
M7->M4, Cause=0x4
Oct 26 11:56:04 2009; from:(FC 0); sensor:(FAN_PRESENT); event:asserted: 0x00
Oct 26 11:56:04 2009; from:(FC 0); sensor:(FAN_PRESENT); event:asserted: 0x01
```

```
Oct 26 11:56:04 2009; from:(FC 0); sensor:(FAN_PRESENT); event:asserted: 0x02
.
.
.
#
```

Note – The output in the example is just a portion of the full output.

Related Information

- "sel Command" on page 62

showpresent Command

Displays components present. Issued on the CMC.

Syntax

```
showpresent
```

Description

This switch-specific command displays physically installed power supplies, line cards, and fabric cards. If a line card or fabric card is present, its activity is also displayed.

Example

The following example shows how to display components sensed in the chassis with the showpresent command.

```
# showpresent
PSU 00 present, state = ON
PSU 01 present, state = ON
PSU 02 present, state = ON
PSU 03 present, state = ON
FC 00 present and active (state = M4)
FC 01 present but not active (state = M1)
FC(F) 02 present but not active (state = unknown)
```

```
FC 03 present and active (state = M4)
FC 04 present and active (state = M4)
.
.
.
#
```

Note – The output in the examples are just a portion of the full output.

In this example, fabric card 1 has been deactivated. Because fabric card slot 2 has had standby voltage disabled, it is not possible for the CMC to determine if the component installed is a fabric card or a fabric card filler.

The string `(state = M4)` means the FRU is in an active state. Other states are identified by the following:

- `M0` – Not installed
- `M1` – Inactive
- `M2` – Activation request
- `M3` – Activation in progress
- `M4` – FRU active
- `M5` – Deactivation request
- `M6` – Deactivation in progress
- `M7` – Communication lost

Related Information

- "`checkfans` Command" on page 4
- "`m9diag` Command" on page 31
- "`psustatus` Command" on page 33
- "`ipmc` Command" on page 58
- "`showunhealthy` Command" on page 78

`showtemps` Command

Displays card temperatures. Issued on the CMC.

Syntax

```
showtemps
```

Description

This switch-specific command displays internal temperatures for all line cards and fabric cards.

Example

The following example shows how to display card temperatures with the `showtemps` command.

```
# showtemps
Retrieving M9 temperatures...
Temperature on FC 0, LM75 min = 22.00 C,  LM75 max = 23.00 C, at adm1026 = 24.00 C
Temperature on FC 1, LM75 min = 23.00 C,  LM75 max = 24.00 C, at adm1026 = 26.00 C
Temperature on FC 2, LM75 min = 21.00 C,  LM75 max = 25.00 C, at adm1026 = 24.00 C
Temperature on FC 3, LM75 min = 21.00 C,  LM75 max = 26.00 C, at adm1026 = 24.00 C
.
.
.
#
```

Note – The output in the example is just a portion of the full output.

Related Information

- "`env_test` Command" on page 20
- "`sensordata` Command" on page 64

`showvoltages` Command

Displays card voltages. Issued on the CMC.

Syntax

```
showvoltages
```

Description

This switch-specific command displays the internal voltages for the line cards and fabric cards. The left column displays the expected voltage, the right column displays the measured voltage.

Example

The following example shows how to display line card and fabric card voltages with the `showvoltages` command.

```
# showvoltages
Reading M9 voltages...
FC 0 readings
  1.8V      = 1.79
  2.5V_0    = 2.51
  2.5V_1    = 2.55
  1.2V_0    = 1.20
  1.2V_1    = 1.19
  3.3V STBY = 3.38
  3.3V      = 3.38
  2.5V STBY = 2.56
  5V        = 5.17
  12V       = 12.10
  1.8V STBY = 1.79
  1.2V STBY = 1.20

FC 1 readings
  1.8V      = 1.81
  2.5V_0    = 2.49
      .
      .
      .
#
```

Note – The output in the example is just a portion of the full output.

Related Information

- "checkpowers Command" on page 7
- "checkpwrfault Command" on page 8
- "checkvoltages Command" on page 10
- "env_test Command" on page 20
- "m9diag Command" on page 31
- "showvoltages Command" on page 40
- "sensordata Command" on page 64

Understanding CLIA Commands

The `clia` *command* commands are an intermediary interface to route directives to the Pigeon Point Shelf Manager, internal to the switch. The CMC is similar to the system controller used in Sun servers. Only the `root` user of the CMC can run the `clia` *command* commands. The format of the `clia` *command* is as follows:

`# clia` *command* `[`*option*`] [`*address*`|`*string*`]`

where:

- *command* is the command being issued.
- *option* is any option for that command.
- *address* is the hexadecimal address of the device being manipulated.
- *string* represents the address device pair.

Command Syntax	Links
`activate` *IPMB_address FRU_ID*	"activate Command (CLIA)" on page 44
`deactivate` *IPMB_address FRU_ID*	"deactivate Command (CLIA)" on page 45
`exit`	"exit Command (CLIA)" on page 46
`fru [-v] [`*addr* `[id=`*FRU_ID* `| type=`*site_type*`]] | [type=`*site_type* `[/`*site_number*`]]`	"fru Command" on page 47
`frucontrol` *IPMB_address FRU_ID option*	"frucontrol Command" on page 49
`fruinfo [-v] [-x]` *IPMB_address FRU_ID*	"fruinfo Command" on page 50
`getfruledstate [-v] [`*IPMB_address*`[`*FRU_ID* `[`*LED_ID* `| ALL]]]`	"getfruledstate Command" on page 51

Command Syntax	Links
`getipmbstate` *IPMB_address* `[`*link*`] (in radial IPMB-0 environment)`	"`getipmbstate` Command" on page 53
`getlanconfig` *channel* `[`*parameter_name* `[`*additional_parameters*`]]`	"`getlanconfig` Command" on page 54
`help [`*command* `[`*subcommand*`]]`	"`help` Command (CLIA)" on page 58
`ipmc [-v] [`*IPMB_address*`]`	"`ipmc` Command" on page 58
`localaddress`	"`localaddress` Command" on page 60
`poll`	"`poll` Command" on page 61
`quit`	"`quit` Command" on page 61
`sel [-v] [`*IPMB_address* `[`*record_count* `[`*starting_entry*`]]]`	"`sel` Command" on page 62
`sensordata [`*IPMB_address* `[`*sensor_name* `| [lun:]`*sensor_number*`]]`	"`sensordata` Command" on page 64
`session`	"`session` Command" on page 65
`setextracted` *IPMB_address FRU_ID*	"`setextracted` Command" on page 66
`setfruledstate` *IPMB_address FRU_ID LED_ID*`|ALL` *LEDOp* `[`*LEDColor*`]` *LEDOp* `= ON	OFF	LOCAL	BLINK` *onTime offTime* `| TEST` *onTime LEDColor* `= BLUE	RED	GREEN	AMBER	ORANGE	WHITE	NONE |` *number*	"`setfruledstate` Command" on page 67
`setipmbstate` *IPMB_address* `A|B [`*link*`] 1|0` (in radial IPMB-0 environment)	"`setipmbstate` Command" on page 69
`setlanconfig` *channel parameter_name additional_parameters*	"`setlanconfig` Command" on page 70
`setlocked` *IPMB_address FRU_ID* `0	1`	"`setlocked` Command" on page 71
`shelf` *subcommand*	"`shelf` Command" on page 72
`shelfaddress [`*up_to_20_characters_of_the_shelf_address*`]`	"`shelfaddress` Command" on page 76
`shmstatus`	"`shmstatus` Command" on page 77
`showunhealthy`	"`showunhealthy` Command" on page 78
`switchover`	"`switchover` Command" on page 79
`terminate`	"`terminate` Command" on page 79
`user [-v] [`*userid*`]`	"`user [-v]` Command" on page 80
`user add` *userid user_name channel_access_flags privilege_level password*	"`user add` Command" on page 81

Command Syntax	Links
`user channel` *userid channel_number channel_access_flags privilege_level*	"`user channel` Command" on page 83
`user delete` *userid*	"`user delete` Command" on page 84
`user enable` *userid* `1|0`	"`user enable` Command" on page 85
`user name` *userid user_name*	"`user name` Command" on page 86
`user passwd` *userid password*	"`user passwd` Command" on page 87
`version`	"`version` Command (CLIA)" on page 88

Related Information

- "Understanding Switch-Specific Commands" on page 1
- "Understanding InfiniBand Software Commands" on page 89

`activate` Command (CLIA)

Activates a FRU. Issued on the CMC.

Syntax

`activate` *IPMB_address FRU_ID*

where:

- *IPMB_address* is the IPMB bus address of the component.
- *FRU_ID* is the FRU identifier of the component or portion of the component.

Description

This CLIA command sends the IPMI command Set `FRU Activation (Activate FRU)` to the specified FRU. The FRU is specified using the IPMB address of the owning IPM controller and the FRU device ID.

Example

The following example shows how to activate the IPM controller at IPMB address 94 (fabric card 0) with the `activate` command.

```
# clia activate 94 0
Pigeon Point Shelf Manager Command Line Interpreter
Command issued via IPMB, status = 0 (0x0)
Command executed successfully
#
```

Related Information

- "`activate` Command (switch-specific)" on page 3
- "`deactivate` Command (CLIA)" on page 45
- "`setlocked` Command" on page 71

`deactivate` Command (CLIA)

Deactivates a FRU. Issued on the CMC.

Syntax

`deactivate` *IPMB_address FRU_ID*

where:

- *IPMB_address* is the IPMB bus address of the component.
- *FRU_ID* is the FRU identifier of the component or portion of the component.

Description

This CLIA command sends the IPMI command `Set FRU Activation (Deactivate FRU)` to the specified FRU.

Example

The following example shows how to deactivate the IPM controller at IPMB address 94 (fabric card 0) with the `deactivate` command.

```
# clia deactivate 94 0
Pigeon Point Shelf Manager Command Line Interpreter
Command issued via IPMB, status = 0 (0x0)
 Command executed successfully
#
```

Related Information

- "`deactivate` Command (switch-specific)" on page 12
- "`activate` Command (CLIA)" on page 44
- "`setlocked` Command" on page 71

`exit` Command (CLIA)

Exits the CLIA interactive mode. Issued on the CMC.

Syntax

```
exit
```

Description

The command exits the CLIA interactive mode. CLIA is entered by issuing `clia` without parameters.

Note – For the simplicity of using CLIA, switch-specific, and InfiniBand software stack commands concurrently, the CLIA interactive mode is not described in the switch documentation set.

Example

The following example shows how to exit the CLIA interactive mode with the `exit` command.

```
# exit
#
```

Related Information

- "`quit` Command" on page 61

`fru` Command

Displays information about a FRU. Issued on the CMC.

Syntax

`fru [-v] [`*IPMB_address* `[id=`*FRU_ID* `| type=`*site_type*`]] | [type=`*site_type* `[/`*site_number*`]]`

where:

- *IPMB_address* is the IPMB bus address of the component.
- *FRU_ID* is the FRU identifier of the component or portion of the component.
- *site_type* is the type of site.
- *site_number* is the number of the site.

Description

This CLIA command shows information about a specific FRU. The following information is shown for the FRU in standard mode:

- IPMB address and the FRU device ID.
- Entity ID, entity instance.
- Site type and number (if known).
- Current hot-swap state, previous hot-swap state, and cause of the last state change for the FRU. The hot-swap states `M0` – `M7` are defined in the PICMG 3.0 specification as follows:
  - `M0` – Not installed

- M1 – Inactive
- M2 – Activation request
- M3 – Activation in progress
- M4 – FRU active
- M5 – Deactivation request
- M6 – Deactivation in progress
- M7 – Communication lost

The following information is shown for the FRU in verbose mode only:

- The FRU device type, device type modifier (only for FRU-device-ID != 0). This information is taken from the FRU sensor data record (SDR) and conforms to section 37.12 of the IPMI specification.
- Device ID string from the FRU SDR.
- Current FRU power level and maximum FRU power level. Current assigned power consumption in Watts.

Options

The -v option enables verbose output.

Example

The following example shows how to get standard information about all FRUs at IPMB address 94 (fabric card 0) with the fru command.

```
# clia fru 94 0
Pigeon Point Shelf Manager Command Line Interpreter
94: FRU # 0
  Entity: (0xa0, 0x60)
  Hot Swap State: M4 (Active), Previous: M3 (Activation In Process), Last State
Change Cause: Normal State Change (0x0)
  Device ID String: "M9 FC"
#
```

Related Information

- "showpresent Command" on page 38
- "frucontrol Command" on page 49
- "fruinfo Command" on page 50

`frucontrol` Command

Controls a FRU. Issued on the CMC.

Syntax

`frucontrol` *IPMB_address FRU_ID option*

where:

- *IPMB_address* is the IPMB bus address of the component.
- *FRU_ID* is the FRU identifier of the component or portion of the component.
- *option* is described in the following topic.

Description

This CLIA command sends a request to the specified FRU to perform the specified operation on the FRU payload.

Options

The parameter *option* specifies the option of the `frucontrol` command to be used:

- `cold_reset` (abbreviated as `cr`) – Performs a cold reset of the FRU payload.
- `warm_reset` (abbreviated as `wr`) – Performs a warm reset of the FRU payload.
- `graceful_reboot` (abbreviated as `gr`) – Performs a graceful reboot of the FRU payload.
- `diagnostic_interrupt` (abbreviated as `di`) – Issues the diagnostic interrupt.

Example

The following example shows how to issue a cold reset command to FRU 0 at IPMB address 94 (fabric card 0) with the `frucontrol` command.

```
# clia frucontrol 94 0 cr
Pigeon Point Shelf Manager Command Line Interpreter
  FRU Control: Controller 0x94, FRU ID # 0, command 0x00, status 0(0x0)
  Command executed successfully
#
```

Related Information

- "`fru` Command" on page 47
- "`fruinfo` Command" on page 50

`fruinfo` Command

Displays user-friendly FRU information. Issued on the CMC.

Syntax

`fruinfo [-v] [-x]` *IPMB_address FRU_ID*

where:

- *IPMB_address* is the IPMB bus address of the component.
- *FRU_ID* is the FRU identifier of the component or portion of the component.

Description

This CLIA command shows FRU information in a user-friendly format.

Options

The following table describes the options to the `fruinfo` command and their purposes:

Option	Purpose
`-v`	Provides verbose output.
`-x`	Appends a hexadecimal dump of the FRU information to the standard output.

Example

The following example shows how to display user-friendly information about line card 8 at IPMB address 92 with the `fruinfo` command.

```
# clia fruinfo 92 0
Pigeon Point Shelf Manager Command Line Interpreter
92: FRU # 0, FRU Info
Common Header:    Format Version = 1
Board Info Area:
  Version     = 1
  Language Code             = 25
  Mfg Date/Time             = Mar 31 16:22:00 2009 (6967702 minutes since 1996)
  Board Manufacturer        = Sun Microsystems, Inc.
  Board Product Name        = Sun Datacenter Switch DCS 648 - Line Card
  Board Serial Number       = 54138100911LC0046
  Board Part Number         = 541-3810-02
  FRU Programmer File ID    = m9_lc_fru-info-54138100911LC0046.inf
Product Info Area:
  Version     = 1
  Language Code             = 25
  Manufacturer Name         = Sun Microsystems, Inc.
  Product Name              = Sun Datacenter Switch DCS 648 - Line Card
  Product Part / Model#     = 541-3810-02
  Product Version           = Rev 2
  Product Serial Number     = 54138100911LC0046
  Asset Tag                 = none
  FRU Programmer File ID    = m9_lc_fru-info-54138100911LC0046.inf
#
```

Related Information

- "`getserialnumbers` Command" on page 29
- "`fru` Command" on page 47
- "`frucontrol` Command" on page 49

`getfruledstate` Command

Displays FRU LED state. Issued on the CMC.

Syntax

`getfruledstate [-v] [`*IPMB_address* `[`*FRU_ID* `[`*LED_ID* `| ALL]]]`

where:

- *IPMB_address* is the IPMB bus address of the component.
- *FRU_ID* is the FRU identifier of the component or portion of the component.
- *LED_ID* is the numeric identifier of the LED.

Description

This CLIA command shows the current FRU LED state on all levels of control that are enabled for the LEDs.

Options

The `-v` option enables verbose output, where information about the colors supported by the LEDs is also shown.

Example

The following example shows how to display status LED states for line card 8 at IPMB address 92 with the `getfruledstate` command.

```
# clia getfruledstate -v 92 0
Pigeon Point Shelf Manager Command Line Interpreter
92: FRU # 0, Led # 0 ("BLUE LED"):
  Local Control LED State: LED OFF
  LED's color capabilities:
     Colors supported(0x02): BLUE
     Default LED Color in Local Control State(0x01): BLUE
     Default LED Color in Override State(0x01): BLUE
92: FRU # 0, Led # 1 ("LED 1"):
  Local Control LED State: LED OFF
  LED's color capabilities:
     Colors supported(0x10): AMBER
     Default LED Color in Local Control State(0x04): AMBER
     Default LED Color in Override State(0x04): AMBER
92: FRU # 0, Led # 2 ("LED 2"):
  Local Control LED State: LED ON, color: GREEN
  LED's color capabilities:
     Colors supported(0x08): GREEN
     Default LED Color in Local Control State(0x03): GREEN
     Default LED Color in Override State(0x03): GREEN
92: FRU # 0, Led # 3 ("LED 3"):
  Override LED State (current state): LED OFF
```

```
   Local Control LED State: LED OFF
   LED's color capabilities:
      Colors supported(0x40): WHITE
      Default LED Color in Local Control State(0x06): WHITE
      Default LED Color in Override State(0x06): WHITE
#
```

Related Information

- "`setfruledstate` Command" on page 67

`getipmbstate` Command

Displays state of IPMB. Issued on the CMC.

Syntax

`getipmbstate` *IPMB_address* [*link*] (in radial IPMB-0 environment)

`getipmbstate` *IPMB_address* (in bused IPMB-0 environment)

where:

- *IPMB_address* is the IPMB bus address of the component.
- *link* is the numeric value of the link (1–95).

Description

This CLIA command shows the current state of IPMB-0 on the target IPM controller.

In a bused environment, or in a radial environment if the target IPMC is not an IPMB hub, the argument *link* is not used.

In the radial environment, if the target IPM controller is an IPMB hub, the command works as follows:

- If *link* is omitted, the command prints information about the state of all radial IPMB links.
- If the *link* is present, the command prints information about the specific radial IPMB link (1 to 95).

In both cases, information about the state of both IPMB-A and IPMB-B is shown.

Example

The following example shows how to display the current state of the IPMB buses on the IPMC at IPMB address 92 (line card 8) with the `getipmbstate` command.

```
# clia getipmbstate 92
Pigeon Point Shelf Manager Command Line Interpreter
92: LUN: 0, Sensor # 1 ("IPMB Physical")
  Bus Status: 0x8  (IPMB-A Enabled, IPMB-B Enabled)
  IPMB A State: 0x08 (LocalControl, No failure)
  IPMB B State: 0x08 (LocalControl, No failure)
#
```

Related Information

- "`ipmc` Command" on page 58
- "`poll` Command" on page 61
- "`setipmbstate` Command" on page 69

`getlanconfig` Command

Displays LAN configuration parameters. Issued on the CMC.

Syntax

`getlanconfig` *channel* [*parameter* [*additional_parameters*]]

where:

- *channel* is the numeric identifier of the channel.
- *parameter* is either the parameter name or the respective number of the parameter.
- *additional_parameters* are additional parameters.

Description

This CLIA command shows the value of the specified LAN configuration parameter on the specified channel. Additional parameters are space-delimited. If no configuration parameter name or number is specified, all configuration parameters for the specified channel are shown. Refer to "LAN Configuration Parameters" on page 56 for information about the configuration parameters.

Example

The following example shows how to display the LAN parameter table for channel 1 with the `getlanconfig` command.

```
# clia getlanconfig 1
Pigeon Point Shelf Manager Command Line Interpreter
Authentication Type Support: 0x17 ( None MD2 MD5 Straight Password/Key )
Authentication Type Enables:
    Callback level: 0x00
    User level: 0x17 ( "None" "MD2" "MD5" "Straight Password/Key" )
    Operator level: 0x17 ( "None" "MD2" "MD5" "Straight Password/Key" )
    Administrator level: 0x17 ( "None" "MD2" "MD5" "Straight Password/Key" )
    OEM level: 0x00
IP Address: 10.60.34.20
IP Address Source: Static Address (Manually Configured) (0x01)
MAC Address: 00:18:49:00:86:32
Subnet Mask: 255.255.255.0
IPv4 Header Parameters: 0x40:0x40:0x10
Primary RMCP Port Number: 0x026f
Secondary RMCP Port Number: 0x0298
BMC-generated ARP Control: 0x02
    Enable BMC-generated ARP Response
Gratuitous ARP Interval: 2.0 seconds
Default Gateway Address: 10.60.34.254
Default Gateway MAC Address: 00:00:0c:07:ac:22
Backup Gateway Address: 0.0.0.0
Backup Gateway MAC Address: N/A
Community String: "public"
Number of Destinations: 16
Destination Type:
    N/A
Destination Address:
    N/A
802.1q VLAN ID: 0 (disabled)
VLAN priority: 0
Cipher Suite Entry count: 15
Supported Cipher Suite IDs: 0h, 1h, 2h, 3h, 4h, 5h, 6h, 7h, 8h, 9h, Ah, Bh, Ch,
Dh, Eh
Cipher Suite Privilege Levels:
   ID 00h, Priv.Level 'User'          (2); ID 01h, Priv.Level 'User'          (2);
   ID 02h, Priv.Level 'Administrator'  (4); ID 03h, Priv.Level 'OEM Proprietary'
(5);
   ID 04h, Priv.Level 'OEM Proprietary' (5); ID 05h, Priv.Level 'OEM Proprietary'
(5);
    ID 06h, Priv.Level 'User'          (2); ID 07h, Priv.Level 'Administrator'
(4);
```

```
   ID 08h, Priv.Level 'OEM Proprietary' (5); ID 09h, Priv.Level 'OEM Proprietary'
(5);
   ID 0Ah, Priv.Level 'OEM Proprietary' (5); ID 0Bh, Priv.Level 'Administrator'
(4);
   ID 0Ch, Priv.Level 'OEM Proprietary' (5); ID 0Dh, Priv.Level 'OEM Proprietary'
(5);
    ID 0Eh, Priv.Level 'OEM Proprietary' (5);
Destination Address VLAN TAGs:
    N/A
#
```

Related Information

- "`setlanconfig` Command" on page 70

LAN Configuration Parameters

The following table lists names and numbers of LAN configuration parameters supported by the `getlanconfig` and `setlanconfig` commands.

Parameter Name	Number	Description
`auth_support`	1	An 8-bit value that contains authentication type support flags for the LAN channel.
`auth_enables`	2	Five 8-bit values that contain authentication type enable flags for Callback, User, Operator, Administrator, and OEM privilege levels for the LAN channel.
`ip`	3	A string value that contains the IP address assigned to the LAN channel in dotted decimal notation (for example, 192.168.0.15).
`ip_source`	4	A value that encodes the source of the assigned IP address.
`mac`	5	A string value that contains the MAC address assigned to the LAN channel as six hexadecimal byte values delimited by colon (:) symbols (for example, 00:A0:24:C6:18:2F).
`subnet_mask`	6	A string value that contains the subnet mask assigned to the LAN channel in dotted decimal notation (for example, 255.255.255.0).
`ipv4_hdr_param`	7	Three 8-bit values that contain various IPv4 header parameters for sending RMCP packets: • Time-to-live • IP header flags (bits [7:5]) • Precedence (bits [7:5]) and type of service (bits [4:1])
`pri_rmcp_port`	8	A 16-bit value that contains the primary RMCP port number (the port used for regular RMCP communication).

Parameter Name	Number	Description
sec_rmcp_port	9	A 16-bit value that contains the secondary RMCP port number (the port used for secure RMCP communication).
arp_control	10	Two flags that control Address Resolution Protocol (ARP) behavior on the LAN channel: • Enable responding to ARP requests • Enable sending gratuitous ARPs
arp_interval	11	The gratuitous ARP interval in seconds, in fixed-point format (where the integral part represents seconds and the fractional part represents milliseconds).
dft_gw_ip	12	A string value that contains the IP address of the default gateway in dotted decimal notation.
dft_gw_mac	13	A string value that contains the MAC address of the default gateway as six hexadecimal byte values delimited by colons (:).
backup_gw_ip	14	A string value that contains the IP address of the backup gateway in dotted decimal notation.
backup_gw_mac	15	A string value that contains the MAC address of the backup gateway as six hexadecimal byte values delimited by colons (:).
community	16	A string value (up to 18 symbols) that is put into the Community String field in PET traps.
destination_count	17	The maximum number of LAN alert destinations supported on the LAN channel.
destination_type	18	The destination type identified by the specified set selector. If no set selector is given, all destination types are shown. Each destination type entry contains the following fields: • Destination type (0–7) • Alert acknowledge flag • Alert acknowledge timeout / retry interval in seconds (1–256) • Number of retries (0–7)
destination_address	19	The destination addresses associated with the specified set selector. If no set selector is given, all destination addresses are shown. Each destination address entry contains the following fields: • Gateway selector: 0 – use default, 1 – use backup • IP address (string in dotted decimal format) • MAC address (string of six hexadecimal byte values delimited by colons (:))

Related Information

- "getlanconfig Command" on page 54
- "setlanconfig Command" on page 70

`help` Command (CLIA)

Provides basic help. Issued on the CMC.

Syntax

`help [`*command* `[`*subcommand*`]]`

where:

- *command* is the command for which you are seeking help.
- *subcommand* is the option or subcommand for the command you are seeking help.

Description

This CLIA command shows help information for supported commands and their syntax.

Example

The following example shows how to get help on the `help` command with the `help` command.

```
# clia help help
Pigeon Point Shelf Manager Command Line Interpreter
 Provides basic help information
 help [<command>]
#
```

`ipmc` Command

Displays IPM controller information. Issued on the CMC.

Syntax

`ipmc [-v] [`*IPMB_address*`]`

where *IPMB_address* is the IPMB bus address of the component.

Description

This CLIA command shows information about the IPM controller at the specified address, or about all IPM controllers known to the Shelf Manager, if *IPMB_address* is omitted.

The following information is shown for the IPM controller in standard mode:

- IPMB address of the controller, as two hexadecimal digits.
- Entity ID and entity instance for the IPM controller.
- Maximum possible FRU device ID for the IPM controller.
- PICMG extension version. This version should be 2.0 for PICMG 3.0-compliant IPM controllers.
- Current hot-swap state, previous hot-swap state, and cause of the last state change for FRU device 0 of the IPM controller (which represents the IPM controller itself). The hot-swap states `M0` – `M7` are defined in the PICMG 3.0 specification as follows:
  - `M0` – Not installed
  - `M1` – Inactive
  - `M2` – Activation request
  - `M3` – Activation in progress
  - `M4` – FRU active
  - `M5` – Deactivation request
  - `M6` – Deactivation in progress
  - `M7` – Communication lost

Options

The `-v` option enables verbose output.

Example

The following example shows how to display information about the IPM controller at IPMB address 92 (line card 8) with the `impc` command.

```
# clia ipmc -v 92
Pigeon Point Shelf Manager Command Line Interpreter
92: Entity: (0xa0, 0x60) Maximum FRU device ID: 0x00
  PICMG Version 2.1
  Hot Swap State: M4 (Active), Previous: M3 (Activation In Process), Last State
Change Cause: Normal State Change (0x0)
  Device ID: 0x00, Revision: 0, Firmware: 0.14 (ver 0.1.4), IPMI ver 1.5
```

```
  Manufacturer ID: 00002a, Product ID: 097c, Auxiliary Rev: 00000000
  Device ID String: "M9 LC"
  Global Initialization: 0xc, Power State Notification: 0xc, Device
Capabilities: 0x29
  Controller provides Device SDRs
  Supported features: 0x29
      "Sensor Device" "FRU Inventory Device" "IPMB Event Generator"
#
```

Related Information

- "`showpresent` Command" on page 38
- "`getipmbstate` Command" on page 53
- "`poll` Command" on page 61
- "`showunhealthy` Command" on page 78

`localaddress` Command

Displays the CMC's IPMB address. Issued on the CMC.

Syntax

```
localaddress
```

Description

This CLIA command shows the IPMB address of the current CMC, based on its hardware address (as opposed to its generic BMC address 0x20).

Example

The following example shows how to display the IPMB address of the CMC with the `localaddress` command.

```
# clia localaddress
Pigeon Point Shelf Manager Command Line Interpreter
Local IPMB Address = 0x10
#
```

Related Information

- "shelf Command" on page 72

poll Command

Polls IPMB controllers. Issued on the CMC.

Syntax

```
poll
```

Description

This CLIA command initiates rediscovery of IPM controllers on IPMB-0 by sending the `Get Device ID` command to all IPMB addresses.

Example

The following example shows how to initiate rediscovery of the IPM controllers on IPMB-0 with the `poll` command.

```
# clia poll
Pigeon Point Shelf Manager Command Line Interpreter
IPMB polling thread started
#
```

Related Information

- "getipmbstate Command" on page 53
- "ipmc Command" on page 58

quit Command

Quits the CLIA interactive mode. Issued on the CMC.

Syntax

```
quit
```

Description

The command quits the CLIA interactive mode. CLIA is entered by issuing `clia` without parameters.

Note – For the simplicity of using CLIA, switch-specific, and InfiniBand software commands concurrently, the CLIA interactive mode is not described in the switch documentation set.

Example

The following example shows how to quit the CLIA interactive mode with the `quit` command.

```
# quit
#
```

Related Information

- "`exit` Command (CLIA)" on page 46

`sel` Command

Displays system event log. Issued on the CMC.

Syntax

`sel [-v] [`*IPMB_address* `[`*record_count* `[`*starting_entry*`]]]`

`sel clear [`*IPMB_address*`]`

`sel info [`*IPMB_address*`]`

where:

- *IPMB_address* is the IPMB bus address of the component.

- *record_count* is the number of subsequent records to be displayed.
- *starting_entry* is the entry number of the first record to be displayed.

Description

This CLIA command shows the contents of the system event log (SEL) on the specified IPM controller (at IPMB address 20h by default). The optional parameter *record_count* indicates how many records from the record number *starting_entry* in the SEL are shown. The optional parameter *starting_entry* is the entry number of the first SEL record to print, relative to the beginning of the SEL.

For each SEL record, the following information fields are shown:

- Record ID
- Record type (currently only events are supported, for which the word Event is shown)
- Timestamp (for timestamped records)
- Source address parameters – IPMB address, LUN, and channel number
- Type and number of the sensor that generated the event
- Event or reading type code
- Three bytes of event data, in raw and processed (if available) formats

The command `sel clear` clears the SEL on the specified IPM controller (at IPMB address 20h by default).

The command `sel info` provides information about the SEL on the specified IPM controller (at IPMB address 20h by default).

Options

The `-v` option makes the SEL entries output more user-friendly.

Example

The following example shows how to display information about the SEL for the CMC with the `sel` command.

```
# clia sel info
Pigeon Point Shelf Manager Command Line Interpreter
20: SEL version: 1.5
  Number of log entries: 912
  Free space: 1776 bytes
```

```
  Last addition timestamp: Nov  6 16:24:41 1983
  Last erase timestamp: Apr 22 00:40:19 2009
  Supported operations: 0x0f
#
```

Related Information

- "showlogs Command" on page 37

sensordata Command

Displays real-world sensor data. Issued on the CMC.

Syntax

`sensordata [`*IPMB_address* `[`*sensor_name* | `[lun:]`*sensor_number*`]]`

where:

- *IPMB_address* is the IPMB bus address of the component.
- *sensor_name* is the identifier string for the sensor.
- *sensor_number* is the number of the sensor.

Description

This CLIA command shows the actual value of the specified sensor (for a threshold-based sensor) or the currently asserted states (for a discrete sensor).

The following information is shown for each sensor:

- IPMB address of the owning IPM controller.
- Sensor number, sensor name (device ID string from the SDR), and the LUN by which the sensor can be accessed.
- The sensor type and event-reading type code.
- The sensor value (for threshold-based sensors) or the mask of currently asserted states (for discrete sensors) in raw form.
- The threshold crossing status, in hexadecimal format and with decoding.

Example

The following example shows how to display temperature values for `LM75_0` on line card 8 at address 92 with the `sensordata` command.

```
# clia sensordata 92 2
Pigeon Point Shelf Manager Command Line Interpreter
92: LUN: 0, Sensor # 2 ("LM75_0")
  Type: Threshold (0x01), "Temperature" (0x01)
  Belongs to entity (0x03, 0x60): FRU # 0
  Status: 0xc0
     All event messages enabled from this sensor
     Sensor scanning enabled
     Initial update completed
  Raw data: 30 (0x1e)
  Processed data: 30.000000 degrees C
  Status: 0x00
#
```

Related Information

- "`env_test` Command" on page 20
- "`i2ctest` Command" on page 31
- "`m9diag` Command" on page 31
- "`showtemps` Command" on page 39
- "`showvoltages` Command" on page 40

`session` Command

Displays RMCP sessions. Issued on the CMC.

Syntax

```
session
```

Description

This CLIA command shows information about active RMCP sessions. The information includes the following items:

- The maximum possible number of sessions and the number of currently active sessions.
- For each currently active session:
  - Session handle.
  - The user ID and name used during session activation.
  - Maximum session privilege level.
  - The IPMI channel number and type.
  - For LAN sessions, peer IP address and port number.

Example

The following example shows how to display information about active RMCP sessions with the `session` command.

```
# clia session
Pigeon Point Shelf Manager Command Line Interpreter
32 sessions possible, 2 sessions currently active
Session: 1
  User: ID 1, Name: ""; Privilege Level: "Administrator"
  Channel: 1 ("LAN_802_3"); Peer IP address: 172.16.2.203, Port: 1764
Session: 2
  User: ID 1, Name: ""; Privilege Level: "Administrator"
  Channel: 1 ("LAN_802_3"); Peer IP address: 172.16.2.203, Port: 1765
#
```

Related Information

- "`user [-v]` Command" on page 80
- "`user channel` Command" on page 83

`setextracted` Command

Instructs the CMC that a FRU has been removed. Issued on the CMC.

Syntax

`setextracted` *IPMB_address FRU_ID*

where:

- *IPMB_address* is the IPMB bus address of the component.
- *FRU_ID* is the FRU identifier of the component or portion of the component.

Description

This CLIA command notifies the CMC that the specified FRU has been physically extracted from the switch chassis.

Example

The following example shows how to notify the CMC that the component at address 92 has been extracted with the `setextracted` command.

```
# clia setextracted 92 0
Pigeon Point Shelf Manager Command Line Interpreter
Set FRU extracted state successfully
#
```

`setfruledstate` Command

Sets the state of a FRU LED. Issued on the CMC.

Syntax

`setfruledstate` *IPMB_address FRU_ID LED_ID*|`ALL` *LEDOp* `[`*LEDColor*`]`
LEDOp = `ON` | `OFF` | `LOCAL` | `BLINK` *onTime offTime* | `TEST` *onTime*
LEDColor = `BLUE` | `RED` | `GREEN` | `AMBER` | `ORANGE` | `WHITE` | `NONE` | *number*

where:

- *IPMB_address* is the IPMB bus address of the component.
- *FRU_ID* is the FRU identifier of the component or portion of the component.
- *LED_ID* is the numeric identifier of the LED.
- *LEDOp* is the operation of the LED.
- *LEDColor* is the color of the LED.
- *onTime* is the duration the LED is illuminated in milliseconds.
- *offTime* is the duration the LED is extinguished in milliseconds.
- *number* is the number respective to the color.

Description

This CLIA command enables you to set the state of a specific LED or all LEDs for the given FRU.

The first argument, *IPMB_address*, is the IPMB address of an IPM controller. The second argument, *FRU_ID*, is the FRU device ID. The third argument can be either an LED ID (a numerical value) or `ALL`. In the latter case, the specified operation applies to all LEDs.

The argument *LEDOp* specifies the operation applied to the FRUs, based on the PICMG 3.0 specification. The operations are defined as follows:

- `ON` – Turns on the LED.
- `OFF` – Turns off the LED.
- `LOCAL` – Reverts to the local control of the LED.
- `BLINK` – Causes the LED to blink, repeatedly turning it on for *onTime* milliseconds and then turning it off for *offTime* milliseconds.
- `TEST` – Runs a lamp test for *onTime* milliseconds.

For the `TEST` operation, *onTime* must be less then 12800 ms (12.8 sec). For the `BLINK` operation, both *onTime* and *offTime* values must be within a 10 – 2500 ms range.

The optional parameter *LEDColor* designates a color, either with a symbolic name or a decimal value. Symbolic names of colors correspond to decimal values in accordance with the PICMG 3.0 specification, as noted in the following list. (If the parameter is not specified, the default LED color is used.)

- `BLUE` = 1
- `RED` = 2
- `GREEN` = 3
- `AMBER` = 4
- `ORANGE` = 6
- `NONE` = 14 (don't change color)

Example

The following example shows how to enable blinking on LED 3 (white Locator LED) of line card 8 at IPMB address 92 with the `setfruledstate` command. The on duration is 100 ms. The off duration is 200 ms.

```
# clia setfruledstate 92 0 3 BLINK 100 200
```

```
Pigeon Point Shelf Manager Command Line Interpreter
  Setting FRU's led state completed successfully, status = 0x0
#
```

Related Information

- "`getfruledstate` Command" on page 51

`setipmbstate` Command

Enables or disables an IPMB link. Issued on the CMC.

Syntax

`setipmbstate` *IPMB_address* `A|B [`*link*`] 1|0` (in radial IPMB-0 environment)

`setipmbstate` *IPMB_address* `A|B 1|0` (in bused IPMB-0 environment)

where:

- *IPMB_address* is the IPMB bus address of the component.
- *link* is the numeric value of the link (1–95).

Description

This CLIA command enables or disables an IPMB link on the target IPM controller. The second argument defines the bus (IPMB-A or IPMB-B) to be enabled or disabled. The last argument defines the operation to be performed – `1` to enable link, `0` to disable link.

In a bused environment, and in a radial environment for target IPM controllers other than an IPMB hub, argument *link* is not used. For an IPMB hub controller in a radial environment, the argument *link* is optional.

If *link* is present, the command enables or disables the specific radial IPMB link (1 to 95). If *link* is omitted, the command enables or disables all the links on the IPMB hub in the radial system.

Example

The following example shows how to disable the IPMB-A link on line card 8 at IPMB address 92 with the `setimpbstate` command.

```
# clia setipmbstate 92 A 0
Pigeon Point Shelf Manager Command Line Interpreter
     Command executed successfully
#
```

Related Information

- "`getipmbstate` Command" on page 53

`setlanconfig` Command

Sets LAN configuration parameters. Issued on the CMC.

Syntax

`getlanconfig` *channel* `[`*parameter* `[`*additional_parameters*`]]`

where:

- *channel* is the numeric identifier of the channel.
- *parameter* is either the parameter name or the respective number of the parameter.
- *additional_parameters* are additional parameters.

Description

This CLIA command sets the value of the specified LAN configuration parameter on the specified channel. Refer to "LAN Configuration Parameters" on page 56 for information about the configuration parameters.

Example

The following example shows how to set the destination address of channel 1 with a set selector 2 and the default gateway with the `setlanconfig` command.

```
# clia setlanconfig 1 destination_address 2 0 172.16.2.100 90:93:93:93:93:93
Pigeon Point Shelf Manager Command Line Interpreter
Destination Addresses set successfully
#
```

Related Information

- "`getlanconfig` Command" on page 54

`setlocked` Command

Sets the locked bit. Issued on the CMC.

Syntax

`setlocked` *IPMB_address* *FRU_ID* `0 | 1`

where:

- *IPMB_address* is the IPMB bus address of the component.
- *FRU_ID* is the FRU identifier of the component or portion of the component.

Description

This CLIA command sets the locked bit for the specified FRU to the specified state (`0` for unlock or `1` for lock).

The locked bit controls whether the FRU is allowed to autonomously progress from state M1 (inactive) to state M2 (activation request). If the locked bit is set, this transition is not allowed.

This command can be used to reactivate a previously manually deactivated FRU by clearing the locked bit for it.

Example

The following example shows how to clear the locked bit for the IPM controller at IPMB address 92 (line card 8), thus allowing it to reactivate using the `setlocked` command.

```
# clia setlocked 92 0 0
Pigeon Point Shelf Manager Command Line Interpreter
Lock set successfully to 0x0
#
```

Related Information

- "`activate` Command (CLIA)" on page 44
- "`deactivate` Command (CLIA)" on page 45

`shelf` Command

Displays shelf FRU information. Issued on the CMC.

Syntax

`shelf [-v]` *command_parameter*

where *command_parameter* is:

- `address_table`
- `cooling_state`
- `fans_state`
- `power_distribution`
- `power_management`
- `pci_connectivity`
- `ha_connectivity`
- `h110_connectivity`
- `point_to_point_connectivity`

Description

The `shelf` command shows key shelf FRU information, plus selected current operating data for the shelf.

The following table lists the parameters supported by the `shelf` command:

Command Parameter	Information Provided by the Command Parameter
cooling_state (abbreviated as `cs`)	Shows the current cooling state of the shelf: • Normal – All temperature sensors show normal operating temperature. • Minor Alert – At least one temperature sensor is in minor alert state. None of the sensors is in major or critical alert state. • Major Alert – At least one temperature sensor is in major alert state. None of the sensors is in critical alert state. • Critical Alert – At least one temperature sensor is in critical alert state.
`fans_state` (abbreviated as `fs`)	Shows the current state of the fan tachometers in the shelf: • Normal – All fan tachometer sensors show normal operating speed. • Minor Alert – At least one fan tachometer sensor is in minor alert state. None of the sensors is in major or critical alert state. • Major Alert – At least one fan tachometer sensor is in major alert state. None of the sensors is in critical alert state. • Critical Alert – At least one fan tachometer sensor is in critical alert state.
`address_table` (abbreviated as `at`)	Shows the Address Table record in the Shelf FRU Info. The following information is provided: • Shelf Address (shown according to its type). • List of address table entries, showing Hardware Address, Site Number, and Site Type for each entry.
`power_distribution` (abbreviated as `pd`)	The following information is provided for each of the power feeds (mostly from the Shelf Power Distribution record of the Shelf FRU Information): • Maximum External Available Current. • Maximum Internal Current. • Minimum Expected Operating Voltage. • Actual Power Available. • Currently Used Power. • List of FRUs connected to the feed, showing Hardware Address and FRU Device ID for each FRU.

Command Parameter	Information Provided by the Command Parameter
`power_management` (abbreviated as `pm`)	The Shelf Power Management record in the Shelf FRU Info. This record contains a list of FRU Power Descriptors. For each descriptor the following information is provided: • Hardware Address. • FRU Device ID. • Maximum FRU Power Capability. • Shelf Manager Controlled Activation. • Delay Before Next Power On.
`pci_connectivity` (abbreviated as `pcic`)	The Shelf PCI Connectivity record in the Shelf FRU Info. The following information is provided: • PCI Slot Descriptor. • DSEL Connection. • Segment ID. • Extended PCI Slot Descriptor. • Geographic Address. • Interface Number. • System Slot Capable.
`ha_connectivity` (abbreviated as `ha`)	The Shelf HA Connectivity record in the Shelf FRU Info. The following information is provided: Radial Connectivity Support.
`h110_connectivity` (abbreviated as `h110c`)	The Shelf H110 Connectivity record in the Shelf FRU Info. The following information is provided: • Geographic Address. • Segment ID.
`point_to_point_connectivity` (abbreviated as `ppc`)	The Shelf Point-to-Point Connectivity record in the Shelf FRU Info. The following information is provided: • Channel Type. • Channel Count. • Slot/ HW Address. • Channel Descriptor.

Options

For the command parameters `cooling_state` and `fans_state`, the verbosity option `-v` is available. The option displays the list of sensors (temperature or fan tachometers) that contribute to the current state. Each sensor is shown as a tuple (*IPMB_address*, *sensor_number*).

Example

The following example shows how to display the IPMB addresses with the `shelf` command.

```
# clia shelf address_table
Pigeon Point Shelf Manager Command Line Interpreter
 PICMG Address Table Record (ID=0x10)
 Version = 1
 Shelf Address            = 0x10
 Address Table Entries#   = 27
  Hw Addr: 08 (10), Site # 1, Type: "Dedicated ShMC" 03
  Hw Addr: 09 (12), Site # 2, Type: "Dedicated ShMC" 03
  Hw Addr: 10 (20), Site # 1, Type: "OEM" c0
  Hw Addr: 10 (20), Site # 1, Type: "Shelf FRU Information" 02
  Hw Addr: 10 (20), Site # 2, Type: "Shelf FRU Information" 02
  Hw Addr: 10 (20), Site # 1, Type: "Power Supply" c5
  Hw Addr: 10 (20), Site # 2, Type: "Power Supply" c5
  Hw Addr: 10 (20), Site # 3, Type: "Power Supply" c5
  Hw Addr: 10 (20), Site # 4, Type: "Power Supply" c5
  Hw Addr: 41 (82), Site # 1, Type: "AdvancedTCA Board" 00
  Hw Addr: 42 (84), Site # 2, Type: "AdvancedTCA Board" 00
  Hw Addr: 43 (86), Site # 3, Type: "AdvancedTCA Board" 00
  Hw Addr: 44 (88), Site # 4, Type: "AdvancedTCA Board" 00
  Hw Addr: 45 (8a), Site # 5, Type: "AdvancedTCA Board" 00
  Hw Addr: 46 (8c), Site # 6, Type: "AdvancedTCA Board" 00
  Hw Addr: 47 (8e), Site # 7, Type: "AdvancedTCA Board" 00
  Hw Addr: 48 (90), Site # 8, Type: "AdvancedTCA Board" 00
  Hw Addr: 49 (92), Site # 9, Type: "AdvancedTCA Board" 00
  Hw Addr: 4a (94), Site # 10, Type: "AdvancedTCA Board" 00
  Hw Addr: 4b (96), Site # 11, Type: "AdvancedTCA Board" 00
  Hw Addr: 4c (98), Site # 12, Type: "AdvancedTCA Board" 00
  Hw Addr: 4d (9a), Site # 13, Type: "AdvancedTCA Board" 00
  Hw Addr: 4e (9c), Site # 14, Type: "AdvancedTCA Board" 00
  Hw Addr: 4f (9e), Site # 15, Type: "AdvancedTCA Board" 00
  Hw Addr: 50 (a0), Site # 16, Type: "AdvancedTCA Board" 00
  Hw Addr: 51 (a2), Site # 17, Type: "AdvancedTCA Board" 00
  Hw Addr: 52 (a4), Site # 18, Type: "AdvancedTCA Board" 00
#
```

Related Information

- "`localaddress` Command" on page 60

`shelfaddress` Command

Manages the Shelf Address field. Issued on the CMC.

Syntax

`shelfaddress` [*up_to_20_characters_of_the_shelf_address*]

`shelfaddress -x` *byte1 .... byteN*

Description

This CLIA command gets or sets the Shelf Address field of the Address Table within shelf FRU information.

Options

Without the option `-x`, the new shelf address is specified by a double-quoted string that can contain any ASCII characters and can be as long as 20 characters.

If the option `-x` is specified, the new shelf address is specified as a sequence of hexadecimal bytes separated with spaces. Up to 20 bytes can be specified, each byte is represented with two hexadecimal digits (the 0x prefix is optional).

Example

The following example shows a sequence of how to set the Shelf Address field with the `shelfaddress` command.

```
# clia shelfaddress
Pigeon Point Shelf Manager Command Line Interpreter
  Shelf Address Info:
# clia shelfaddress "New Shelf Address"
Pigeon Point Shelf Manager Command Line Interpreter
  Shelf Address Info set successfully
# clia shelfaddress
Pigeon Point Shelf Manager Command Line Interpreter
  Shelf Address Info: "New Shelf Address"
#
```

Related Information

- "user name Command" on page 86

shmstatus Command

Displays CMC status. Issued on the CMC.

Syntax

```
shmstatus [-v]
```

Description

This CLIA command returns the CMC status in redundant configurations: `Active` or `Backup`. The ready for operation flag is a parameter that shows as `Yes`. The `-v` option permits more detailed output.

Options

The `-v` option enables verbose output.

Example

The following example shows how to display the CMC status with the `shmstatus` command.

```
# clia shmstatus -v
Pigeon Point Shelf Manager Command Line Interpreter
Host: "Active"
Ready For Operation: Yes
Detailed State Flags: "Shelf FRU Found" "Backup Healthy" "Initial Update Sent"
"Initial Update Received" "RMCP Up"
#
```

Related Information

- "i2ctest Command" on page 31
- "showpresent Command" on page 38

- "showunhealthy Command" on page 78

showunhealthy Command

Displays problematic components. Issued on the CMC.

Syntax

```
showunhealthy
```

Description

This CLIA command shows the list of FRUs that appear to have a problem. The FRUs might be identified as:

- `Communication lost.`
- `Communication lost due to local failure.`
- `Unexpected deactivation.`

For each FRU, the following information is shown:

- IPMB address
- FRU ID
- Current hot-swap state
- Previous hot-swap state
- Cause of the last state change

Example

The following example shows how to display the list of unhealthy components in the system with the `showunhealthy` command.

```
# clia showunhealthy
Pigeon Point Shelf Manager Command Line Interpreter
There are no unhealthy components in the shelf.
#
```

Related Information

- "showpresent Command" on page 38

- "ipmc Command" on page 58
- "shmstatus Command" on page 77

switchover Command

Switches between CMCs. Issued on the CMC.

Syntax

```
switchover
```

Description

This CLIA command initiates switchover of the redundant CMC instances.

Example

The following example shows how to initiate the switchover from either the active or backup instance with the switchover command.

```
# clia switchover
 This Shelf Manager is now active, but is shutting down to trigger a switchover.
#
```

Related Information

- "activate Command (CLIA)" on page 44
- "terminate Command" on page 79

terminate Command

Terminates the CMC instance. Issued on the CMC.

Syntax

```
terminate
```

Description

This CLIA command terminates the CMC instance without rebooting the shelf management card. The instance can be either active or backup.

Note – By terminating the CMC instance, you sever the connection with the CMC. You must re-access the CMC to regain control.

Example

The following example shows how to terminate the CMC instance with the `terminate` command.

```
# clia terminate
#
```

Related Information

- "`deactivate` Command (CLIA)" on page 45
- "`switchover` Command" on page 79

`user [-v]` Command

Displays user information. Issued on the CMC.

Syntax

`user [-v] [`*userid*`]`

where *userid* is a valid user identifying number.

Description

This CLIA command shows information about users. When you type the command with a `-v` option, it also shows information about disabled users. If you specify the optional user ID, only information about the user with that ID is shown.

The following items of information are shown:

- User ID

- User name
- Channel access information for each IPMI channel – The maximum privilege level of that user on that channel and channel access flags.

  If the channel access information is the same for several channels, the output is coalesced and the range of channels is shown.

Example

The following example shows how to display user information with the `user -v` command.

```
# clia user -v
Pigeon Point Shelf Manager Command Line Interpreter
  1: ""
     Channels 0-15 Privilege level: "Administrator"
        Flags: "IPMI Messaging"
  2: "openhpi"
     Channels 0-15 Privilege level: "OEM Proprietary"
        Flags: "IPMI Messaging"
#
```

Related Information

- "`user add` Command" on page 81
- "`user channel` Command" on page 83
- "`user delete` Command" on page 84
- "`user enable` Command" on page 85
- "`user name` Command" on page 86
- "`user passwd` Command" on page 87

`user add` Command

Adds a user. Issued on the CMC.

Syntax

`user add` *userid user_name channel_access_flags privilege_level password*

where:

- *userid* is a valid user identifying number.
- *user_name* is the username (truncated to 16 characters).
- *channel_access_flags* is described in the following topic.
- *privilege_level* is the user privilege level.
- *password* is the user's password (truncated to 16 characters).

Description

This CLIA command adds a new user to the system. The channel access flag is the first byte of the SetUserInfo command, of which only bits 4, 5, and 6 are meaningful. Their use is as follows:

- Bit 6 – IPMI messaging enabled.
- Bit 5 – Link authentication enabled.
- Bit 4 – Restricted to callback.

Note – The user *password* is enclosed in quotes (" ").

Example

The following example shows how to add user 3 with the name `test_user`, administrator privilege level, and password `passwurd` with the `user add` command.

```
# clia user add 3 "test_user" 0x40 4 "passwurd"
Pigeon Point Shelf Manager Command Line Interpreter
  User 3 added successfuly
# clia user
Pigeon Point Shelf Manager Command Line Interpreter
  1: ""
     Channels 0-15 Privilege level: "Administrator"
        Flags: "IPMI Messaging"
 2: "openhpi"
     Channels 0-15 Privilege level: "OEM Proprietary"
        Flags: "IPMI Messaging"
 3: "test_user"
     Channels 0-15 Privilege level: "Administrator"
        Flags: "IPMI Messaging"
#
```

Related Information

- "`user delete` Command" on page 84
- "`user enable` Command" on page 85

`user channel` Command

Modifies user's channel access. Issued on the CMC.

Syntax

`user channel` *userid channel_number channel_access_flags privilege_level*

where:

- *userid* is a valid user identifying number.
- *channel_number* is the channel number.
- *channel_access_flags* is described in the following topic.
- *privilege_level* is the user privilege level.

Description

This CLIA command is used to modify the channel access setting for a specified channel and user by the user ID.) The channel access flag is the first byte of the `SetUserInfo` command, of which only bits 4, 5, and 6 are meaningful. Their use is as follows:

- Bit 6 – IPMI messaging enabled.
- Bit 5 – Link authentication enabled.
- Bit 4 – Restricted to callback.

Example

The following example shows how to change the maximum privilege level for user 3 on channel 5 to `User` with the `user channel` command.

```
# clia user 3
Pigeon Point Shelf Manager Command Line Interpreter
  3: "administrator"
     Channels 0-15 Privilege level: "Administrator"
        Flags: "IPMI Messaging"
```

```
# clia user channel 3 5 0x60 2
Pigeon Point Shelf Manager Command Line Interpreter
  User 3, channel 5 access updated successfuly
# clia user 3
Pigeon Point Shelf Manager Command Line Interpreter
  3: "administrator"
     Channels 0-4 Privilege level: "Administrator"
        Flags: "IPMI Messaging"
     Channel 5 Privilege level: "User"
        Flags: "Link Authentication" "IPMI Messaging"
     Channels 6-15 Privilege level: "Administrator"
        Flags: "IPMI Messaging"
#
```

Related Information

- "`user add` Command" on page 81
- "`user name` Command" on page 86
- "`user passwd` Command" on page 87

`user delete` Command

Deletes a user. Issued on the CMC.

Syntax

`user delete` *userid*

where *userid* is a valid user identifying number.

Description

This CLIA command deletes the user specified by the userid.

Example

The following example shows how to delete the user with identifying number 3 using the `user delete` command.

```
# clia user delete 3
Pigeon Point Shelf Manager Command Line Interpreter
  User 3 deleted successfuly
#
```

Related Information

- "`user add` Command" on page 81
- "`user enable` Command" on page 85

`user enable` Command

Disables or enables a user. Issued on the CMC.

Syntax

`user enable` *userid* `1|0`

where *userid* is a valid user identifying number.

Description

This CLIA command enables or disables a user by *userid*. The last command parameter specifies the requested action, as follows:

- 0 – Disables the specified user.
- nonzero – Enables the specified user.

Note – Like the `user delete` command, once disabled, a user is no longer listed with the `user` command. Therefore, you must remember the users you have disabled, if you want to enable them at a later time.

Example

The following example shows how to disable the user with identifying number 3 using the `user enable` command.

```
# clia user enable 3 0
Pigeon Point Shelf Manager Command Line Interpreter
  User 3 disabled successfuly
#
```

Related Information

- "`user add` Command" on page 81
- "`user delete` Command" on page 84

`user name` Command

Modifies a user's name. Issued on the CMC.

Syntax

`user name` *userid* *user_name*

where:

- *userid* is a valid user identifying number.
- *user_name* is the username (truncated to 16 characters).

Description

This CLIA command is used to modify the user name for the specified user.

Note – The string for the new user name is provided without quotes (" ").

Example

The following example shows how to change the name of user with identifying number 3 to `administrator` using the `user name` command.

```
# clia user 3
Pigeon Point Shelf Manager Command Line Interpreter
  3: "test_user"
     Channels 0-15 Privilege level: "Administrator"
        Flags: "IPMI Messaging"
# clia user name 3 administrator
Pigeon Point Shelf Manager Command Line Interpreter

# clia user 3
Pigeon Point Shelf Manager Command Line Interpreter
  3: "administrator"
     Channels 0-15 Privilege level: "Administrator"
        Flags: "IPMI Messaging"
#
```

Related Information

- "`user channel` Command" on page 83
- "`user passwd` Command" on page 87

`user passwd` Command

Modifies the user's password. Issued on the CMC.

Syntax

`user passwd` *userid password*

where:

- *userid* is a valid user identifying number.
- *password* is the user's password (truncated to 16 characters).

Description

This CLIA command is used to modify the password for the specified user.

Note – The user *password* is enclosed in quotes (" ").

Example

The following example shows how to change the password of user with identifying number 3 to `root` with the `user passwd` command.

```
# clia user passwd 3 "root"
Pigeon Point Shelf Manager Command Line Interpreter
  User 3, password changed successfuly
#
```

Related Information

- "`user channel` Command" on page 83
- "`user name` Command" on page 86

`version` Command (CLIA)

Displays CMC software version. Issued on the CMC.

Syntax

```
version
```

Description

This CLIA command shows the version information for the CMC software.

Example

The following example shows how to display the CMC software version with the `version` command.

```
# clia version
Pigeon Point Shelf Manager Command Line Interpreter
Pigeon Point Shelf Manager ver. 2.6.3
```

```
Pigeon Point and the stylized lighthouse logo are trademarks of Pigeon Point
Systems.
Copyright (c) 2002-2008 Pigeon Point Systems
All rights reserved
Build date/time: Oct  6 2009 09:26:43
Carrier: SUN_M9
Carrier subtype: 0; subversion: 1
#
```

Related Information

- "getfwversion Command" on page 25
- "getpsufwver Command" on page 28
- "mcmversion Command" on page 33

Understanding InfiniBand Software Commands

The InfiniBand software commands are a means of monitoring and controlling many aspects of the InfiniBand fabric. These commands are run from the Linux InfiniBand host, which is also the host of the Subnet Manager. Only the `root` user of the Linux InfiniBand host can run the InfiniBand software commands. The format of the InfiniBand software commands is typically as follows:

`#` *command* `[`*option*`]` `[`*option*`]` `...`

where:

- *command* is the command being issued.
- *option* is any option for that command.

Command Syntax	Links
`ib_read_bw [-p` *TCP_port*`][-d` *device*`][-i` *IB_port*`][-m` *mtu*`][-o` *num*`][-s` *size*`][-a][-t` *depth*`][-n` *iters*`][-b][-V][-e][-F][`*IP_address*`]`	"ib_read_bw Command" on page 92
`ib_read_lat [-p` *TCP_port*`][-d` *device*`][-i` *IB_port*`][-c RC|UC][-m` *mtu*`][-o` *num*`][-s` *size*`][-a][-t` *depth*`][-n` *iters*`][-C][-H][-U][-V][-e][-F][`*IP_address*`]`	"ib_read_lat Command" on page 95
`ib_send_bw [-p` *TCP_port*`][-d` *device*`][-i` *IB_port*`][-c RC|UC|UD][-m` *mtu*`] [-s` *size*`][-a][-t` *tdepth*`][g][-r` *rdepth*`][-n` *iters*`][-I` *size*`][-b][-V][-e] [-N][-F][`*IP_address*`]`	"ib_send_bw Command" on page 98

Command Syntax	Links
`ib_send_lat [-p` *TCP_port*`][-d` *device*`][-i` *IB_port*`][-c RC|UC|UD][-m` *mtu*`] [-s` *size*`][-a][-l][-t` *depth*`][-g][-n` *iters*`][-I` *size*`][-C][-H][-U][-V][-e] [-F][`*IP_address*`]`	"ib_send_lat Command" on page 101
`ib_write_bw [-p` *TCP_port*`][-d` *device*`][-i` *IB_port*`][-c RC|UC|UD][-m` *mtu*`] [-g` *posts*`][-q` *num*`][-s` *size*`][-a][-t` *depth*`][-n` *iters*`][-I` *size*`][-b][-V] [-N][-F][`*IP_address*`]`	"ib_write_bw Command" on page 103
`ib_write_lat [-p` *TCP_port*`][-d` *device*`][-i` *IB_port*`][-c RC|UC][-m` *mtu*`][-s` *size*`][-a][-t` *depth*`][-n` *iters*`][-I` *size*`][-C][-H][-U][-V][-F][`*IP_address*`]`	"ib_write_lat Command" on page 106
`ibaddr [-d][-D][-G][-l][-g][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`][-V][-h] [`*lid*`|`*dr_path*`|`*guid*`]`	"ibaddr Command" on page 109
`ibcheckerrors [-h][-b][-v][-N][`*topology*`|-C` *ca_name* `-P` *ca_port* `-t` *timeout*`]`	"ibcheckerrors Command" on page 111
`ibchecknet [-h][-N][`*topology*`|-C` *ca_name* `-P` *ca_port* `-t` *timeout*`]`	"ibchecknet Command" on page 113
`ibchecknode [-h][-v][-N][-G][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`] [`*lid*`|`*dr_path*`|`*guid*`]`	"ibchecknode Command" on page 114
`ibcheckport [-h][-v][-N][-G][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]` *lid*`|`*guid* *port*	"ibcheckport Command" on page 116
`ibcheckportstate [-G][-h][-N][-v][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`][`*lid*`|`*guid*`] `*port*	"ibcheckportstate Command" on page 117
`ibcheckportwidth [-G][-h][-N][-v][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`][`*lid*`|`*guid*`] `*port*	"ibcheckportwidth Command" on page 119
`ibcheckstate [-h][-N][-v][`*topology*`|-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]`	"ibcheckstate Command" on page 120
`ibcheckwidth [-h][-N][-v][`*topology*`|-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]`	"ibcheckwidth Command" on page 122
`ibclearcounters [-h][`*topology*`|-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]`	"ibclearcounters Command" on page 124
`ibclearerrors [-h][-N][`*topology*`|-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]`	"ibclearerrors Command" on page 125
`ibdatacounters [-b][-h][-N][-v][`*topology*`|-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]`	"ibdatacounters Command" on page 127
`ibdatacounts [-b][-G][-h][-N][-v][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]` *lid*`|`*guid* *port*	"ibdatacounts Command" on page 128
`ibdiagnet [-c` *count*`][-v][-r][-o` *outputdir*`][-t` *topology*`][-s` *system*`][-i` *device*`][-p` *port*`][-wt` *topology*`][-pm][-pc][-P` *PM* `=` *value*`][-lw 1x|4x|12x][-ls 2.5|5|10][-skip` *checks*`][-load_db` *file*`][-h][-V]`	"ibdiagnet Command" on page 130

Command Syntax	Links
`ibdiagpath -n[src_name,]dst_name|-l[src_lid,]dst_lid|-d p1,p2,p3,...[-c count][-v][-o outputdir][-t topology][-s system][-i device][-p port][-wt topology][-pm][-pc][-P PM = value][-lw 1x|4x|12x][-ls 2.5|5|10][-skip checks][-load_db file][-h][-V]`	"ibdiagpath Command" on page 134
`ibhosts [-h][topology|-C ca_name][-P ca_port][-t timeout]`	"ibhosts Command" on page 138
`ibnetdiscover [-d][-e][-v][-s][-l][-g][-H][-S][-R][-C ca_name][-P ca_port][-t timeout][-V][-p][-h][topology]`	"ibnetdiscover Command" on page 140
`ibnodes [-h][topology|-C ca_name][-P ca_port][-t timeout]`	"ibnodes Command" on page 143
`ibping [-d][-e][-G][-h][-s smlid][-v][-V][-C ca_name][-P ca_port][-t timeout][-c count][-f][-o oui][-S] lid|guid`	"ibping Command" on page 144
`ibportstate [-d][-D][-e][-G][-h][-s smlid][-v][-C ca_name][-P ca_port][-t timeout] lid|dr_path|guid port [op]`	"ibportstate Command" on page 147
`ibroute [-d][-a][-n][-D][-e][-G][-h][-M][-s smlid][-v][-V][-C ca_name][-P ca_port][-t timeout][lid|dr_path|guid [startlid [endlid]]]`	"ibroute Command" on page 149
`ibrouters [-h][topology|-C ca_name][-P ca_port][-t timeout]`	"ibrouters Command" on page 151
`ibstat [-d][-e][-h][-l][-s][-p][-v][-V] ca_name [ca_port]`	"ibstat Command" on page 153
`ibstatus [-h][devname[:IB_port]]...`	"ibstatus Command" on page 155
`ibswitches [-h][topology|-C ca_name][-P ca_port][-t timeout]`	"ibswitches Command" on page 156
`ibsysstat [-d][-e][-G][-h][-s smlid][-v][-V][-C ca_name][-P ca_port][-t timeout][-o oui][-S] lid|guid [op]`	"ibsysstat Command" on page 158
`ibtracert [-d][-D][-G][-h][-m mlid][-s smlid][-v][-V][-C ca_name][-P ca_port][-t timeout][lid|dr_path|guid [startlid [endlid]]]`	"ibtracert Command" on page 160
`ibv_devices`	"ibv_devices Command" on page 162
`ibv_devinfo [-d device][-i IB_port][-h][-l][-v]`	"ibv_devinfo Command" on page 163
`ibv_rc_pingpong [-p TCP_port][-d device][-i IB_port][-s size][-r depth] [-n iters][-l level][-e][-h]`	"ibv_rc_pingpong Command" on page 164
`ibv_srq_pingpong [-p TCP_port][-d device][-i IB_port][-s size][-q numqps][-r depth][-n iters][-l level][-e][-h]`	"ibv_srq_pingpong Command" on page 167
`ibv_uc_pingpong [-p TCP_port][-d device][-i IB_port][-s size][-r depth] [-n iters][-l level][-e][-h]`	"ibv_uc_pingpong Command" on page 169

Command Syntax	Links
`ibv_ud_pingpong [-p` *TCP_port*`][-d` *device*`][-i` *IB_port*`][-s` *size*`][-r` *depth*`][-n` *iters*`][-l` *level*`][-e][-h]`	"`ibv_ud_pingpong` Command" on page 172
`ofed_info`	"`ofed_info` Command" on page 174
`opensm [-F` *filename*`][-c` *filename*`][-g` *guid*`][-l` *lmc*`][-p` *priority*`][-smkey` *SMKey*`][-r][-R` *engine*`][-A][-z][-M` *filename*`][-U` *filename*`][-S` *filename*`][-a` *path*`][-u` *path*`][-X` *path*`][-m` *path*`][-o][-s` *interval*`][-t` *timeout*`][-maxsmps` *number*`][-console[off|local|socket|loopback]][-console-port` *port*`] [-i` *filename*`][-f` *path*`][-L` *size*`][-e][-P` *filename*`][-N][-Q[-Y` *filename*`]] [-y][-B][-I][-v][-V][-D` *flags*`][-d`*opt*`][-h]`	"`opensm` Command" on page 175
`/etc/init.d/opensmd start|stop|status`	"`opensmd` Daemon" on page 181
`osmtest [-f c|a|v|s|e|f|m|q|t][-w` *time*`][-d`*opt*`][-m` *lid*`][-g` *guid*`][-p][-i` *filename*`][-s`*opt*`][-M`*opt*`][-t` *timeout*`][-l` *path*`][-v][-vf` *flags*`][-h]`	"`osmtest` Command" on page 185
`perfquery [-d][-e][-G][-h][-a][-l][-r][-R][-v][-V][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`][`*lid*`|`*guid* `[[`*port*`][`*reset_mask*`]]]`	"`perfquery` Command" on page 188
`saquery [-h][-d][-p][-N][-D][-S][-I][-L][-l][-G][-O][-U][-c][-s][-g][-m][-x][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`][--src-to-dst` *source*`:`*destination*`][--sgid-to-dgid` *source-destination*`][`*name*`|`*lid*`|`*guid*`]`	"`saquery` Command" on page 191
`sminfo [-d][-e] -s` *state* `-p` *priority* `-a` *activity* `[-D][-G][-h][-v][-V][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]` *smlid*`|`*smdr_path*	"`sminfo` Command" on page 194
`smpdump [-s][-D][-h][-V][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]` *lid*`|`*dr_path* *attr* `[`*mod*`]`	"`smpdump` Command" on page 195
`smpquery [-d][-D][-e][-G][-h][-v][-V][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]` *lid*`|`*dr_path*`|`*guid* `[`*op* *params*`]`	"`smpquery` Command" on page 197
`vendstat [-d][-e][-G][-h][-N][-w][-v][-V][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]` *lid*`|`*guid*	"`vendstat` Command" on page 199

Related Information

- "Understanding Switch-Specific Commands" on page 1
- "Understanding CLIA Commands" on page 42

`ib_read_bw` Command

Performs a read bandwidth diagnostic. Issued on the Linux InfiniBand host.

Syntax

`ib_read_bw [-p` *TCP_port*`][-d` *device*`][-i` *IB_port*`][-m` *mtu*`][-o` *num*`][-s` *size*`][-a][-t` *depth*`][-n` *iters*`][-b][-V][-e][-F][`*IP_address*`]`

where:

- *TCP_port* is the TCP port.
- *device* is the InfiniBand device.
- *IB_port* is the InfiniBand port.
- *mtu* is the size of the MTU.
- *num* is the outstanding reads/atom.
- *size* is the size of the messages.
- *depth* is the size of the TX queue.
- *iters* is the number of message exchanges.
- *IP_address* is the IP address of the remote node host.

Description

This InfiniBand software command performs a bandwidth diagnostic between two nodes in the InfiniBand fabric. The command is dependent upon the Internet Protocol, so the InfiniBand fabric must be configured with Internet Protocol over InfiniBand (IPoIB). The command is a client-server, in that a remote node is configured as a server, while a local node performs as a client.

The command is first run locally on the server. The command is then run again locally on the client, pointing to the IP address of the server. The diagnostic checks the bandwidth of the data transfer from the server to the client. The connection uses the Reliable Datagram transport.

Note – This command only functions if your InfiniBand fabric is configured with IPoIB.

Options

The following table describes the options to the `ib_read_bw` command and their purposes:

Option	Purpose
`-p`	Uses the TCP port for initial synchronization.

Option	Purpose
-d	Uses the InfiniBand device.
-i	Uses the InfiniBand port.
-m	Sets the size of the MTU.
-o	Sets the number of outstanding reads/atom.
-s	Sets the size of the message to exchange.
-a	Runs all sizes, from 2 to 223.
-t	Sets the size of the TX queue.
-n	Performs iters message exchanges.
-b	Measures bidirectional bandwidth.
-V	Displays the version information.
-e	Sleeps on CQ events.
-F	Does not fail, even if the cpufreq_ondemand module is loaded.

Example

The following example shows how to run a diagnostic between a local node client and a remote node server with the ib_read_bw command. First configure the remote node server:

```
# ib_read_bw
------------------------------------------------------------------
                    RDMA_Read BW Test
Connection type : RC
  local address:  LID 0x01, QPN 0x990406, PSN 0xd5c3ba RKey 0x3682003a VAddr
0x002ad63cb81000
  remote address: LID 0x05, QPN 0x220406, PSN 0xa67d82, RKey 0x2af2043f VAddr
0x002b364ab80000
Mtu : 1024
#
```

Note – The output is not displayed until the local node client issues the respective command.

Then run the command on the local node client:

```
# ib_read_bw 192.168.200.100
---------------------------------------------------------------------
                    RDMA_Read BW Test
Connection type : RC
  local address:  LID 0x05, QPN 0x220406, PSN 0xa67d82 RKey 0x2af2043f VAddr
0x002b364ab80000
  remote address: LID 0x01, QPN 0x990406, PSN 0xd5c3ba, RKey 0x3682003a VAddr
0x002ad63cb81000
Mtu : 1024
---------------------------------------------------------------------
 #bytes #iterations    BW peak[MB/sec]    BW average[MB/sec]
 65536        1000             763.28               763.26
---------------------------------------------------------------------
#
```

Related Information

- "ib_read_lat Command" on page 95
- "ib_send_bw Command" on page 98
- "ib_write_bw Command" on page 103

ib_read_lat Command

Performs a read latency diagnostic. Issued on the Linux InfiniBand host.

Syntax

ib_read_lat [-p *TCP_port*][-d *device*][-i *IB_port*][-c RC|UC][-m *mtu*][-o *num*][-s *size*][-a][-t *depth*][-n *iters*][-C][-H][-U][-V][-e][-F][*IP_address*]

where:

- *TCP_port* is the TCP port.
- *device* is the InfiniBand device.
- *IB_port* is the InfiniBand port.
- *mtu* is the size of the MTU.
- *num* is the outstanding reads/atom.
- *size* is the size of the messages.
- *depth* is the size of the TX queue.

- *iters* is the number of message exchanges.
- *IP_address* is the IP address of the remote node host.

Description

This InfiniBand software command performs a latency diagnostic between two nodes in the InfiniBand fabric. The command is dependent upon the Internet Protocol, so the InfiniBand fabric must be configured with Internet Protocol over InfiniBand (IPoIB). The command is a client-server, in that a remote node is configured as a server, while a local node performs as a client.

The command is first run locally on the server. The command is then run again locally on the client, pointing to the IP address of the server. The diagnostic checks the latency of the data transfer from the server to the client. The connection uses the Reliable Datagram transport. Optionally, you can set the connection between the nodes to use Reliable Connection or Unreliable Connection transport.

Note – This command only functions if your InfiniBand fabric is configured with IPoIB.

Options

The following table describes the options to the `ib_read_lat` command and their purposes:

Option	Purpose
`-p`	Uses the TCP port for initial synchronization.
`-d`	Uses the InfiniBand device.
`-i`	Uses the InfiniBand port.
`-c`	Sets the connection type.
`-m`	Sets the size of the MTU.
`-o`	Sets the number of outstanding reads/atom.
`-s`	Sets the size of the message to exchange.
`-a`	Runs all sizes, from 2 to 223.
`-t`	Sets the size of the TX queue.
`-n`	Performs iters message exchanges.
`-C`	Reports time as CPU cycle units.

Option	Purpose
-H	Prints all results.
-U	Prints all results unsorted.
-V	Displays the version information.
-e	Sleeps on CQ events.
-F	Does not fail on different CPU frequencies.

Example

The following example shows how to run a diagnostic between a local node client and a remote node server with the `ib_read_lat` command. First configure the remote node server:

```
# ib_read_lat
------------------------------------------------------------------
                    RDMA_Read Latency Test
Connection type : RC
  local address: LID 0x01 QPN 0x9a0406 PSN 0x434b46 RKey 0x3688003a VAddr
0x00000001bfe080
  remote address: LID 0x05 QPN 0x230406 PSN 0x99013d RKey 0x2af8043f VAddr
0x000000118a3080
Mtu : 1024
#
```

Note – The output is not displayed until the local node client issues the respective command.

Then run the command on the local node client:

```
# ib_read_lat 192.168.200.100
------------------------------------------------------------------
                    RDMA_Read Latency Test
Connection type : RC
  local address: LID 0x05 QPN 0x230406 PSN 0x99013d RKey 0x2af8043f VAddr
0x000000118a3080
  remote address: LID 0x01 QPN 0x9a0406 PSN 0x434b46 RKey 0x3688003a VAddr
0x00000001bfe080
Mtu : 1024
------------------------------------------------------------------
 #bytes #iterations    t_min[usec]    t_max[usec]  t_typical[usec]
```

```
      2        1000           9.44           96.39              9.73
---------------------------------------------------------------------
#
```

Related Information

- "ib_read_bw Command" on page 92
- "ib_send_lat Command" on page 101
- "ib_write_lat Command" on page 106

ib_send_bw Command

Performs a read bandwidth diagnostic. Issued on the Linux InfiniBand host.

Syntax

ib_send_bw [-p *TCP_port*][-d *device*][-i *IB_port*][-c RC|UC|UD][-m *mtu*] [-s *size*][-a][-t *tdepth*][g][-r *rdepth*][-n *iters*][-I *size*][-b][-V][-e] [-N][-F][*IP_address*]

where:

- *TCP_port* is the TCP port.
- *device* is the InfiniBand device.
- *IB_port* is the InfiniBand port.
- *mtu* is the size of the MTU.
- *size* is the size of the messages.
- *tdepth* is the size of the TX queue.
- *rdepth* is the size of the RX queue.
- *iters* is the number of message exchanges.
- *IP_address* is the IP address of the remote node host.

Description

This InfiniBand software command performs a bandwidth diagnostic between two nodes in the InfiniBand fabric. The command is dependent upon the Internet Protocol, so the InfiniBand fabric must be configured with Internet Protocol over InfiniBand (IPoIB). The command is a client-server, in that a remote node is configured as a server, while a local node performs as a client.

The command is first run locally on the server. The command is then run again locally on the client, pointing to the IP address of the server. The diagnostic checks the bandwidth of the data transfer from the server to the client. The connection uses the Reliable Datagram transport. Optionally, you can set the connection between the nodes to use Reliable Connection, Unreliable Connection, or Unreliable Datagram transport.

Note – This command only functions if your InfiniBand fabric is configured with IPoIB.

Options

The following table describes the options to the `ib_send_bw` command and their purposes:

Option	Purpose
`-p`	Uses the TCP port for initial synchronization.
`-d`	Uses the InfiniBand device.
`-i`	Uses the InfiniBand port.
`-c`	Sets the connection type.
`-m`	Sets the size of the MTU.
`-s`	Sets the size of the message to exchange.
`-a`	Runs all sizes, from 2 to 223.
`-t`	Sets the size of the TX queue.
`-g`	Sends messages to the multicast group for the UD connection.
`-r`	Sets the RX queue larger than the TX queue.
`-n`	Performs iters message exchanges.
`-I`	Sets the maximum message size to be sent in inline mode.
`-b`	Measures bidirectional bandwidth.
`-V`	Displays the version information.
`-e`	Sleeps on CQ events.
`-N`	Cancels the peak bandwidth calculation.
`-F`	Does not fail, even if the `cpufreq_ondemand` module is loaded.

Example

The following example shows how to run a diagnostic between a local node client and a remote node server with the `ib_send_bw` command. First configure the remote node server:

```
# ib_send_bw
------------------------------------------------------------------
                    Send BW Test
Connection type : RC
Inline data is used up to 400 bytes message
  local address:  LID 0x01, QPN 0x9b0406, PSN 0x49c0fe
  remote address: LID 0x05, QPN 0x240406, PSN 0xaf88e
Mtu : 1024
#
```

Note – The output is not displayed until the local node client issues the respective command.

Then run the command on the local node client:

```
# ib_send_bw 192.168.200.100
------------------------------------------------------------------
                    Send BW Test
Connection type : RC
Inline data is used up to 400 bytes message
  local address:  LID 0x05, QPN 0x240406, PSN 0xaf88e
  remote address: LID 0x01, QPN 0x9b0406, PSN 0x49c0fe
Mtu : 1024
------------------------------------------------------------------
 #bytes #iterations    BW peak[MB/sec]    BW average[MB/sec]
  65536        1000             785.57                785.57
------------------------------------------------------------------
#
```

Related Information

- "`ib_read_bw` Command" on page 92
- "`ib_send_lat` Command" on page 101
- "`ib_write_bw` Command" on page 103

`ib_send_lat` Command

Performs a read latency diagnostic. Issued on the Linux InfiniBand host.

Syntax

`ib_send_lat [-p` *TCP_port*`][-d` *device*`][-i` *IB_port*`][-c RC|UC|UD][-m` *mtu*`] [-s` *size*`][-a][-l][-t` *depth*`][-g][-n` *iters*`][-I` *size*`][-C][-H][-U][-V][-e] [-F][`*IP_address*`]`

where:

- *TCP_port* is the TCP port.
- *device* is the InfiniBand device.
- *IB_port* is the InfiniBand port.
- *mtu* is the size of the MTU.
- *size* is the size of the messages.
- *depth* is the size of the TX queue.
- *iters* is the number of message exchanges.
- *IP_address* is the IP address of the remote node host.

Description

This InfiniBand software command performs a latency diagnostic between two nodes in the InfiniBand fabric. The command is dependent upon the Internet Protocol, so the InfiniBand fabric must be configured with Internet Protocol over InfiniBand (IPoIB). The command is a client-server, in that a remote node is configured as a server, while a local node performs as a client.

The command is first run locally on the server. The command is then run again locally on the client, pointing to the IP address of the server. The diagnostic checks the latency of the data transfer from the server to the client. The connection uses the Reliable Datagram transport. Optionally, you can set the connection between the nodes to use Reliable Connection, Unreliable Connection, or Unreliable Datagram transport.

Note – This command only functions if your InfiniBand fabric is configured with IPoIB.

Options

The following table describes the options to the `ib_send_lat` command and their purposes:

Option	Purpose
`-p`	Uses the TCP port for initial synchronization.
`-d`	Uses the InfiniBand device.
`-i`	Uses the InfiniBand port.
`-c`	Sets the connection type.
`-m`	Sets the size of the MTU.
`-s`	Sets the size of the message to exchange.
`-a`	Runs all sizes, from 2 to 223.
`-l`	Specifies signal completion on each message.
`-t`	Sets the size of the TX queue.
`-g`	Sends messages to the multicast group for the UD connection.
`-n`	Performs iters message exchanges.
`-I`	Sets the maximum message size to be sent in inline mode.
`-C`	Reports time as CPU cycle units.
`-H`	Prints all results.
`-U`	Prints all results unsorted.
`-V`	Displays the version information.
`-e`	Sleeps on CQ events.
`-F`	Does not fail, even if the cpufreq_ondemand module is loaded.

Example

The following example shows how to run a diagnostic between a local node client and a remote node server with the `ib_send_lat` command. First configure the remote node server:

```
# ib_send_lat
------------------------------------------------------------------
                    Send Latency Test
Inline data is used up to 400 bytes message
Connection type : RC
```

```
  local address: LID 0x01 QPN 0x9c0406 PSN 0x94e3e1
  remote address: LID 0x05 QPN 0x250406 PSN 0xbecb59
Mtu : 1024
---------------------------------------------------------------------
 #bytes #iterations    t_min[usec]    t_max[usec]  t_typical[usec]
     2        1000           5.98          31.40              6.14
---------------------------------------------------------------------
#
```

Note – The output is not displayed until the local node client issues the respective command.

Then run the command on the local node client:

```
# ib_send_lat 192.168.200.100
---------------------------------------------------------------------
                    Send Latency Test
Inline data is used up to 400 bytes message
Connection type : RC
  local address: LID 0x05 QPN 0x250406 PSN 0xbecb59
  remote address: LID 0x01 QPN 0x9c0406 PSN 0x94e3e1
Mtu : 1024
---------------------------------------------------------------------
 #bytes #iterations    t_min[usec]    t_max[usec]  t_typical[usec]
     2        1000           5.97          65.62              6.13
---------------------------------------------------------------------
#
```

Related Information

- "ib_read_lat Command" on page 95
- "ib_send_bw Command" on page 98
- "ib_write_lat Command" on page 106

ib_write_bw Command

Performs a write bandwidth diagnostic. Issued on the Linux InfiniBand host.

Syntax

```
ib_write_bw [-p TCP_port][-d device][-i IB_port][-c RC|UC|UD][-m mtu]
[-g posts][-q num][-s size][-a][-t depth][-n iters][-I size][-b][-V]
[-N][-F][IP_address]
```

where:

- *TCP_port* is the TCP port.
- *device* is the InfiniBand device.
- *IB_port* is the InfiniBand port.
- *mtu* is the size of the MTU.
- *posts* is the number of posts for each queue pair.
- *num* is the quantity of queue pairs.
- *size* is the size of the messages.
- *depth* is the size of the TX queue.
- *iters* is the number of message exchanges.
- *IP_address* is the IP address of the remote node host.

Description

This InfiniBand software command performs a bandwidth diagnostic between two nodes in the InfiniBand fabric. The command is dependent upon the Internet Protocol, so the InfiniBand fabric must be configured with Internet Protocol over InfiniBand (IPoIB). The command is a client-server, in that a remote node is configured as a server, while a local node performs as a client.

The command is first run locally on the server. The command is then run again locally on the client, pointing to the IP address of the server. The diagnostic checks the bandwidth of the data transfer from the client to the server. The connection uses the Reliable Datagram transport. Optionally, you can set the connection between the nodes to use Reliable Connection, Unreliable Connection, or Unreliable Datagram transport.

Note – This command only functions if your InfiniBand fabric is configured with IPoIB.

Options

The following table describes the options to the `ib_write_bw` command and their purposes:

Option	Purpose
`-p`	Uses the TCP port for initial synchronization.
`-d`	Uses the InfiniBand device.
`-i`	Uses the InfiniBand port.
`-c`	Sets the connection type.
`-m`	Sets the size of the MTU.
`-g`	Specifies the number of posts for each queue pair in the chain.
`-q`	Sets the number of queue pairs.
`-s`	Sets the size of the message to exchange.
`-a`	Runs all sizes, from 2 to 2^{23}.
`-t`	Sets the size of the TX queue.
`-n`	Performs iters message exchanges.
`-I`	Sets the maximum message size to be sent in inline mode.
`-b`	Measures bidirectional bandwidth.
`-V`	Displays the version information.
`-N`	Cancels the peak bandwidth calculation.
`-F`	Does not fail, even if the `cpufreq_ondemand` module is loaded.

Example

The following example shows how to run a diagnostic between a local node client and a remote node server with the `ib_write_bw` command. First configure the remote node server:

```
# ib_write_bw
------------------------------------------------------------------
                    RDMA_Write BW Test
Number of qp's running 1
Connection type : RC
Each Qp will post up to 100 messages each time
Inline data is used up to 400 bytes message
```

```
  local address:  LID 0x01, QPN 0x9d0406, PSN 0x30ea7b RKey 0x369c003a VAddr
0x002ada954eb000
  remote address: LID 0x05, QPN 0x260406, PSN 0x5e303b, RKey 0x2b0c043f VAddr
0x002b94b3356000
Mtu : 1024
#
```

Note – The output is not displayed until the local node client issues the respective command.

Then run the command on the local node client:

```
# ib_write_bw 192.168.200.100
------------------------------------------------------------------
                    RDMA_Write BW Test
Number of qp's running 1
Connection type : RC
Each Qp will post up to 100 messages each time
Inline data is used up to 400 bytes message
  local address:  LID 0x05, QPN 0x260406, PSN 0x5e303b RKey 0x2b0c043f VAddr
0x002b94b3356000
  remote address: LID 0x01, QPN 0x9d0406, PSN 0x30ea7b, RKey 0x369c003a VAddr
0x002ada954eb000
Mtu : 1024
------------------------------------------------------------------
 #bytes #iterations    BW peak[MB/sec]    BW average[MB/sec]
  65536        5000             790.41               790.41
------------------------------------------------------------------
#
```

Related Information

- "ib_read_bw Command" on page 92
- "ib_send_bw Command" on page 98
- "ib_write_lat Command" on page 106

ib_write_lat Command

Performs a write latency diagnostic. Issued on the Linux InfiniBand host.

Syntax

```
ib_write_lat [-p TCP_port][-d device][-i IB_port][-c RC|UC][-m
mtu][-s size][-a][-t depth][-n iters][-I
size][-C][-H][-U][-V][-F][IP_address]
```

where:

- *TCP_port* is the TCP port.
- *device* is the InfiniBand device.
- *IB_port* is the InfiniBand port.
- *mtu* is the size of the MTU.
- *size* is the size of the messages.
- *depth* is the size of the TX queue.
- *iters* is the number of message exchanges.
- *IP_address* is the IP address of the remote node host.

Description

This InfiniBand software command performs a latency diagnostic between two nodes in the InfiniBand fabric. The command is dependent upon the Internet Protocol, so the InfiniBand fabric must be configured with Internet Protocol over InfiniBand (IPoIB). The command is a client-server, in that a remote node is configured as a server, while a local node performs as a client.

The command is first run locally on the server. The command is then run again locally on the client, pointing to the IP address of the server. The diagnostic checks the latency of the data transfer from the client to the server. The connection uses the Reliable Datagram transport. Optionally, you can set the connection between the nodes to use Reliable Connection or Unreliable Connection transport.

Note – This command only functions if your InfiniBand fabric is configured with IPoIB.

Options

The following table describes the options to the `ib_write_lat` command and their purposes:

Option	Purpose
`-p`	Uses the TCP port for initial synchronization.
`-d`	Uses the InfiniBand device.
`-i`	Uses the InfiniBand port.
`-c`	Sets the connection type.
`-m`	Sets the size of the MTU.
`-s`	Sets the size of the message to exchange.
`-a`	Runs all sizes, from 2 to 223.
`-t`	Sets the size of the TX queue.
`-n`	Performs iters message exchanges.
`-I`	Sets the maximum message size to be sent in inline mode.
`-C`	Reports time as CPU cycle units.
`-H`	Prints all results.
`-U`	Prints all results unsorted.
`-V`	Displays the version information.
`-F`	Does not fail, even if the `cpufreq_ondemand` module is loaded.

Example

The following example shows how to run a diagnostic between a local node client and a remote node server with the `ib_write_lat` command. First configure the remote node server:

```
# ib_write_lat
------------------------------------------------------------------
                    RDMA_Write Latency Test
Inline data is used up to 400 bytes message
Connection type : RC
 local address: LID 0x01 QPN 0x9e0406 PSN 0x41c40f RKey 0x36a2003a VAddr
0x0000001fe70002
  remote address: LID 0x05 QPN 0x270406 PSN 0x5a66a3 RKey 0x2b12043f VAddr
0x00000018c00002
Mtu : 1024
```

```
---------------------------------------------------------------------
 #bytes #iterations    t_min[usec]    t_max[usec]  t_typical[usec]
      2        1000           4.39          30.41             4.65
---------------------------------------------------------------------
#
```

Note – The output is not displayed until the local node client issues the respective command.

Then run the command on the local node client:

```
# ib_write_lat 192.168.200.100
---------------------------------------------------------------------
                    RDMA_Write Latency Test
Inline data is used up to 400 bytes message
Connection type : RC
  local address: LID 0x05 QPN 0x270406 PSN 0x5a66a3 RKey 0x2b12043f VAddr
0x00000018c00002
  remote address: LID 0x01 QPN 0x9e0406 PSN 0x41c40f RKey 0x36a2003a VAddr
0x0000001fe70002
Mtu : 1024
---------------------------------------------------------------------
 #bytes #iterations    t_min[usec]    t_max[usec]  t_typical[usec]
      2        1000           4.20          54.50             4.65
---------------------------------------------------------------------
#
```

Related Information

- "ib_read_lat Command" on page 95
- "ib_send_lat Command" on page 101
- "ib_write_bw Command" on page 103

ibaddr Command

Queries InfiniBand addresses. Issued on the Linux InfiniBand host.

Syntax

`ibaddr [-d][-D][-G][-l][-g][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`][-V][-h] [`*lid*`|`*dr_path*`|`*guid*`]`

where:

- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *lid* is the local identifier.
- *dr_path* is the directed path.
- *guid* is the global unit identifier.

Description

This InfiniBand software command displays the LID and range as well as the GID address of the port specified. The local port information is provided by default.

Note – This command is also used as a simple address resolver.

Options

The following table describes the options to the `ibaddr` command and their purposes:

Option	Purpose
`-d`	Sets the debug level. Can be used several times to increase the debug level.
`-D`	Uses the directed path address. The path is a comma delimited sequence of out ports.
`-G`	Shows the LID range and GID for port GUID addresses.
`-l`	Shows the LID range only.
`-g`	Shows the GID address only.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.
`-t`	Overrides the default timeout.
`-V`	Displays the version information.
`-h`	Provides help.

Example

The following example shows how to display the local port's GID and LID range with the `ibaddr` command.

```
# ibaddr
GID fe80::3:ba00:100:c70a LID start 0x3a end 0x3a
#
```

Related Information

- `ibaddr` man page
- "`ibroute` Command" on page 149
- "`ibtracert` Command" on page 160

`ibcheckerrors` Command

Validates InfiniBand fabric and report errors. Issued on the Linux InfiniBand host.

Syntax

`ibcheckerrors [-h][-b][-v][-N][`*topology*`|-C` *ca_name* `-P` *ca_port* `-t` *timeout*`]`

where:

- *topology* is the topology file.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.

Description

This InfiniBand software command is a script that uses the topology file created by the `ibnetdiscover` command to scan the InfiniBand fabric to validate the connectivity and report errors from the port counters.

Options

The following table describes the options to the `ibcheckerrors` command and their purposes:

Option	Purpose
`-h`	Provides help.
`-b`	Enables brief mode. Reduced output is only if errors are present. Does not identify the errors.
`-v`	Provides verbose output.
`-N`	Uses mono mode instead of color mode.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.
`-t`	Overrides the default timeout.

Example

The following example shows how to check error counters for all LIDs in the InfiniBand fabric with the `ibcheckerrors` command.

```
# ibcheckerrors
#warn: counter SymbolErrors = 65532     (threshold 10) lid 85 port 255
Error check on lid 85 (Sun DCS 648 shmm1500 LC slot 6 switch 1) port all:  FAILED
#warn: counter SymbolErrors = 65535     (threshold 10) lid 1 port 255
Error check on lid 1 (Sun DCS 648 shmm1500 LC slot 6 switch 0) port all:  FAILED
.
.
.
## Summary: 55 nodes checked, 0 bad nodes found
##          1296 ports checked, 0 ports have errors beyond threshold
#
```

Note – The output in the example is just a portion of the full output.

Related Information

- `ibcheckerrors` man page
- "`getportcounters` Command" on page 26
- "`getsymerr` Command" on page 30

- "ibchecknode Command" on page 114
- "ibcheckport Command" on page 116
- "ibnetdiscover Command" on page 140

ibchecknet Command

A simplified version of the ibcheckerrors command. Issued on the Linux InfiniBand host.

Syntax

ibchecknet [-h][-N][*topology*|-C *ca_name* -P *ca_port* -t *timeout*]

where:

- *topology* is the topology file.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.

Description

This InfiniBand software command is a script that uses the topology file created by the ibnetdiscover command to scan the InfiniBand fabric to validate the connectivity and report errors from the port counters.

Options

The following table describes the options to the ibchecknet command and their purposes:

Option	Purpose
-h	Provides help.
-N	Uses mono mode instead of color mode.
-C	Uses the specified channel adapter name.
-P	Uses the specified channel adapter port.
-t	Overrides the default timeout.

Example

The following example shows how to check error counters for all LIDs in the InfiniBand fabric with the `ibchecknet` command.

```
# ibchecknet
#warn: counter SymbolErrors = 66 (threshold 10) lid 24 port 255
Error check on lid 24 (Sun DCS 648 shmm 1500 LC slot 6 switch 0) port 11: FAILED
#warn: Link configured as 1X  lid 24 port 11
Port check lid 24 port 11: FAILED
# Checked Switch: nodeguid 0x00bad0cc003215b2 with failure
#warn: counter SymbolErrors = 10976 (threshold 10) lid 78 port 255
.
.
.
## Summary: 37 nodes checked, 0 bad nodes found
##          552 ports checked, 2 bad ports found
##          14 ports have errors beyond threshold
#
```

Note – The output in the example is just a portion of the full output.

Related Information

- `ibchecknet` man page
- "`ibcheckerrors` Command" on page 111
- "`ibchecknode` Command" on page 114
- "`ibcheckport` Command" on page 116
- "`ibnetdiscover` Command" on page 140

`ibchecknode` Command

Validates InfiniBand nodes and reports errors. Issued on the Linux InfiniBand host.

Syntax

`ibchecknode -v [-h][-N][-G][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]` `[`*lid*`|`*dr_path*`|`*guid*`]`

where:

- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *lid* is the local identifier.
- *guid* is the global unit identifier.

Description

This InfiniBand software command checks node connectivity and performs a simple check to verify the functionality of the specified node.

Note – The port address is a LID, unless the `-G` option is used to specify a GUID address.

Options

The following table describes the options to the `ibchecknode` command and their purposes:

Option	Purpose
`-h`	Provides help.
`-N`	Uses mono mode instead of color mode.
`-G`	Uses the port GUID address.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.
`-t`	Overrides the default timeout.

Example

The following example shows how to check if LID 15, port 26 is active with the `ibchecknode` command.

```
# ibchecknode -v 15
Port check lid 15: OK
#
```

Related Information

- `ibchecknode` man page
- "`ibaddr` Command" on page 109
- "`smpquery` Command" on page 197

`ibcheckport` Command

Validates InfiniBand ports and reports errors. Issued on the Linux InfiniBand host.

Syntax

`ibcheckport [-h][-v][-N][-G][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]` *lid*|*guid* *port*

where:

- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *lid* is the local identifier.
- *guid* is the global unit identifier.
- *port* is the port being validated.

Description

This InfiniBand software command checks port connectivity and performs simple sanity checks for the specified port.

Note – The port address is a LID, unless the `-G` option is used to specify a GUID address.

Options

The following table describes the options to the `ibcheckport` command and their purposes:

Option	Purpose
`-h`	Provides help.
`-v`	Provides verbose output.
`-N`	Uses mono mode instead of color mode.
`-G`	Uses the port GUID address.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.
`-t`	Overrides the default timeout.

Example

The following example shows how to check port 26 on LID 15 with the `ibcheckport` command.

```
# ibcheckport -v 15 26
#error: Physical link state is Disabled  lid 15 port 26
Port check lid 15 port 26:  FAILED
#
```

Related Information

- `ibcheckport` man page
- "`getportstatus` Command" on page 27
- "`ibaddr` Command" on page 109
- "`smpquery` Command" on page 197

`ibcheckportstate` Command

Validates an InfiniBand port. Issued on the Linux InfiniBand host.

Syntax

`ibcheckportstate -v [-G][-h][-N][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`][`*lid*`|`*guid*`] port`

where:

- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *lid* is the local identifier.
- *guid* is the global unit identifier.
- *port* is the port being validated.

Description

This InfiniBand software command checks the specified port for the logical (`Active`) and physical (`LinkUp`) states.

Note – The port address is a LID, unless the `-G` option is used to specify a GUID address.

Options

The following table describes the options to the `ibcheckportstate` command and their purposes:

Option	Purpose
`-G`	Uses the port GUID address.
`-h`	Provides help.
`-N`	Uses mono mode instead of color mode.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.
`-t`	Overrides the default timeout.

Example

The following example shows how to check port 26 on LID 15 with the `ibcheckportstate` command.

```
# ibcheckportstate -v 15 26
#error: Physical link state is Disabled  lid 15 port 26
Port check lid 15 port 26:  FAILED
#
```

Related Information

- `ibcheckportstate` man page
- "`getportstatus` Command" on page 27
- "`ibaddr` Command" on page 109
- "`smpquery` Command" on page 197

`ibcheckportwidth` Command

Validates InfiniBand ports for 1x link width. Issued on the Linux InfiniBand host.

Syntax

`ibcheckportwidth -v [-G][-h][-N][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`][`*lid*|*guid*`]` *port*

where:

- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *guid* is the global unit identifier.
- *port* is the port being validated.

Description

This InfiniBand software command checks connectivity and if the specified port is at 1x link width.

Note – The port address is a LID, unless the `-G` option is used to specify a GUID address.

Options

The following table describes the options to the `ibcheckportwidth` command and their purposes:

Option	Purpose
`-G`	Uses the port GUID address.
`-h`	Provides help.
`-N`	Uses mono mode instead of color mode.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.
`-t`	Overrides the default timeout.

Example

The following example shows how to check the width of port 26 on LID 15 with the `ibcheckportwidth` command.

```
# ibcheckportwidth -v 15 26
Port check lid 15 port 26: OK
#
```

Related Information

- `ibcheckportwidth` man page
- "`getportstatus` Command" on page 27
- "`ibaddr` Command" on page 109
- "`smpquery` Command" on page 197

`ibcheckstate` Command

Displays ports that are `LinkUp` but not `Active`. Issued on the Linux InfiniBand host.

Syntax

`ibcheckstate [-h][-N][-v][`*topology*`|-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]`

where:

- *topology* is the topology file.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.

Description

This InfiniBand software command is a script that uses the topology file created by the `ibnetdiscover` command. The script scans the InfiniBand fabric to validate the port logical and physical state, and reports any ports that have a logical state other than `Active` or a physical state other than `LinkUp`.

Options

The following table describes the options to the `ibcheckstate` command and their purposes:

Option	Purpose
`-h`	Provides help.
`-N`	Uses mono mode instead of color mode.
`-v`	Provides verbose output.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.
`-t`	Overrides the default timeout.

Example

The following example shows how to check the state of all ports with the `ibcheckstate` command.

```
# ibcheckstate -v
# Checking Switch: nodeguid 0x00066a00d80001dd
Node check lid 4:  OK
```

```
Port check lid 4 port 24:  OK
Port check lid 4 port 19:  OK
.
.
.
# Checking Ca: nodeguid 0x0002c90200001818
Node check lid 5:  OK
Port check lid 5 port 1:  OK

## Summary: 5 nodes checked, 0 bad nodes found
##          10 ports checked, 0 ports with bad state found
#
```

Note – The output in the example is just a portion of the full output.

Related Information

- `ibcheckstate` man page
- "`ibchecknode` Command" on page 114
- "`ibcheckportstate` Command" on page 117
- "`ibnetdiscover` Command" on page 140

`ibcheckwidth` Command

Finds 1x links in the InfiniBand fabric. Issued on the Linux InfiniBand host.

Syntax

`ibcheckwidth [-h][-N][-v][`*topology*`|-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]`

where:

- *topology* is the topology file.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.

Description

This InfiniBand software command is a script that uses the topology file created by the `ibnetdiscover` command. The script scans the InfiniBand fabric to validate `Active` link widths and report those which are 1x links.

Options

The following table describes the options to the `ibcheckwidth` command and their purposes:

Option	Purpose
`-h`	Provides help.
`-N`	Uses mono mode instead of color mode.
`-v`	Provides verbose output.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.
`-t`	Overrides the default timeout.

Example

The following example shows how to check the 1x links for all ports with the `ibcheckwidth` command.

```
# ibcheckwidth -v
# Checking Switch: nodeguid 0x00066a00d80001dd
Node check lid 4:  OK
Port check lid 4 port 24:  OK
Port check lid 4 port 19:  OK
Port check lid 4 port 17:  OK
.
.
.
# Checking Ca: nodeguid 0x0002c90200001818
Node check lid 5:  OK
Port check lid 5 port 1:  OK
## Summary: 5 nodes checked, 0 bad nodes found
##          10 ports checked, 0 ports with 1x width in error found
#
```

Note – The output in the example is just a portion of the full output.

Related Information

- `ibcheckwidth` man page
- "`ibchecknode` Command" on page 114
- "`ibcheckportwidth` Command" on page 119
- "`ibnetdiscover` Command" on page 140

`ibclearcounters` Command

Clears port counters in the InfiniBand fabric. Issued on the Linux InfiniBand host.

Syntax

`ibclearcounters [-h][`*topology*`|-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]`

where:

- *topology* is the topology file.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.

Description

This InfiniBand software command is a script that clears the Performance Manager agent port counters by either discovering the InfiniBand fabric topology or using an existing topology file. The counters are:

- `XmtData`
- `RcvData`
- `XmtPkts`
- `RcvPkts`

Options

The following table describes the options to the `ibclearcounters` command and their purposes:

Option	Purpose
`-h`	Provides help.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.
`-t`	Overrides the default timeout.

Example

The following example shows how to clear the Performance Manager agent port counters with the `ibclearcounters` command.

```
# ibclearcounters
# Summary: 74 nodes cleared 0 errors
#
```

Related Information

- `ibclearcounters` man page
- "`ibnetdiscover` Command" on page 140
- "`perfquery` Command" on page 188

`ibclearerrors` Command

Clears error counters in the InfiniBand fabric. Issued on the Linux InfiniBand host.

Syntax

`ibclearerrors [-h][-N][`*topology*`|-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]`

where:

- *topology* is the topology file.
- *ca_name* is the channel adapter name.

- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.

Description

This InfiniBand software command is a script that clears the Performance Manager agent error counters in `PortCounters` by either discovering the InfiniBand fabric topology or using an existing topology file.

Options

The following table describes the options to the `ibclearerrors` command and their purposes:

Option	Purpose
`-h`	Provides help.
`-N`	Uses mono mode instead of color mode.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.
`-t`	Overrides the default timeout.

Example

The following example shows how to clear all error counters with the `ibclearerrors` command.

```
# ibclearerrors
## Summary: 5 nodes cleared 0 errors
#
```

Related Information

- `ibclearerrors` man page
- "`clearboardstat` Command" on page 11
- "`getsymerr` Command" on page 30
- "`ibnetdiscover` Command" on page 140
- "`perfquery` Command" on page 188

`ibdatacounters` Command

Queries the InfiniBand fabric for data counters. Issued on the Linux InfiniBand host.

Syntax

`ibdatacounters [-b][-h][-N][-v][`*topology*`|-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]`

where:

- *topology* is the topology file.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.

Description

This InfiniBand software command is a script that uses the topology file created by the `ibnetdiscover` command. The script scans the InfiniBand fabric to validate the connectivity and reports the values of the data counters.

Options

The following table describes the options to the `ibdatacounters` command and their purposes:

Option	Purpose
`-b`	Enables brief mode. Reduced output is only if errors are present. Does not identify the errors.
`-h`	Provides help.
`-N`	Uses mono mode instead of color mode.
`-v`	Provides verbose output.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.
`-t`	Overrides the default timeout.

Example

The following example shows how to display the data counters for all ports with the `ibdatacounters` command.

```
# ibdatacounters
Error check on lid 85 (Sun DCS 648 shmm1500 LC slot 6 switch 1) port all:  FAILED
# Port counters: Lid 85 port 1
XmtData:.........................360
RcvData:.........................360
XmtPkts:.........................5
RcvPkts:.........................5
# Port counters: Lid 85 port 2
.
.
.
RcvPkts:.........................147
## Summary: 55 nodes checked, 0 bad nodes found
##          1296 ports checked
#
```

Note – The output in the example is just a portion of the full output.

Related Information

- `ibdatacounters` man page
- "`ibdatacounts` Command" on page 128
- "`ibnetdiscover` Command" on page 140

`ibdatacounts` Command

Displays InfiniBand fabric port data counters. Issued on the Linux InfiniBand host.

Syntax

`ibdatacounts [-b][-G][-h][-N][-v][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]` *lid*|*guid* *port*

where:

- *ca_name* is the channel adapter name.

- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *lid* is the local identifier.
- *guid* is the global unit identifier.
- *port* is the port being validated.

Description

This InfiniBand software command returns the Performance Manager agent data counters from a specified port or node.

Note – The port address is a LID, unless the `-G` option is used to specify a GUID address.

Options

The following table describes the options to the `ibdatacounts` command and their purposes:

Option	Purpose
`-b`	Enables brief mode. Reduced output is only if errors are present. Does not identify the errors.
`-G`	Uses the port GUID address.
`-h`	Provides help.
`-N`	Uses mono mode instead of color mode.
`-v`	Provides verbose output.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.
`-t`	Overrides the default timeout.

Example

The following example shows how to display the data counters for LID 4, port 14 with the `ibdatacounts` command.

```
# ibdatacounts 4 14
# Port counters: Lid 4 port 14
XmtData:.......................1704
RcvData:.......................1041
XmtPkts:.......................26
RcvPkts:.......................15
#
```

Related Information

- `ibdatacounts` man page
- "`getportcounters` Command" on page 26
- "`ibaddr` Command" on page 109
- "`perfquery` Command" on page 188

`ibdiagnet` Command

Performs InfiniBand fabric diagnostic. Issued on the Linux InfiniBand host.

Syntax

`ibdiagnet [-c` *count*`][-v][-r][-o` *outputdir*`][-t` *topology*`][-s` *system*`][-i` *device*`][-p` *port*`][-wt` *topology*`][-pm][-pc][-P` *PM* `=` *value*`][-lw 1x|4x|12x][-ls 2.5|5|10][-skip` *checks*`][-load_db` *file*`][-h][-V]`

where:

- *count* is the number of packets.
- *outputdir* is the output directory.
- *topology* is the topology file.
- *system* is the local system name.
- *device* is the index of the device connecting to the InfiniBand fabric.
- *port* is the port of the device.
- *PM* is the Performance Manager counter number.
- *value* is the threshold of the Performance Manager counter.

- *checks* is one or more strings that identifies the checks made:
  - `dup_guids`
  - `zero_guids`
  - `pm`
  - `logical_state`
  - `part`
  - `ipoib`
  - `all`
- *file* is the subnet database `.db` file.

Description

This InfiniBand software command scans the InfiniBand fabric using directed route packets, extracting all the available information regarding the connectivity and devices. This command produces a set of files in the output directory. By default, the output directory is `/tmp`. The following table describes the files.

File Name	Description
`ibdiagnet.log`	Dump of all the application reports generated according to the provided flags.
`ibdiagnet.lst`	List of all the nodes, ports, and links in the fabric.
`ibdiagnet.fdbs`	Dump of the unicast forwarding tables of the fabric switches.
`ibdiagnet.mcfdbs`	Dump of the multicast forwarding tables of the fabric switches.
`ibdiagnet.masks`	In case of duplicate port/node GUIDs, these file include the map between masked GUIDs and real GUIDs.
`ibdiagnet.sm`	List of all the Subnet Manager (state and priority) in the fabric.
`ibdiagnet.pm`	Dump of the Performance Manager counters values, for the fabric links.
`ibdiagnet.pkey`	Dump of the existing partitions and their member host ports.
`ibdiagnet.mcg`	Dump of the multicast groups, their properties, and member host ports.
`ibdiagnet.db`	Dump of the internal subnet database. This file can be loaded in later runs using the `-load_db` option.

During the discovery phase, the command also checks for duplicate node/port GUIDs in the InfiniBand fabric. If such an error is detected, it is displayed on the standard output.

After the discovery phase is completed, directed route packets are sent multiple times to detect possible problematic paths on which packets might be lost. A report of suspected bad links is displayed on the standard output.

If requested with the `-r` option, a full report of fabric qualities is displayed, including:

- Subnet Manager report
- Number of nodes and systems
- Hop-count information containing maximal hop-count, an example path, and a hop-count histogram
- All CA-to-CA paths traced
- Credit loop report
- MGID-MLID-HCAs multicast group and report
- Partitions report
- IPoIB report

Note – If the InfiniBand fabric includes only one CA, then CA-to-CA paths are not reported. Additionally, if a topology file is provided, the `ibdiagnet` command uses the names defined in the topology file for the output reports.

Options

The following table describes the options to the `ibdiagnet` command and their purposes:

Option	Purpose
`-c`	Sets the minimum number of packets sent across each link.
`-v`	Provides verbose output.
`-r`	Provides a report of fabric qualities.
`-t`	Specifies the topology file name.
`-s`	Specifies the local system name.
`-i`	In the case of multiple devices on the local system, this option specifies the index of the device of the port used to connect to the InfiniBand fabric.
`-p`	Specifies the local device port number used to connect to the InfiniBand fabric.
`-o`	Specifies the output directory.
`-lw`	Specifies the expected link width.
`-ls`	Specifies the expected link speed.
`-pm`	Dumps all the fabric link Performance Manager counters into `ibdiagnet.pm`.

Option	Purpose
-pc	Resets all the fabric link Performance Manager counters.
-P	Use the Performance Manager counter of PM set to the threshold of value.
-skip	Skips the executions of the selected checks. One or more checks can be specified.
-wt	Writes out the discovered topology into the given file.
-load_db	Loads subnet data from the given .db file and skips the subnet discovery stage. **Note -** Some checks require actual subnet discovery, and are disabled if load_db is specified. Those checks are duplicate/zero GUIDs, link state, and Subnet Manager status.
-h	Provides help.
-V	Displays the version information.

Example

The following example shows how to test the InfiniBand fabric with the ibdiagnet command. The command checks for 4x link width and 10 Gbyte/sec speed and dumps the performance manager counters and then clears them.

```
# ibdiagnet -lw 4x -ls 10 -pm -pc
Loading IBDIAGNET from: /usr/lib64/ibdiagnet1.2
-W- Topology file is not specified.
  Reports regarding cluster links will use direct routes.
Loading IBDM from: /usr/lib64/ibdm1.2
-I- Using port 2 as the local port.
-I- Discovering ... 55 nodes (54 Switches & 1 CA-s) discovered.
-I---------------------------------------------------
-I- Bad Guids/LIDs Info
-I---------------------------------------------------
-I- No bad Guids were found
-I---------------------------------------------------
-I- Links With Logical State = INIT
-I---------------------------------------------------
-I- No bad Links (with logical state = INIT) were found
-I---------------------------------------------------
-I- PM Counters Info
-I---------------------------------------------------
-I- No illegal PM counters values were found
-I---------------------------------------------------
-I- Links With links width != 4x (as set by -lw option)
-I---------------------------------------------------
-I- No unmatched Links (with width != 4x) were found
```

```
-I---------------------------------------------------
-I- Links With links speed != 10 (as set by -ls option)
-I- No unmatched Links (with speed != 10) were found
-I---------------------------------------------------
-I- Fabric Partitions Report (see ibdiagnet.pkey for a full hosts list)
-I---------------------------------------------------
-I-    PKey:0x7fff Hosts:1 full:1 partial:0
-I---------------------------------------------------
-I- IPoIB Subnets Check
-I---------------------------------------------------
-I- Subnet: IPv4 PKey:0x7fff QKey:0x00000b1b MTU:2048Byte rate:10Gbps SL:0x00
-W- Suboptimal rate for group. Lowest member rate:40Gbps > group-rate:10Gbps
-I---------------------------------------------------
-I- Bad Links Info
-I- No bad link were found
-I---------------------------------------------------
----------------------------------------------------------------
-I- Stages Status Report:
  STAGE                                 Errors Warnings
  Bad GUIDs/LIDs Check                  0      0
  Link State Active Check               0      0
  Performance Counters Report           0      0
  Specific Link Width Check             0      0
  Specific Link Speed Check             0      0
  Partitions Check                      0      0
  IPoIB Subnets Check                   0      1
Please see /tmp/ibdiagnet.log for complete log
----------------------------------------------------------------
-I- Done. Run time was 42 seconds.
#
```

Related Information

- `ibdiagnet` man page
- "`ibdiagpath` Command" on page 134

`ibdiagpath` Command

Traces the InfiniBand fabric diagnostic path. Issued on the Linux InfiniBand host.

Syntax

`ibdiagpath -n[`*src_name*`,]`*dst_name*`|-l[`*src_lid*`,]`*dst_lid*`|-d` *p1,p2,p3,...*`[-c` *count*`][-v][-o` *outputdir*`][-t` *topology*`][-s` *system*`][-i` *device*`][-p` *port*`][-wt` *topology*`][-pm][-pc][-P` *PM* `=` *value*`][-lw 1x|4x|12x][-ls 2.5|5|10][-h][-V]`

where:

- *src_name* is the source port.
- *dst_name* is the destination port.
- *src_lid* is the source LID.
- *dst_lid* is the destination LID.
- *p1,p2,p3,...* is the directed route.
- *count* is the number of packets.
- *outputdir* is the output directory.
- *topology* is the topology file.
- *system* is the local system name.
- *device* is the index of the device connecting to the InfiniBand fabric.
- *port* is the port of the device.
- *PM* is the performance manager counter number.
- *value* is the threshold of the performance manager counter.

Description

This InfiniBand software command traces a path between two end-points and provides information regarding the nodes and ports traversed along the path. The command uses device-specific health queries for the different devices encountered. The way the `ibdiagpath` command operates is determined from the addressing mode specified on the command line:

- If directed route addressing is used, the local node is the source node and the route to the destination port is known.
- If LID route addressing is used, the source and destination ports of a route are specified by their LIDs.

In LID route addressing, the actual path from the local port to the source port, and from the source port to the destination port, is defined by means of Subnet Management Linear Forwarding Table queries of the switch nodes along those paths.

Note – When the `ibdiagpath` command queries the performance counters along the path between the source and destination ports, it always traverses the LID route, regardless of whether a directed route is specified. If one or more links along the LID route are not in the `Active` state, the `ibdiagpath` command reports an error.

This command produces a set of files in the output directory. By default, the output directory is `/tmp`. The following table describes the files.

File Name	Description
`ibdiagpath.log`	Dump of all the application reports generated according to the provided flags.
`ibdiagpath.pm`	Dump of the Performance Manager counters values, for the fabric links.

Options

The following table describes the options to the `ibdiagpath` command and their purposes:

Option	Purpose
`-n`	Identifies the source and destination ports.
`-l`	Identifies the source and destination LIDs.
`-d`	Sets directed route from the local node to the destination node.
`-c`	Sets the minimum number of packets sent across each link.
`-v`	Provides verbose output.
`-t`	Specifies the topology file name.
`-s`	Specifies the local system name.
`-i`	In the case of multiple devices on the local system, this option specifies the index of the device of the port used to connect to the InfiniBand fabric.
`-p`	Specifies the local device port number used to connect to the InfiniBand fabric.
`-o`	Specifies the output directory.
`-lw`	Specifies the expected link width.
`-ls`	Specifies the expected link speed.
`-pm`	Dumps all the fabric link Performance Manager counters into `ibdiagpath.pm`.
`-pc`	Resets all the fabric link Performance Manager counters.
`-P`	Use the Performance Manager counter of PM set to the threshold of value.
`-h`	Provides help.

Option	Purpose
-V	Displays the version information.

Example

The following example shows how to display the route from LID 1 to LID 5 with the `ibdiagpath` command.

```
# ibdiagpath -l 1,5
Loading IBDIAGPATH from: /usr/lib64/ibdiagpath1.2
-W- Topology file is not specified.
  Reports regarding cluster links will use direct routes.
Loading IBDM from: /usr/lib64/ibdm1.2
-I- Using port 2 as the local port.
-I----------------------------------------------------
-I- Traversing the path from local to source
-I----------------------------------------------------
-I- From: lid=0x003a guid=0x0003ba000100c70a dev=26428 nsn34-39/P2
-I- To:   lid=0x0001 guid=0x00bad0cc010016a2 dev=48438 Port=22
-I----------------------------------------------------
-I- Traversing the path from source to destination
-I----------------------------------------------------
-I- From: lid=0x0001 guid=0x00bad0cc010016a2 dev=48438 Port=1
-I- To:   lid=0x0039 guid=0x00bad0fc0100f8b2 dev=48438 Port=7
-I- From: lid=0x0039 guid=0x00bad0fc0100f8b2 dev=48438 Port=13
-I- To:   lid=0x0005 guid=0x0021283a841513a2 dev=48438 Port=1
-I----------------------------------------------------
-I- PM Counters Info
-I----------------------------------------------------
-I- No illegal PM counters values were found
-I----------------------------------------------------
-I- Path Partitions Report
-I----------------------------------------------------
-I- Source Port=22 lid=0x0001 guid=0x00bad0cc010016a2 dev=48438 Port 22
     PKeys:0xffff 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
     0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
0x0000
     0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
0x0000
-I- Destination  lid=0x0005 guid=0x0021283a841513a2 dev=48438 PKeys:Not-Enforced
-I- Path shared PKeys: 0xffff
-I----------------------------------------------------
-I- IPoIB Path Check
-I----------------------------------------------------
-I- Subnet: IPv4 PKey:0x7fff QKey:0x00000b1b MTU:2048Byte rate:10Gbps SL:0x00
-I----------------------------------------------------
```

```
-I- QoS on Path Check
-I-------------------------------------------------
-I- The following SLs can be used:0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
----------------------------------------------------------------
-I- Stages Status Report:
  STAGE                                  Errors Warnings
  LFT Traversal: local to source         0      0
  LFT Traversal: source to destination   0      0
  Performance Counters Report            0      0
  Path Partitions Check                  0      0
  Path IPoIB Check                       0      0
  QoS on Path Check                      0      0
Please see /tmp/ibdiagpath.log for complete log
----------------------------------------------------------------
-I- Done. Run time was 0 seconds.
#
```

Related Information

- `ibdiagpath` man page
- "`ibdiagnet` Command" on page 130

`ibhosts` Command

Displays host nodes. Issued on the Linux InfiniBand host.

Syntax

`ibhosts [-h][`*topology*`|-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]`

where:

- *topology* is the topology file.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.

Description

This InfiniBand software command is a script that discovers the InfiniBand fabric topology or uses the existing topology file to extract the channel adapter nodes.

Options

The following table describes the options to the `ibhosts` command and their purposes:

Option	Purpose
`-h`	Provides help.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.
`-t`	Overrides the default timeout.

Example

The following example shows how to display the host node GUIDs with the `ibhosts` command.

```
# ibhosts
Ca      : 0x5080020000911314 ports 1 "nsn32-50 HCA-1"
Ca      : 0x5080020000911310 ports 1 "nsn32-20 HCA-1"
Ca      : 0x50800200008e532c ports 1 "ib-71 HCA-1"
Ca      : 0x50800200008e5328 ports 1 "ib-70 HCA-1"
Ca      : 0x50800200008296a4 ports 2 "ib-90 HCA-1"
Ca      : 0x50800200008296a0 ports 2 "ib-91 HCA-1"
Ca      : 0x508002000082dc34 ports 1 "ib-231 HCA-2"
Ca      : 0x0144f6c666b50100 ports 2 "nsn32-10 HCA-1"
.
.
.
#
```

Note – The output in the example is just a portion of the full output and varies for each InfiniBand topology.

Related Information

- `ibhosts` man page
- "`ibnetdiscover` Command" on page 140
- "`ibnodes` Command" on page 143

ibnetdiscover Command

Discovers the InfiniBand topology. Issued on the Linux InfiniBand host.

Syntax

ibnetdiscover [-d][-e][-v][-s][-l][-g][-H][-S][-R][-C *ca_name*][-P *ca_port*][-t *timeout*][-V][--node-name-map *map*][-p][-h][*topology*]

where:

- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *map* is the file name of the node name map.
- *topology* is the topology file.

Description

This InfiniBand software command performs InfiniBand fabric discovery and outputs a human readable topology file. Nodes, node types, node descriptions, links, port numbers, port LIDs, and GUIDs are displayed. The output is directed to a topology file, if the file name is specified.

The output of the topology file follows this basic format for each node:

```
vendid=vendor_ID_in_hex
devid=device _ID_in_hex
and
sysimgguid=GUID_in_hex
and/or
switchguid=GUID_in_hex(portGUID_in_hex)
Switch ports_total "type-nodeGUID_in_hex" # "NodeDescription" base port 0 lid LID lmc 0
or
caguid=GUID_in_hex
Ca ports_total "type-nodeGUID_in_hex" # "NodeDescription"
and
[port] "type-nodeGUID_in_hex"[port](portGUID_in_hex) # "NodeDescription" lid LID widthspeed
[port] "type-nodeGUID_in_hex"[port](portGUID_in_hex) # "NodeDescription" lid LID widthspeed
.
.
.
```

For example:

```
vendid=0x8f1
devid=0x5a05
switchguid=0x8f10400410015(8f10400410015)
Switch  8 "S-0008f10400410015" # "SW-6IB4 Voltaire" base port 0 lid 3 lmc 0
[6] "H-0008f10403960984"[1](8f10403960985) # "MT23108 InfiniHost Mellanox
Technologies" lid 16 4xSDR
[4] "H-005442b100004900"[1](5442b100004901) # "MT23108 InfiniHost Mellanox
Technologies" lid 12 4xSDR
.
.
.
```

Options

The following table describes the options to the `ibnetdiscover` command and their purposes:

Option	Purpose
`-d`	Sets the debug level. Can be used several times to increase the debug level.
`-e`	Displays send and receive errors.
`-v`	Provides verbose output.
`-s`	Shows more information.
`-l`	Lists the connected nodes.
`-g`	Shows the grouping and switch external ports correspondence.
`-H`	Lists the connected channel adapters.
`-S`	Lists the connected switches.
`-R`	Lists the connected routers.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.
`-t`	Overrides the default timeout.
`-V`	Displays the version information.
`--node-name-map`	Reads the node name map file.

Option	Purpose
-p	Returns a list of connected ports, including status information: • LID • portnum • GUID • link width • link speed • NodeDescription
-h	Provides help.

Example

The following example shows how to discover the InfiniBand fabric topology with the ibnetdiscover command.

```
# ibnetdiscover
#
# Topology file: generated on Thu Oct 29 16:49:36 2009
#
# Initiated from node 0003ba000100c708 port 0003ba000100c70a
vendid=0x2c9
devid=0xbd36
sysimgguid=0x3ba2550282543
switchguid=0x21283a83ae11d2(21283a83ae11d2)
Switch  36 "S-0021283a83ae11d2" # "Sun DCS 648 shmm1500 LC slot 1 switch 3" base
port 0 lid 35 lmc 0
 [1]     "S-00bad0fc0100f8a2"[19] # "Sun DCS 648 shmm1500 FC slot 8 switch 0"
lid 56 4xQDR
 [2]     "S-00bad0fc0100f8b2"[36] # "Sun DCS 648 shmm1500 FC slot 8 switch 1"
lid 57 4xQDR
 [3]     "S-0021283a77bdf7a2"[19] # "Sun DCS 648 shmm1500 FC slot 7 switch 0"
lid 54 4xQDR
.
.
.
vendid=0x2c9
devid=0x673c
sysimgguid=0x3ba000100c70b
caguid=0x3ba000100c708
Ca      2 "H-0003ba000100c708" # "nsn34-39 HCA-1"
[2](3ba000100c70a)      "S-00bad0cc010016a2"[22]                # lid 58 lmc 0
"Sun DCS 648 shmm1500 LC slot 6 switch 0" lid 1 4xQDR
#
```

Note – The output in the example is just a portion of the full output.

Related Information

- `ibnetdiscover` man page

`ibnodes` Command

Displays InfiniBand nodes in topology. Issued on the Linux InfiniBand host.

Syntax

`ibnodes [-h][`*topology*`|-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]`

where:

- *topology* is the topology file.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.

Description

This InfiniBand software command is a script that discovers the InfiniBand fabric topology or uses the existing topology file to extract the InfiniBand nodes of the channel adapters, switches, and routers.

Options

The following table describes the options to the `ibnodes` command and their purposes:

Option	Purpose
`-h`	Provides help.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.

Option	Purpose
-t	Overrides the default timeout.

Example

The following example shows how to display the node GUIDs with the `ibnodes` command.

```
# ibnodes
Ca      : 0x0003ba000100c708 ports 2 "nsn34-39 HCA-1"
Switch  : 0x0021283a83ae11d2 ports 36 "Sun DCS 648 shmm1500 LC slot 1 switch 3"
base port 0 lid 35 lmc 0
Switch  : 0x0021283a83b112b2 ports 36 "Sun DCS 648 shmm1500 LC slot 2 switch 1"
base port 0 lid 17 lmc 0
Switch  : 0x0021283a83b112a2 ports 36 "Sun DCS 648 shmm1500 LC slot 2 switch 0"
base port 0 lid 4 lmc 0
Switch  : 0x0021283a83b112d2 ports 36 "Sun DCS 648 shmm1500 LC slot 2 switch 3"
base port 0 lid 36 lmc 0
.
.
.
#
```

Note – The output in the example is just a portion of the full output and varies for each InfiniBand topology.

Related Information

- `ibnodes` man page
- "`getbaseguid` Command" on page 22
- "`ibnetdiscover` Command" on page 140

`ibping` Command

Pings an InfiniBand address. Issued on the Linux InfiniBand host.

Syntax

`ibping [-d][-e][-G][-h][-s` *smlid*`][-v][-V][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`][-c` *count*`][-f][-o` *oui*`][-S]` *lid*|*guid*

where:

- *smlid* is the Subnet Manager LID.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *count* is the number of packets.
- *oui* is the OUI number.
- *lid* is the local identifier.
- *guid* is the global unit identifier.

Description

This InfiniBand software command uses vendor supplied management datagrams to validate connectivity between InfiniBand nodes. The command is a client-server, in that a remote node is configured as a server, while a local node performs as a client. Output is similar to the Internet Protocol `ping` command.

The command is first run locally on the server. The command is then run again locally on the client, pointing to the LID of the server. The ping repeats every second, until the Ctrl-C key combination exits on the server.

Options

The following table describes the options to the `ibping` command and their purposes:

Option	Purpose
`-d`	Sets the debug level. Can be used several times to increase the debug level.
`-e`	Displays send and receive errors.
`-G`	Uses the port GUID address.
`-h`	Provides help.
`-s`	Uses *smlid* as the target LID for Subnet Manager or Subnet Administrator queries.

Option	Purpose
-v	Provides verbose output.
-V	Displays the version information.
-C	Uses the specified channel adapter name.
-P	Uses the specified channel adapter port.
-t	Overrides the default timeout.
-c	Stops after *count* packets.
-f	Sends packets back to back without delay.
-o	Uses specified OUI number to multiplex vendor management datagrams.
-S	Starts in server mode.

Example

The following example shows how to ping a remote node server from a local node client with the `ibping` command. First configure the remote node server:

```
# ibping -S
<CTRL-C to exit server>
#
```

Then run the command on the local node client:

```
# ibping 1
Pong from nsn105-100.nsn.sfbay.sun.com (Lid 1): time 0.205 ms
Pong from nsn105-100.nsn.sfbay.sun.com (Lid 1): time 0.102 ms
Pong from nsn105-100.nsn.sfbay.sun.com (Lid 1): time 0.095 ms
Pong from nsn105-100.nsn.sfbay.sun.com (Lid 1): time 0.096 ms
Pong from nsn105-100.nsn.sfbay.sun.com (Lid 1): time 0.102 ms
Pong from nsn105-100.nsn.sfbay.sun.com (Lid 1): time 0.093 ms
Pong from nsn105-100.nsn.sfbay.sun.com (Lid 1): time 0.096 ms
Pong from nsn105-100.nsn.sfbay.sun.com (Lid 1): time 0.091 ms
--- nsn105-100.nsn.sfbay.sun.com (Lid 1) ibping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8557 ms
rtt min/avg/max = 0.091/0.108/0.205 ms
#
```

Related Information

- `ibping` man page

`ibportstate` Command

Manages the state and link speed of an InfiniBand port. Issued on the Linux InfiniBand host.

Syntax

`ibportstate [-d][-D][-e][-G][-h][-s` *smlid*`][-v][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]` *lid*|*dr_path*|*guid* *port* `[`*op*`]`

where:

- *smlid* is the Subnet Manager LID.
- *topology* is the topology file.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *lid* is the local identifier.
- *dr_path* is the directed path.
- *guid* is the global unit identifier.
- *port* is the port being validated.
- *op* is the operation to perform on the port:
  - `enable`
  - `disable`
  - `reset`
  - `speed` *number* (where *number* is `1` for SDR, `2` for DDR, and `4` for QDR)
  - `query` (default)

Description

This InfiniBand software command queries the logical and physical state of an InfiniBand port. The command can return the link width and speed of a switch port, as well as enabling, disabling, or resetting the port. The command can also set the link speed of any InfiniBand port.

Note – Speed changes are not affected until the port undergoes link renegotiation. Additionally, speed values are additive for enabling. For example, `speed 7` is 2.5, 5.0, and 10.0 Gbyte/sec.

Options

The following table describes the options to the `ibportstate` command and their purposes:

Option	Purpose
`-d`	Sets the debug level. Can be used several times to increase the debug level.
`-D`	Uses the directed path address. The path is a comma delimited sequence of out ports.
`-e`	Displays send and receive errors.
`-G`	Uses the port GUID address.
`-h`	Provides help.
`-s`	Uses *smlid* as the target LID for Subnet Manager or Subnet Administrator queries.
`-v`	Provides verbose output.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.
`-t`	Overrides the default timeout.

Example

The following example shows how to query the state and link speed of LID 15, port 14 with the `ibportstate` command.

```
# ibportstate 15 14
PortInfo:
# Port info: Lid 15 port 14
LinkState:.....................Active
PhysLinkState:.................LinkUp
LinkWidthSupported:............1X or 4X
LinkWidthEnabled:..............1X or 4X
LinkWidthActive:...............4X
LinkSpeedSupported:............2.5 Gbps or 5.0 Gbps or 10.0 Gbps
LinkSpeedEnabled:..............2.5 Gbps or 5.0 Gbps or 10.0 Gbps
LinkSpeedActive:...............10.0 Gbps
Peer PortInfo:
# Port info: Lid 15 DR path slid 1; dlid 65535; 0,14 port 10
LinkState:.....................Active
PhysLinkState:.................LinkUp
LinkWidthSupported:............1X or 4X
LinkWidthEnabled:..............1X or 4X
```

```
LinkWidthActive:.................4X
LinkSpeedSupported:..............2.5 Gbps or 5.0 Gbps or 10.0 Gbps
LinkSpeedEnabled:................2.5 Gbps or 5.0 Gbps or 10.0 Gbps
LinkSpeedActive:.................10.0 Gbps
#
```

Related Information

- `ibportstate` man page
- "`disableswitchport` Command" on page 15
- "`enableswitchport` Command" on page 19
- "`resetswitch` Command" on page 34
- "`setlinkspeed` Command" on page 36

`ibroute` Command

Queries InfiniBand switch forwarding tables. Issued on the Linux InfiniBand host.

Syntax

`ibroute [-d][-a][-n][-D][-e][-G][-h][-M][-s` *smlid*`][-v][-V][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`][`*lid*`|`*dr_path*`|`*guid* `[`*startlid* `[`*endlid*`]]]`

where:

- *smlid* is the Subnet Manager LID.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *lid* is the local identifier.
- *dr_path* is the directed path.
- *guid* is the global unit identifier.
- *startlid* is the starting local identifier.
- *endlid* is the ending local identifier.

Description

This InfiniBand software command uses SMPs to display the forwarding tables for the specified switch LID and optionally, the LID range. By default, the range is all valid entries from `1` to `FDBTop`.

Options

The following table describes the options to the `ibroute` command and their purposes:

Option	Purpose
`-a`	Shows all LIDs in the range, including invalid entries.
`-n`	Does not try to resolve destinations.
`-d`	Sets the debug level. Can be used several times to increase the debug level.
`-D`	Uses the directed path address. The path is a comma delimited sequence of out ports.
`-e`	Displays send and receive errors.
`-G`	Uses the port GUID address.
`-h`	Provides help.
`-M`	Shows multicast forwarding tables.
`-s`	Uses *smlid* as the target LID for Subnet Manager or Subnet Administrator queries.
`-v`	Provides verbose output.
`-V`	Displays the version information.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.
`-t`	Overrides the default timeout.

Example

The following example shows how to display the forwarding table for LID 15 with the `ibroute` command.

```
# ibroute 15
Unicast lids [0x0-0x55] of switch Lid 15 guid 0x0021283a83b710b2 (Sun DCS 648
shmm1500 LC slot 0 switch 1):
  Lid  Out   Destination
     Port     Info
0x0001 002 : (Switch portguid 0x00bad0cc010016a2: 'Sun DCS 648 shmm1500 LC slot
6 switch 0')
0x0002 002 : (Switch portguid 0x0021283a83b710a2: 'Sun DCS 648 shmm1500 LC slot
0 switch 0')
0x0003 002 : (Switch portguid 0x0021283a83ae11a2: 'Sun DCS 648 shmm1500 LC slot
1 switch 0')
0x0004 002 : (Switch portguid 0x0021283a83b112a2: 'Sun DCS 648 shmm1500 LC slot
2 switch 0')
.
.
.
0x003a 001 : (Channel Adapter portguid 0x0003ba000100c70a: 'nsn34-39 HCA-1')
0x0055 002 : (Switch portguid 0x00bad0cc010016b2: 'Sun DCS 648 shmm1500 LC slot
6 switch 1')
55 valid lids dumped
#
```

Note – The output in the example is just a portion of the full output.

Related Information

- `ibroute` man page
- "`ibtracert` Command" on page 160

`ibrouters` Command

Displays InfiniBand router nodes in topology. Issued on the Linux InfiniBand host.

Syntax

`ibrouters [-h][`*topology*`|-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]`

where:

- *topology* is the topology file.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.

Description

This InfiniBand software command is a script that discovers the InfiniBand fabric topology or uses an existing topology file to extract the router nodes.

Options

The following table describes the options to the `ibrouters` command and their purposes:

Option	Purpose
`-h`	Provides help.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.
`-t`	Overrides the default timeout.

Example

The following example shows how to display router nodes with the `ibrouters` command.

```
# ibrouters
Router : 0x00066a00d80003fd ports 24 "Hellios 4024 Router " enhanced port 0 lid
7 lmc 0
#
```

Note – The output in the example is just a portion of the full output and varies for each InfiniBand topology.

Related Information

- `ibrouters` man page
- "`ibnetdiscover` Command" on page 140
- "`ibnodes` Command" on page 143

`ibstat` Command

Queries basic status of InfiniBand devices. Issued on the Linux InfiniBand host.

Syntax

`ibstat [-d][-e][-h][-l][-s][-p][-V]` *ca_name* `[`*ca_port*`]`

where:

- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.

Description

This InfiniBand software command displays basic information retrieved from the local InfiniBand driver. Output of the command includes:

- `LID`
- `SMLID`
- `port logical state`
- `link width`
- `port physical state`

The `ibstat` command is similar to the `ibstatus` command, however, the `ibstat` command is a binary executable, has options to display channel adapters and ports, and provides more information than the `ibstatus` command.

Options

The following table describes the options to the `ibstat` command and their purposes:

Option	Purpose
`-d`	Sets the debug level. Can be used several times to increase the debug level.
`-e`	Displays send and receive errors.
`-h`	Provides help.
`-l`	Lists all InfiniBand devices.
`-s`	Provides short output.
`-p`	Shows port list.
`-V`	Displays the version information.

Example

The following example shows how to display the basic status from the local InfiniBand driver with the `ibstat` command.

```
# ibstat
CA 'mlx4_0'
  CA type: MT26428
  Number of ports: 2
  Firmware version: 2.5.926
  Hardware version: a0
  Node GUID: 0x0003ba000100c708
  System image GUID: 0x0003ba000100c70b
  Port 1:
      State: Down
      Physical state: Polling
      Rate: 10
      Base lid: 0
      LMC: 0
      SM lid: 0
      Capability mask: 0x02510868
      Port GUID: 0x0003ba000100c709
  Port 2:
      State: Active
      Physical state: LinkUp
      Rate: 40
      Base lid: 58
      LMC: 0
```

```
      SM lid: 58
      Capability mask: 0x0251086a
      Port GUID: 0x0003ba000100c70a
#
```

Related Information

- `ibstat` man page
- "`ibstatus` Command" on page 155

`ibstatus` Command

Queries basic status of InfiniBand devices. Issued on the Linux InfiniBand host.

Syntax

`ibstatus [-h][`*devname*`[:`*IB_port*`]]...`

where:

- *devname* is the InfiniBand device name.
- *IB_port* is the port number of the InfiniBand device.

Description

This InfiniBand software command displays basic information retrieved from the local InfiniBand driver. Output of the command includes:

- `LID`
- `SMLID`
- `port logical state`
- `link width`
- `port physical state`

Note – This command is an alternative to the `ibstat` command.

Options

The `-h` option provides help.

Example

The following example shows how to display the basic status from the local InfiniBand driver with the `ibstatus` command.

```
# ibstatus
Infiniband device 'mlx4_0' port 1 status:
  default gid:     fe80:0000:0000:0000:0003:ba00:0100:c709
  base lid:        0x0
  sm lid:          0x0
  state:           1: DOWN
  phys state:      2: Polling
  rate:            10 Gb/sec (4X)
Infiniband device 'mlx4_0' port 2 status:
  default gid:     fe80:0000:0000:0000:0003:ba00:0100:c70a
  base lid:        0x3a
  sm lid:          0x3a
  state:           4: ACTIVE
  phys state:      5: LinkUp
  rate:            40 Gb/sec (4X QDR)
#
```

Related Information

- `ibstatus` man page
- "`ibstat` Command" on page 153

`ibswitches` Command

Displays InfiniBand switch node in the topology. Issued on the Linux InfiniBand host.

Syntax

`ibswitches [-h][`*topology*`|-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]`

where:

- *topology* is the topology file.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.

Description

This InfiniBand software command is a script the discovers the InfiniBand fabric topology or uses an existing topology file to extract the switch nodes.

Options

The following table describes the options to the `ibswitches` command and their purposes:

Option	Purpose
`-h`	Provides help.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.
`-t`	Overrides the default timeout.

Example

The following example shows how to display the switch GUIDs with the `ibswitches` command.

```
# ibswitches
Switch  : 0x0021283a83ae11d2 ports 36 "Sun DCS 648 shmm1500 LC slot 1 switch 3"
base port 0 lid 35 lmc 0
Switch  : 0x0021283a83b112b2 ports 36 "Sun DCS 648 shmm1500 LC slot 2 switch 1"
base port 0 lid 17 lmc 0
Switch  : 0x0021283a83b112a2 ports 36 "Sun DCS 648 shmm1500 LC slot 2 switch 0"
base port 0 lid 4 lmc 0
Switch  : 0x0021283a83b112d2 ports 36 "Sun DCS 648 shmm1500 LC slot 2 switch 3"
base port 0 lid 36 lmc 0
Switch  : 0x0021283a841513c2 ports 36 "Sun DCS 648 shmm1500 LC slot 3 switch 2"
base port 0 lid 31 lmc 0
.
.
.
#
```

Note – The output in the example is just a portion of the full output and varies for each InfiniBand topology.

Related Information

- `ibswitches` man page
- "`getbaseguid` Command" on page 22
- "`ibnetdiscover` Command" on page 140
- "`ibnodes` Command" on page 143

`ibsysstat` Command

Displays system status of an InfiniBand address. Issued on the Linux InfiniBand host.

Syntax

`ibsysstat [-d][-e][-G][-h][-s` *smlid*`][-v][-V][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`][-o` *oui*`][-S]` *lid*|*guid* `[`*op*`]`

where:

- *smlid* is the Subnet Manager LID.
- *topology* is the topology file.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *oui* is the OUI number.
- *lid* is the local identifier.
- *guid* is the global unit identifier.
- *op* is the operation to perform on the node:
  - `ping` – Verify connectivity to the server.
  - `host` – Obtain host information from the server.
  - `cpu` – Obtain CPU information from the server.

Description

This InfiniBand software command uses vendor supplied management datagrams to validate connectivity between InfiniBand nodes and return other information about the node. The command is a client-server, in that a remote node is configured as a server, while a local node performs as a client.

The command is first run locally on the server. The command is then run again locally on the client, pointing to the LID of the server. The ping repeats every second, until the Ctrl-C key combination exits on the server.

Options

The following table describes the options to the `ibsysstat` command and their purposes:

Option	Purpose
`-d`	Sets the debug level. Can be used several times to increase the debug level.
`-e`	Displays send and receive errors.
`-G`	Uses the port GUID address.
`-h`	Provides help.
`-s`	Uses *smlid* as the target LID for Subnet Manager/Subnet Administrator queries.
`-v`	Provides verbose output.
`-V`	Displays the version information.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.
`-t`	Overrides the default timeout.
`-o`	Uses specified OUI number to multiplex vendor management datagrams.
`-S`	Starts in server mode.

Example

The following example shows how to retrieve system information about a remote node server from a local node client with the `ibsysstat` command. First configure the remote node server:

```
# ibsysstat -S
<CTRL-C to exit server>
#
```

Then run the command on the local node client:

```
# ibsysstat 1 ping
sysstat ping succeeded
# ibsysstat 1 host
nsn105-100.nsn.sfbay.sun.com
# ibsysstat 1 cpu
cpu 0: model  AMD Opteron(tm) Processor 848 MHZ  2189.680
cpu 1: model  AMD Opteron(tm) Processor 848 MHZ  2189.680
cpu 2: model  AMD Opteron(tm) Processor 848 MHZ  2189.680
cpu 3: model  AMD Opteron(tm) Processor 848 MHZ  2189.680
#
```

Related Information

- `ibsysstat` man page

`ibtracert` Command

Traces the InfiniBand path. Issued on the Linux InfiniBand host.

Syntax

`ibtracert [-d][-D][-G][-h][-m` *mlid*`][-s` *smlid*`][-v][-V][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`][`*lid*`|`*dr_path*`|`*guid* `[`*startlid* `[`*endlid*`]]]`

where:

- *mlid* is the multicast LID.
- *smlid* is the Subnet Manager LID.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *lid* is the local identifier.
- *dr_path* is the directed path.
- *guid* is the global unit identifier.
- *startlid* is the starting LID for a range.
- *endlid* is the ending LID for a range.

Description

This InfiniBand software command uses SMPs to trace the path from a source GID or LID to a destination GID or LID. Each responding hop in the path is displayed. The `-m` option enables multicast path tracing between source and destination nodes.

Options

The following table describes the options to the `ibtracert` command and their purposes:

Option	Purpose
`-d`	Sets the debug level. Can be used several times to increase the debug level.
`-D`	Uses the directed path address. The path is a comma delimited sequence of out ports.
`-G`	Uses the port GUID address.
`-h`	Provides help.
`-m`	Shows the multicast trace of the specified MLID.
`-s`	Uses *smlid* as the target LID for Subnet Manager or Subnet Administrator queries.
`-v`	Provides verbose output.
`-V`	Displays the version information.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.
`-t`	Overrides the default timeout.

Example

The following example shows how to display the path from LID 58 to LID 57 with the `ibtracert` command.

```
# ibtracert 58 57
From ca {0x0003ba000100c708} portnum 2 lid 58-58 "nsn34-39 HCA-1"
[2] -> switch port {0x00bad0cc010016a2}[22] lid 1-1 "Sun DCS 648 shmm1500 LC slot
6 switch 0"
```

```
[1] -> switch port {0x00bad0fc0100f8b2}[7] lid 57-57 "Sun DCS 648 shmm1500 FC
slot 8 switch 1"
To switch {0x00bad0fc0100f8b2} portnum 0 lid 57-57 "Sun DCS 648 shmm1500 FC slot
8 switch 1"
#
```

Related Information

- `ibtracert` man page
- "`ibroute` Command" on page 149

`ibv_devices` Command

Displays the local RDMA device. Issued on the Linux InfiniBand host.

Syntax

`ibv_devices`

Description

This InfiniBand software command lists the local RDMA devices available from userspace.

Example

The following example shows how to display the local InfiniBand device name and GUID with the `ibv_devices` command.

```
# ibv_devices
 device                 node GUID
 ------              ----------------
 mlx4_0              0003ba000100c708
#
```

Related Information

- `ibv_devices` man page
- "`ibv_devinfo` Command" on page 163

`ibv_devinfo` Command

Queries local RDMA devices. Issued on the Linux InfiniBand host.

Syntax

`ibv_devinfo [-d` *device*`][-i` *IB_port*`][-h][-l][-v]`

where:

- *device* is the InfiniBand device.
- *IB_port* is the InfiniBand port.

Description

This InfiniBand software command displays information about the RDMA devices available from userspace.

Note – This command is similar to the `ibv_devices` command, but the output contains more information.

Options

The following table describes the options to the `ibv_devinfo` command and their purposes:

Option	Purpose
`-d`	Uses the InfiniBand device.
`-i`	Uses the InfiniBand port.
`-h`	Provides help.
`-l`	Lists only names of remote direct memory access devices.
`-v`	Provides verbose output.

Example

The following example shows how to display the local InfiniBand device information with the `ibv_devinfo` command.

```
# ibv_devinfo
hca_id:mlx4_0
  fw_ver:          2.5.9266
  node_guid:       0003:ba00:0100:c708
  sys_image_guid:  0003:ba00:0100:c70b
  vendor_id:       0x02c9
  vendor_part_id:  26428
  hw_ver:          0xA0
  board_id:        SUN0150000001
  phys_port_cnt:2
      port:1
          state:        PORT_DOWN (1)
          max_mtu:      2048 (4)
          active_mtu:   2048 (4)
          sm_lid:       0
          port_lid:     0
          port_lmc:     0x00
      port:2
          state:        PORT_ACTIVE (4)
          max_mtu:      2048 (4)
          active_mtu:   2048 (4)
          sm_lid:       58
          port_lid:     58
          port_lmc:     0x00
#
```

Related Information

- `ibv_devinfo` man page
- "`ibv_devices` Command" on page 162

`ibv_rc_pingpong` Command

Conducts a simple InfiniBand Reliable Connection transport test. Issued on the Linux InfiniBand host.

Syntax

```
ibv_rc_pingpong [-p TCP_port][-d device][-i IB_port][-s size][-r depth]
[-n iters][-l level][-e][-h][IP_address]
```

where:

- *TCP_port* is the TCP port.
- *device* is the InfiniBand device.
- *IB_port* is the InfiniBand port.
- *size* is the size of the ping-pong messages.
- *depth* is the number of depth receives to post at one time.
- *iters* is the number of message exchanges.
- *level* is the service level of the queue pair.
- *IP_address* is the IP address of the remote node host.

Description

This InfiniBand software command runs a simple ping-pong test over the InfiniBand fabric through the Reliable Connection transport. The command is dependent upon the Internet Protocol, so the InfiniBand fabric must be configured with Internet Protocol over InfiniBand (IPoIB). The command is a client-server, in that a remote node is configured as a server, while a local node performs as a client.

The command is first run locally on the server. The command is then run again locally on the client, pointing to the IP address of the server.

Note – This command only functions if your InfiniBand fabric is configured with IPoIB.

Options

The following table describes the options to the `ibv_rc_pingpong` command and their purposes:

Option	Purpose
`-p`	Uses the TCP port for initial synchronization.
`-d`	Uses the InfiniBand device.
`-i`	Uses the InfiniBand port.

Option	Purpose
-s	Sets size of ping-pong messages.
-r	Posts depth receives.
-n	Performs *iters* message exchanges.
-l	Sends messages of the service level.
-e	Sleeps while waiting for work completion events.
-h	Provides help.

Example

The following example shows how to run a simple ping-pong test between a local node client and a remote node server using the Reliable Connection transport with the `ibv_rc_pingpong` command. First configure the remote node server:

```
# ibv_rc_pingpong
  local address:  LID 0x0001, QPN 0xb10406, PSN 0x9e941e
  remote address: LID 0x0005, QPN 0x3a0406, PSN 0xa6cf6e
8192000 bytes in 0.03 seconds = 2487.23 Mbit/sec
1000 iters in 0.03 seconds = 26.35 usec/iter
#
```

Note – The output is not displayed until the local node client issues the respective command.

Then run the command on the local node client:

```
# ibv_rc_pingpong 192.168.200.100
  local address:  LID 0x0005, QPN 0x3a0406, PSN 0xa6cf6e
  remote address: LID 0x0001, QPN 0xb10406, PSN 0x9e941e
8192000 bytes in 0.03 seconds = 2506.06 Mbit/sec
1000 iters in 0.03 seconds = 26.15 usec/iter
#
```

Related Information

- `ibv_rc_pingpong` man page
- "`ibv_srq_pingpong` Command" on page 167
- "`ibv_uc_pingpong` Command" on page 169
- "`ibv_ud_pingpong` Command" on page 172

`ibv_srq_pingpong` Command

Conducts a simple InfiniBand Shared Receive Queue test. Issued on the Linux InfiniBand host.

Syntax

`ibv_srq_pingpong [-p` *TCP_port*`][-d` *device*`][-i` *IB_port*`][-s` *size*`][-q` *numqps*`][-r` *depth*`][-n` *iters*`][-l` *level*`][-e][-h][`*IP_address*`]`

where:

- *TCP_port* is the TCP port.
- *device* is the InfiniBand device.
- *IB_port* is the InfiniBand port.
- *size* is the size of the ping-pong messages.
- *numqps* is the number of queue pairs to test.
- *depth* is the number of depth receives to post at one time.
- *iters* is the number of message exchanges.
- *level* is the service level of the queue pair.
- *IP_address* is the IP address of the remote node host.

Description

This InfiniBand software command runs a simple ping-pong test over the InfiniBand fabric through the Reliable Connection transport, using multiple queue pairs and the single Shared Receive Queue. The command is dependent upon the Internet Protocol, so the InfiniBand fabric must be configured with Internet Protocol over InfiniBand (IPoIB). The command is a client-server, in that a remote node is configured as a server, while a local node performs as a client.

The command is first run locally on the server. The command is then run again locally on the client, pointing to the IP address of the server.

Note – This command only functions if your InfiniBand fabric is configured with IPoIB.

Options

The following table describes the options to the `ibv_srq_pingpong` command and their purposes:

Option	Purpose
`-p`	Uses the TCP port for initial synchronization.
`-d`	Uses the InfiniBand device.
`-i`	Uses the InfiniBand port.
`-s`	Sets size of ping-pong messages.
`-q`	Sets the number of queue pairs for testing.
`-r`	Posts depth receives.
`-n`	Performs *iters* message exchanges.
`-l`	Sends messages of the service level.
`-e`	Sleeps while waiting for work completion events.
`-h`	Provides help.

Example

The following example shows how to run a simple Shared Receive Queue test between a local node client and a remote node server using the Reliable Connection transport with the `ibv_srq_pingpong` command. First configure the remote node server:

```
# ibv_srq_pingpong
  local address:  LID 0x0001, QPN 0xa10406, PSN 0xcf5915
  local address:  LID 0x0001, QPN 0xa10407, PSN 0xeba912
  local address:  LID 0x0001, QPN 0xa10408, PSN 0xd036be
  local address:  LID 0x0001, QPN 0xa10409, PSN 0xae2efe
  local address:  LID 0x0001, QPN 0xa1040a, PSN 0x725f70
  .
  .
  .
  remote address: LID 0x0005, QPN 0x2a0412, PSN 0x0bbc53
  remote address: LID 0x0005, QPN 0x2a0413, PSN 0x01ab04
  remote address: LID 0x0005, QPN 0x2a0414, PSN 0x8aed72
  remote address: LID 0x0005, QPN 0x2a0415, PSN 0xdd554e
8192000 bytes in 0.02 seconds = 3974.77 Mbit/sec
1000 iters in 0.02 seconds = 16.49 usec/iter
```

Note – The output is not displayed until the local node client issues the respective command.

Then run the command on the local node client:

```
# ibv_srq_pingpong 192.168.200.100
  local address:  LID 0x0005, QPN 0x2a0406, PSN 0x9b3436
  local address:  LID 0x0005, QPN 0x2a0407, PSN 0x8d28e8
  local address:  LID 0x0005, QPN 0x2a0408, PSN 0x2c5c12
  local address:  LID 0x0005, QPN 0x2a0409, PSN 0x59fef9
  local address:  LID 0x0005, QPN 0x2a040a, PSN 0x62fc18
  .
  .
  .
  remote address: LID 0x0001, QPN 0xa10411, PSN 0x132ed9
  remote address: LID 0x0001, QPN 0xa10412, PSN 0xe0631b
  remote address: LID 0x0001, QPN 0xa10413, PSN 0x4deaf5
  remote address: LID 0x0001, QPN 0xa10414, PSN 0x172045
  remote address: LID 0x0001, QPN 0xa10415, PSN 0x32fe5b
8192000 bytes in 0.01 seconds = 4785.40 Mbit/sec
1000 iters in 0.01 seconds = 13.69 usec/iter
```

Note – The output in the example is just a portion of the full output.

Related Information

- `ibv_srq_pingpong` man page
- "`ibv_rc_pingpong` Command" on page 164
- "`ibv_uc_pingpong` Command" on page 169
- "`ibv_ud_pingpong` Command" on page 172

`ibv_uc_pingpong` Command

Conducts a simple InfiniBand Unreliable Connection transport test. Issued on the Linux InfiniBand host.

Syntax

```
ibv_uc_pingpong [-p TCP_port][-d device][-i IB_port][-s size][-r depth]
[-n iters][-l level][-e][-h][IP_address]
```

where:

- *TCP_port* is the TCP port.
- *device* is the InfiniBand device.
- *IB_port* is the InfiniBand port.
- *size* is the size of the ping-pong messages.
- *depth* is the number of depth receives to post at one time.
- *iters* is the number of message exchanges.
- *level* is the service level of the queue pair.
- *IP_address* is the IP address of the remote node host.

Description

This InfiniBand software command runs a simple ping-pong test over the InfiniBand fabric through the Unreliable Connection transport. The command is dependent upon the Internet Protocol, so the InfiniBand fabric must be configured with Internet Protocol over InfiniBand (IPoIB). The command is a client-server, in that a remote node is configured as a server, while a local node performs as a client.

The command is first run locally on the server. The command is then run again locally on the client, pointing to the IP address of the server.

Note – This command only functions if your InfiniBand fabric is configured with IPoIB.

Options

The following table describes the options to the `ibv_uc_pingpong` command and their purposes:

Option	Purpose
`-p`	Uses the TCP port for initial synchronization.
`-d`	Uses the InfiniBand device.
`-i`	Uses the InfiniBand port.

Option	Purpose
-s	Sets size of ping-pong messages.
-r	Posts depth receives.
-n	Performs *iters* message exchanges.
-l	Sends messages of the service level.
-e	Sleeps while waiting for work completion events.
-h	Provides help.

Example

The following example shows how to run a simple ping-pong test between a local node client and a remote node server using the Unreliable Connection transport with the `ibv_uc_pingpong` command. First configure the remote node server:

```
# ibv_rc_pingpong
  local address:  LID 0x0001, QPN 0xb20406, PSN 0x5723a6
  remote address: LID 0x0005, QPN 0x3b0406, PSN 0xd9e2f4
8192000 bytes in 0.03 seconds = 2519.55 Mbit/sec
1000 iters in 0.03 seconds = 26.01 usec/iter
#
```

Note – The output is not displayed until the local node client issues the respective command.

Then run the command on the local node client:

```
# ibv_uc_pingpong 192.168.200.100
  local address:  LID 0x0005, QPN 0x3b0406, PSN 0xd9e2f4
  remote address: LID 0x0001, QPN 0xb20406, PSN 0x5723a6
8192000 bytes in 0.03 seconds = 2535.54 Mbit/sec
1000 iters in 0.03 seconds = 25.85 usec/iter
#
```

Related Information

- `ibv_uc_pingpong` man page
- "`ibv_rc_pingpong` Command" on page 164
- "`ibv_srq_pingpong` Command" on page 167
- "`ibv_ud_pingpong` Command" on page 172

`ibv_ud_pingpong` Command

Conducts a simple InfiniBand Unreliable Datagram transport test. Issued on the Linux InfiniBand host.

Syntax

`ibv_ud_pingpong [-p` *TCP_port*`][-d` *device*`][-i` *IB_port*`][-s` *size*`][-r` *depth*`][-n` *iters*`][-l` *level*`][-e][-h][`*IP_address*`]`

where:

- *TCP_port* is the TCP port.
- *device* is the InfiniBand device.
- *IB_port* is the InfiniBand port.
- *size* is the size of the ping-pong messages.
- *depth* is the number of depth receives to post at one time.
- *iters* is the number of message exchanges.
- *level* is the service level of the queue pair.
- *IP_address* is the IP address of the remote node host.

Description

This InfiniBand software command runs a simple ping-pong test over the InfiniBand fabric through the Unreliable Datagram transport. The command is dependent upon the Internet Protocol, so the InfiniBand fabric must be configured with Internet Protocol over InfiniBand (IPoIB). The command is a client-server, in that a remote node is configured as a server, while a local node performs as a client.

The command is first run locally on the server. The command is then run again locally on the client, pointing to the IP address of the server.

Note – This command only functions if your InfiniBand fabric is configured with IPoIB.

Options

The following table describes the options to the `ibv_ud_pingpong` command and their purposes:

Option	Purpose
`-p`	Uses the TCP port for initial synchronization.
`-d`	Uses the InfiniBand device.
`-i`	Uses the InfiniBand port.
`-s`	Sets size of ping-pong messages.
`-r`	Posts depth receives.
`-n`	Performs *iters* message exchanges.
`-l`	Sends messages of the service level.
`-e`	Sleeps while waiting for work completion events.
`-h`	Provides help.

Example

The following example shows how to run a simple ping-pong test between a local node client and a remote node server using the Unreliable Datagram transport with the `ibv_ud_pingpong` command. First configure the remote node server:

```
# ibv_ud_pingpong
  local address:  LID 0x0001, QPN 0xb40406, PSN 0x0ac0ba
  remote address: LID 0x0005, QPN 0x3d0406, PSN 0x87b264
4096000 bytes in 0.03 seconds = 1032.97 Mbit/sec
1000 iters in 0.03 seconds = 31.72 usec/iter
#
```

Note – The output is not displayed until the local node client issues the respective command.

Then run the command on the local node client:

```
# ibv_ud_pingpong 192.168.200.100
  local address:  LID 0x0005, QPN 0x3d0406, PSN 0x87b264
  remote address: LID 0x0001, QPN 0xb40406, PSN 0x0ac0ba
```

```
4096000 bytes in 0.03 seconds = 1041.74 Mbit/sec
1000 iters in 0.03 seconds = 31.45 usec/iter
#
```

Related Information

- `ibv_ud_pingpong` man page
- "`ibv_rc_pingpong` Command" on page 164
- "`ibv_srq_pingpong` Command" on page 167
- "`ibv_uc_pingpong` Command" on page 169

`ofed_info` Command

Displays OFED software version information. Issued on the Linux InfiniBand host.

Syntax

```
ofed_info
```

Description

This InfiniBand software command displays the version numbers, gits, and commit numbers for the various packages that constitute your installation of the OFED software suite.

Example

The following example shows how to display the OFED version information with the `ofed_info` command.

```
# ofed_info
OFED-1.5-beta1
management:
http://www.openfabrics.org/downloads/management/daily/libibumad-1.3.2_20090831
_ce6f8dd.tar.gz
http://www.openfabrics.org/downloads/management/daily/libibmad-1.3.2_20090831_
ce6f8dd.tar.gz
http://www.openfabrics.org/downloads/management/daily/opensm-3.3.2_20090831_ce
6f8dd.tar.gz
```

```
http://www.openfabrics.org/downloads/management/daily/infiniband-diags-1.5.2_2
0090831_ce6f8dd.tar.gz
libibverbs:
http://www.openfabrics.org/downloads/libibverbs/libibverbs-1.1.2-0.1.gb00dc7d.
tar.gz
libmthca:
.
.
.
ibsim:
git://git.openfabrics.org/ofed_1_4/ibsim.git ofed_1_4
commit a76132ae36dde8302552d896e35bd29608ac9524
ofa_kernel-1.5:
Git:
git://git.openfabrics.org/ofed_1_5/linux-2.6.git ofed_kernel_1_5
commit ec80ecd59d1ad658737c7cf98c46c8729de6092d
# MPI
mvapich-1.1.0-3390.src.rpm
mvapich2-1.2p1-1.src.rpm
openmpi-1.3.2-1.src.rpm
mpitests-3.1-891.src.rpm
ib_bonding:
http://www.openfabrics.org/~monis/ofed_1_4/ib-bonding-0.9.0-40.src.rpm
rds-tools:
http://www.openfabrics.org/~vlad/ofed_1_4/rds-tools/rds-tools-1.4-1.src.rpm
rnfs-utils:
http://www.openfabrics.org/~swise/ofed_1_5/rnfs-utils/rnfs-utils-1.1.5-7.OFED.
src.rpm
#
```

Note – The output in the example is just a portion of the full output.

`opensm` Command

Runs the InfiniBand Subnet Manager and Subnet Administrator. Issued on the Linux InfiniBand host.

Syntax

`opensm [-F` *filename*`][-c` *filename*`][-g` *guid*`][-l` *lmc*`][-p` *priority*`][-smkey` *SMKey*`][-r][-R` *engine*`][-A][-z][-M` *filename*`][-U` *filename*`][-S` *filename*`][-a` *path*`][-u` *path*`][-X` *path*`][-m` *path*`][-o][-s` *interval*`][-t` *timeout*`][-maxsmps`

number][-console[off|local|socket|loopback]][-console-port *port*] [-i *filename*][-f *path*][-L *size*][-e][-P *filename*][-N][-Q[-Y *filename*]] [-y][-B][-I][-v][-V][-D *flags*][-d*opt*][-h]

where:

- *filename* is the path and file for the respective option.
- *guid* is the global unit identifier.
- *lmc* is the LMC.
- *priority* is the priority of the instance, `0` (low)–`15` (high).
- *SMKey* is the Subnet Manager encryption key (64 bits).
- *engine* is the routing algorithm.
- *path* is the path to a file used by the option.
- *interval* is the time in seconds between fabric sweeps.
- *timeout* is the timeout in milliseconds.
- *number* is the number of SMP management datagrams.
- *port* is the alternate Telnet port.
- *size* is the maximum size of the log file, in MB.
- *flag* is the log verbosity flag.
- *opt* is the debug option.

Description

This InfiniBand software command initiates an instance of the OpenSM Subnet Manager and Subnet Administrator, which is required to initialize the InfiniBand hardware. OpenSM can be started in a default configuration with the `opensm` command. Options to the `opensm` command can configure the operation of the Subnet Manager to better fit the needs of your InfiniBand fabric.

OpenSM uses several files for operation and configuration. The default directory is `/etc/opensm`. The following table lists the files and their description.

File Name	Description
`opensm.conf`	Default OpenSM configuration file.
`ib-node-name-map`	Default node name map file.
`partitions.conf`	Default partition configuration file.
`qos-policy.conf`	Default Quality of Service policy configuration file.
`prefix-routes.conf`	Default prefix routes file.

Options

The following table describes the options to the `opensm` command and their purposes:

Option	Purpose
`-F`	Uses the OpenSM configuration *filename*. The file `/etc/opensm/opensm.conf` is used by default.
`-c`	Creates the OpenSM configuration file and exit.
`-g`	Binds OpenSM to the local port of *guid* value. If no value for *guid* is provided, OpenSM will display available GUIDs, and wait for user input.
`-l`	Specifies the InfiniBand fabric's LMC value. The value of *lmc* must be within the range of 0 to 7. *lmc* values greater than 0 permit multiple paths between ports and are used only if the topology has multiple paths between ports. The number of LIDs assigned to each port is 2^{lmc}. By default, *lmc* is 0, so there is only one path between any two ports.
`-p`	Sets the priority of the Subnet Manager instance. The master Subnet Manager and handover conditions are based upon the priority and GUID of the Subnet Managers.
`-smkey`	Specifies the Subnet Manager's 64-bit encryption key.
`-r`	Reassigns LIDs to all end nodes. By default, OpenSM attempts to preserve existing LID assignments while resolving multiple use of the same LID. **Note -** Use of the `-r` option might disrupt fabric traffic.
`-R`	Selects the routing algorithm to use. Multiple algorithms can be specified in the order of use upon failure. The list is comma delimited. Algorithms supported are: • `minhop` – Min Hop. This default algorithm routes through the minimum number of hops with an optimized path. • `updn` – Up Down. Like `minhop`, but constrained with ranking rules. This algorithm is used if the fabric is not a pure fat-tree, where a deadlock might occur, due to a loop in the fabric. • `ftree` – Fat-Tree. This algorithm optimizes routing for a congestion-free shift communication pattern. The algorithm is for symmetrical or almost symmetrical fat-tree fabrics. Fat-tree routing is also constrained to ranking rules. • `lash` – LASH. This algorithm uses InfiniBand virtual layers to provide deadlock-free shortest-path routing while distributing the path between the layers. An advantage of the LASH algorithm is that it avoids use of a potentially congested root node. • `dor` – DOR. Based upon the Min Hop algorithm, but avoids port equalization except for redundant links between the same two switches. The algorithm provides deadlock-free routes for hypercubes and meshes, when the fabric is cabled as a hypercube or mesh.
`-A`	Enables the unicast routing cache, which prevents routing recalculations in the event that there was no topology change detected during a sweep or if recalculation is not really necessary. For example, when a host is rebooted.
`-z`	Forces the routing engine to make connectivity between root switches for InfiniBand architecture compliance. Use of the `-z` option might permit the occurrence of a deadlock.

Option	Purpose
-M	Specifies the name of the LID matrix dump file from where the Min Hop tables are loaded.
-U	Specifies the name of the LFT file from where switch forwarding tables are loaded.
-S	Specifies the name of the Subnet Administrator DB dump file from where the Subnet Administrator database is loaded.
-a	Sets the root nodes for the Up Down or Fat-Tree algorithm to the GUIDs in the provided file. One per line.
-u	Sets the compute nodes for the Fat-Tree algorithm to the GUIDs in the provided file. One GUID per line.
-X	Sets the order port GUIDs are routed for the Min Hop and Up Down algorithms to the GUIDs in the provided file. One per line.
-m	Sets a mapping of the IDs to the node GUIDs used by the Up Down algorithm in the provided file. File format is one GUID and ID per line.
-o	OpenSM configures the fabric once and exits. Port remains `Active`.
-s	Specifies the amount of time, in seconds, between fabric sweeps. Default is 10 seconds. An *interval* of `0` disables sweeps.
-t	Overrides the default timeout.
-maxsmps	Specifies the maximum number of outstanding VL15 SMP management datagrams at any one time. Default is 4. A *number* of `0` permits unlimited SMP management datagrams.
-console	Sets the state of the OpenSM console. States are the following: • `off` – This is the default. • `local` • `socket` • `loopback` **Note -** The `socket` and `loopback` states are only available if OpenSM was built with the `--enable-console-socket` option.
-console-port	Specifies an alternative Telnet port for the `socket` state.
-i	Defines a set of node GUID and port pairs to be ignored by the link load equalization algorithm. Pairs are provided in the *filename*.
-f	Instructs OpenSM where to send the log file. Default is `/var/log/opensm.log`. To direct the log to standard output, set *path* to `stdout`.
-L	Sets the maximum *size* of the log file in MB. The log file is truncated when reaching this limit, so the newest entries are lost.
-e	Deletes the log file.
-P	Defines the optional partition configuration *filename*. Default is `/etc/opensm/partitions.conf`.
-N	Disables partition enforcement on switch external ports.
-Q	Enables Quality of Service setup.

Option	Purpose
-Y	Specifies the optional Quality of Service policy *filename*. Default is /etc/opensm/qos-policy.conf. The policy file contains a list of configuration parameters. The parameters have the following format: qos*typeparameter*. Where *type* is one of the following: • nothing – The parameter affects the entire fabric. • _ca – The parameters are set for the channel adapters. • _rtr – The parameters are set for the routers. • _sw*n* – The paramters are set for switch port *n*. • _swe – The parameters are set for switch external ports. Where *parameter* is one of the following: • _max_vls *value* – The maximum number of virtual lanes in the fabric. • _high_limit *value* – The limit of the High Priority component of the virtual lane arbitration table (IBA 7.6.9). • _vlarb_low *data* – The low priority virtual lane arbitration table template. *data* is 15 pairs of virtual lane and weight (*x*:*y*), separated by commas. • _vlarb_high *data* – The high priority virtual lane arbitration table template. *data* is 15 pairs of virtual lane and weight (*x*:*y*), separated by commas. • _sl2vl *data* – The SL2VL mapping table (IBA 7.6.6) template. *data* is a comma delimited sequence of virtual lanes corresponding to service levels 0 through 15.
-y	Prevents OpenSM from exiting upon fatal initialization issues. For example, duplicate GUIDS or 12x links with poorly configured lane reversal.
-B	Runs OpenSM in the background as a daemon.
-I	Starts OpenSM in an inactive state.
-v	Increases the log verbosity.
-V	Sets log verbosity to maximum. The same as -D 0xFF.

Option	Purpose
`-D`	Sets the log verbosity according to the following bit *flags*: • `0x01` – `ERROR` (error messages) • `0x02` – `INFO` (basic messages, low volume) • `0x04` – `VERBOSE` (moderate volume) • `0x08` – `DEBUG` (diagnostic, high volume) • `0x10` – `FUNCS` (function entry and exit, very high volume) • `0x20` – `FRAMES` (dumps all SMP and GMP frames) • `0x40` – `ROUTING` (dumps FDB routing information) The default is `ERROR` + `INFO`, or `0x03`. The option `-D 0` turns off all log messaging. **Note -** High verbosity levels might require increasing the transaction timeout with the `-t` *timeout* option.
`-d`	Specifies debug and an option to the debug. The value of `-d`*opt* is as follows: • `-d0` – Ignores other Subnet Manager nodes. • `-d1` – Forces single threaded dispatching. • `-d2` – Forces log flushing after each log message. • `-d3` – Disables multicast support
`-h`	Displays usage information.

Example

The following example shows how to initiate the Subnet Manager using the Fat-Tree routing algorithm with the `opensm` command.

```
# opensm -R ftree -A -a /etc/opensm/guid.txt -s 120
```

Note – The `guid.txt` file is a listing of fabric card root node GUIDs used by the Fat-Tree algorithm. See the *Sun Datacenter InfiniBand Switch 648 Installation Guide* for more information.

Related Information

- `opensm` man page
- "`osmtest` Command" on page 185
- "`sminfo` Command" on page 194

opensmd Daemon

Starts and stops the Subnet Manager. Issued on the Linux InfiniBand host.

Syntax

```
/etc/init.d/opensmd start|stop|status
```

Description

Like the "opensm Command" on page 175, the opensmd daemon initiates an instance of the OpenSM Subnet Manager. The daemon reads the /etc/opensm/opensm.conf configuration file upon startup, which it uses to configure the Subnet Manager. The opensmd daemon is more user-friendly because it is not necessary to provide elaborate command-line options and arguments.

The following table describes the more common parameters provided to the opensmd daemon in the /etc/opensm/opensm.conf file and their default values.

Parameter	Description	Default Value
guid	Port GUID on which the Subnet Manager is running.	0x0000000000000000
m_key	M_Key value sent to all ports qualifying all Set(PortInfo).	0x0000000000000000
m_key_lease_period	Lease period used for the M_Key in seconds.	0
sm_key	SM_Key value of the Subnet Manager used for authentication.	0x0000000000000001
sa_key	SM_Key value used to qualify received Subnet Administrator queries as trusted.	0x0000000000000001
subnet_prefix	Subnet prefix.	0xfe80000000000000
lmc	LMC value used on this subnet.	0
lmc_esp0	lmc_esp0 determines whether LMC value is used for enhanced switch port 0. If TRUE, LMC value for subnet is used for ESP0. Otherwise, LMC value for ESP0s is 0.	FALSE

Parameter	Description	Default Value
packet_life_time	Maximum time a packet can live in a switch. Actual time is 4.096 usec * 2^packet_life_time. A value of 0x14 disables the mechanism.	0x12
vl_stall_count	Number of sequential packets dropped that cause the port to enter the VLStalled state. Do not set to 0.	0x07
leaf_vl_stall_count	Similar to vl_stall_count, this value is for switch ports driving a CA or router port. Do not set to 0.	0x07
head_of_queue_lifetime	Maximum time a packet can wait at the head of the transmission queue. Actual time is 4.096 usec * 2^head_of_queue_lifetime. A value of 0x14 disables the mechanism.	0x12
leaf_head_of_queue_lifetime	Maximum time a packet can wait at the head of the queue on a switch port connected to a CA or router port. Actual time is 4.096 usec * 2^leaf_head_of_queue_lifetime.	0x10
max_op_vls	Maximum operational virtual lanes.	5
force_link_speed	Force PortInfo:LinkSpeedEnabled on switch ports. Values are: • 0 – Make no change. • 1 – 2.5 Gbps • 3 – 2.5 Gbps or 5.0 Gbps • 5 – 2.5 Gbps or 10.0 Gbps • 7 – 2.5 Gbps or 5.0 Gbps or 10.0 Gbps • 2, 4, 6, 8–14 Reserved Default is 15: set to PortInfo:LinkSpeedSupported	15
subnet_timeout	Subnet timeout for all the ports. Actual timeout is 4.096 usec * 2^subnet_timeout.	18
local_phy_errors_threshold	Threshold of local PHY errors for sending trap 129.	0x08
overrun_errors_threshold	Threshold of credit overrun errors for sending trap 130.	0x08
partition_config_file	Partition configuration file.	/etc/opensm/partitions.conf

Parameter	Description	Default Value
no_partition_enforcement	Disable partition enforcement by switches.	FALSE
sweep_interval	Number of seconds between subnet sweeps (0 disables).	10
reassign_lids	If TRUE, reassign all LIDs.	FALSE
force_heavy_sweep	If TRUE, force heavy sweeps.	FALSE
sweep_on_trap	If TRUE, heavy sweep on trap. Successive identical traps (>10) are suppressed.	TRUE
port_profile_switch_nodes	If TRUE, count switches as link subscriptions.	FALSE
routing_engine	Routing engine: minhop, updn, file, ftree, lash, or dor. Multiple routing engines are comma delimited.	ftree
sm_priority	Subnet Manager priority determines the master. Range is 0 (lowest priority) to 15 (highest).	0
ignore_other_sm	If TRUE, ignore other Subnet Managers.	FALSE
sminfo_polling_timeout	Timeout in msec between consecutive polls of active master Subnet Manager.	1000
polling_retry_number	Number of failing polls of remote Subnet Manager to declare it dead.	3
honor_guid2lid_file	If TRUE, honor the guid2lid file when coming out of standby state.	FALSE
max_wire_smps	Maximum number of SMPs sent in parallel.	4
transaction_timeout	Maximum time in msec allowed for a transaction to complete.	200
max_msg_fifo_timeout	Maximum time in msec a message can stay in the incoming message queue. If two or more messages exceed this time in the queue, any Subnet Administrator request is returned with a BUSY status.	10000
single_thread	Use a single thread for handling Subnet Administrator queries.	FALSE
daemon	Daemon mode.	FALSE
sm_inactive	Deactivate the Subnet Manager.	FALSE
babbling_port_policy	Babbling Port Policy.	FALSE
log_flags	Log flags used.	0x03

Parameter	Description	Default Value
force_log_flush	Force flush of the log file after each log message.	FALSE
log_file	Log file to be used.	/var/log/opensm.log
log_max_size	Maximum size of the log file in MB. If overrun, log is restarted.	0
accum_log_file	If TRUE, accumulates the log over multiple OpenSM sessions.	TRUE
dump_files_dir	Directory for the OpenSM dump files.	/var/log/
no_clients_rereg	If TRUE, disables client reregistration.	FALSE
disable_multicast	If TRUE, multicast support and multicast routing is disabled.	FALSE
exit_on_fatal	If TRUE, opensm exits on fatal initialization issues.	TRUE
console	Console is either off or local.	off
console_port	Telnet port for console (default 10000).	10000
prefix_routes_file	Prefix routes file name.	/etc/opensm/prefix-routes.conf

Options

The following table describes the options to the opensmd daemon and their purposes:

Option	Purpose
start	Initiates the OpenSM Subnet Manager using the /etc/opensm/opensm.conf file to configure the Subnet Manager.
stop	Terminates the OpenSM Subnet Manager.
status	Provides the status of the Subnet Manager.

Example

The following example shows how to start the Subnet Manager with the `opensmd` daemon.

```
# /etc/init.d/opensmd start
Starting IB Subnet Manager.                                [  OK  ]
#
```

Related Information

- "`opensm` Command" on page 175

`osmtest` Command

Runs the InfiniBand Subnet Manager and administration test program. Issued on the Linux InfiniBand host.

Syntax

`osmtest [-f c|a|v|s|e|f|m|q|t][-w` *time*`][-d`*opt*`][-m` *lid*`][-g` *guid*`][-p][-i` *filename*`][-s`*opt*`][-M`*opt*`][-t` *timeout*`][-l` *path*`][-v][-vf` *flags*`][-h]`

where:

- *time* is the wait time in seconds.
- *opt* is a numeric option.
- *lid* is the local identifier.
- *guid* is the global unit identifier.
- *filename* is the path and name of the inventory file.
- *timeout* is the timeout in milliseconds.
- *path* is the path and name of the log file.
- *flags* is the log verbosity.

Description

The `osmtest` command creates an inventory file of all available nodes, ports, and path records, and compares the file with a previously created file. After the first run of OpenSM, you can create the inventory file with the `-f c` option. At a later time, you can run `osmtest -f v` to identify where there have been changes.

The `osmtest` command conducts the following tests:

- Multicast compliancy test
- Event forwarding test
- Service record registration test
- RMPP stress test
- Small Subnet Administrator queries test

Options

The following table describes the options to the `osmtest` command and their purposes:

Option	Purpose
`-f`	Directs `osmtest` to run a specific test flow: • `c` – Creates an inventory file with all nodes, ports, and paths. • `a` – Runs all validation tests. • `v` – Validates only the given inventory file. • `s` – Runs service registration, deregistration, and lease tests. • `e` – Runs the event forwarding test. • `f` – Floods the Subnet Administrator with queries according to the stress mode. • `m` – Multicast flow. • `q` – Quality of Service information, dumps the VLarb and SL2VL tables. • `t` – Runs trap 64/65 flow.
`-w`	Specifies the wait time for the trap 64/65 flow.
`-d`	Specifies debug and an option to the debug. The value of `-d`*opt* is as follows: • `-d0` – Ignore other Subnet Manager nodes. • `-d1` – Force single threaded dispatching. • `-d2` – Force log flushing after each log message. • `-d3` – Disable multicast support
`-m`	Specifies the maximum LID to be searched during the inventory file build.
`-g`	Binds OpenSM to the local port of *guid* value. If no value for *guid* is provided, OpenSM will display available GUIDs and wait for user input.

Option	Purpose
-p	Displays a list of available port GUIDs to which osmtest could bind.
-i	Specifies the *filename* of the inventory file. The default is osmtest.dat.
-s	Runs the specified stress test instead of the normal test suite. The value of -s*opt* is as follows: • -s1 – Single management datagram response Subnet Administrator queries. • -s2 – Multimanagement datagram RMPP response Subnet Administrator queries. • -s3 – Multimanagement datagram RMPP path record Subnet Administrator queries.
-M	Specifies length of multicast test. The value of -M*opt* is as follows: • -M1 – Short multicast flow, single mode. • -M2 – Short multicast flow, multiple mode. • -M3 – Long multicast flow, single mode. • -M4 – Long multicast flow, multiple mode. In single mode, osmtest is tested alone. In multiple mode, osmtest is run with other applications using multicast with OpenSM.
-t	Overrides the default timeout.
-l	Sets the log file to be *path*. The default action is to direct the log to standard output.
-v	Increases the log verbosity.
-V	Sets the log verbosity to maximum. The same as -vf 0xFF
-vf	Sets the log verbosity according to the following bit *flags*: • 0x01 – ERROR (error messages) • 0x02 – INFO (basic messages, low volume) • 0x04 – VERBOSE (moderate volume) • 0x08 – DEBUG (diagnostic, high volume) • 0x10 – FUNCS (function entry and exit, very high volume) • 0x20 – FRAMES (dumps all SMP and GMP frames) • 0x40 – ROUTING (dumps FDB routing information) The default is ERROR + INFO, or 0x03. The option -vf 0 turns off all log messaging. **Note -** High verbosity levels might require increasing the transaction timeout with the -t *timeout* option.
-h	Displays usage information.

Example

The following example shows how to run all validation tests with the osmtest command.

```
# osmtest -f a
Command Line Arguments
Done with args
```

```
  Flow = All Validations
Oct 29 17:14:13 716178 [BADFCA10] 0x7f -> Setting log level to: 0x03
Oct 29 17:14:13 716533 [BADFCA10] 0x02 -> osm_vendor_init: 1000 pending umads
specified
Oct 29 17:14:13 744266 [BADFCA10] 0x02 -> osm_vendor_bind: Binding to port
0x3ba000100c70a
Oct 29 17:14:13 778486 [BADFCA10] 0x02 -> osmtest_validate_sa_class_port_info:
------------------------------
SA Class Port Info:
 base_ver:1
 class_ver:2
 cap_mask:0x2602
 cap_mask2:0x0
 resp_time_val:0x10
------------------------------
Oct 29 17:14:13 778557 [BADFCA10] 0x01 -> osmtest_create_db: ERR 0130: Unable to
open inventory file (osmtest.dat)
Oct 29 17:14:13 778585 [BADFCA10] 0x01 -> osmtest_run: ERR 0145: Database
creation failed (IB_ERROR)
OSMTEST: TEST "All Validations" FAIL
#
```

Note – The output in the example is just a portion of the full output.

Related Information

- `osmtest` man page

`perfquery` Command

Queries InfiniBand port counters. Issued on the Linux InfiniBand host.

Syntax

`perfquery [-d][-e][-G][-h][-a][-l][-r][-R][-v][-V][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`][`*lid*`|`*guid* `[[`*port*`][`*reset_mask*`]]]`

where:

- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.

- *lid* is the local identifier.
- *guid* is the global unit identifier.
- *port* is the port being queried.
- *reset_mask* is the two-byte mask.

Description

This InfiniBand software command uses the performance management GMPs to acquire the `PortCounters` or `PortExtendedCounters` from the Performance Manager agent at the node or port specified.

Note – The data values retrieved from `PortCounters` and `PortExtendedCounters` are represented as octets divided by 4.

Note – Providing a *port* value of 255 ensures that the operation is performed on all ports.

Options

The following table describes the options to the `perfquery` command and their purposes:

Option	Purpose
`-d`	Sets the debug level. Can be used several times to increase the debug level.
`-e`	Shows extended port counters.
`-a`	Shows aggregate counters for all ports.
`-G`	Uses the port GUID address.
`-h`	Provides help.
`-l`	Loops through all ports.
`-r`	Resets the counters after reading.
`-R`	Resets counters only.
`-v`	Provides verbose output.
`-V`	Displays the version information.
`-C`	Uses the specified channel adapter name.

Option	Purpose
-P	Uses the specified channel adapter port.
-t	Overrides the default timeout.

Example

The following example shows how to display all port counters for LID 4, port 24 with the `perfquery` command.

```
# perfquery 4 24
# Port counters: Lid 4 port 24
PortSelect:......................24
CounterSelect:...................0x1b01
SymbolErrors:....................0
LinkRecovers:....................0
LinkDowned:......................0
RcvErrors:.......................0
RcvRemotePhysErrors:.............0
RcvSwRelayErrors:................0
XmtDiscards:.....................0
XmtConstraintErrors:.............0
RcvConstraintErrors:.............0
CounterSelect2:..................0x00
LinkIntegrityErrors:.............0
ExcBufOverrunErrors:.............0
VL15Dropped:.....................0
XmtData:.........................0
RcvData:.........................0
XmtPkts:.........................0
RcvPkts:.........................0
XmtWait:.........................0
#
```

Related Information

- `perfquery` man page
- "`ibcheckerrors` Command" on page 111
- "`ibdatacounters` Command" on page 127
- "`ibdatacounts` Command" on page 128

`saquery` Command

Queries InfiniBand fabric administration attributes. Issued on the Linux InfiniBand host.

Syntax

`saquery [-h][-d][-p][-N][-D][-S][-I][-L][-l][-G][-O][-U][-c][-s][-g][-m][-x][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`][--src-to-dst` *source*`:`*destination*`][--sgid-to-dgid` *source-destination*`][`*name*`|`*lid*`|`*guid*`]`

where:

- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *lid* is the local identifier.
- *guid* is the global unit identifier.
- *name* is the query name described in the following table:

Query Names	Alias for *name*	Format
ClassPortInfo	CPI	
NodeRecord	NR	
PortInfoRecord	PIR	
SL2VLTableRecord	SL2VL	[[*lid*]/[*in_port*]/[*out_port*]]
PKeyTableRecord	PKTR	[[*lid*]/[*port*]/[*block*]]
VLArbitrationTableRecord	VLAR	[[*lid*]/[*port*]/[*block*]]
InformInfoRecord	IIR	
LinkRecord	LR	[[*from_lid*]/[*from_port*]] [[*to_lid*]/[*to_port*]]
ServiceRecord	SR	
PathRecord	PR	
MCMemberRecord	MCMR	
LFTRecord	LFTR	[[*lid*]/[*block*]]
MFTRecord	MFTR	[[*mlid*]/[*position*]/[*block*]]

Description

This InfiniBand software command performs the selected Subnet Administrator query. Node records are queried by default.

Options

The following table describes the options to the `saquery` command and their purposes:

Option	Purpose
`-h`	Provides help.
`-d`	Sets the debug level. Can be used several times to increase the debug level.
`-p`	Displays the PathRecord information.
`-N`	Displays the NodeRecord information.
`-D`	Displays the NodeDescriptions of channel adapters only.
`-S`	Displays ServiceRecord information.
`-I`	Displays InformInfoRecord information.
`-L`	Returns the LIDs of the specified name.
`-l`	Returns the unique LID of the specified name.
`-G`	Returns the GUIDs of the specified name.
`-O`	Returns the name of the specified LID.
`-U`	Returns the name of the specified GUID.
`-G`	Uses the port GUID address.
`-c`	Displays the Subnet Administrator class port information.
`-s`	Returns the PortInforRecords with the `isSM` or `isSMdisabled` capability mask bit enabled.
`-g`	Displays multicast group information.
`-m`	Displays multicast member information. If a group is specified, provides only the GUID and node description for each entry.
`-x`	Displays LinkRecord information.
`--src-to-dst`	Displays a PathRecord for *source*:*destination*, where *source* and *destination* are either node names or LIDs.
`--sgid-to-dgid`	Displays a PathRecord for *source*-*destination*, where *source* and *destination* are GIDs in an IPv6 format acceptable to inet_pton.

Option	Purpose
-C	Uses the specified channel adapter name.
-P	Uses the specified channel adapter port.
-t	Overrides the default timeout.

Example

The following example shows how to produce a node record dump of every LID in the InfiniBand fabric with the `saquery` command.

```
# saquery
NodeRecord dump:
     lid.....................0x3A
     reserved................0x0
     base_version............0x1
     class_version...........0x1
     node_type...............Channel Adapter
     num_ports...............0x2
     sys_guid................0x0003ba000100c70b
     node_guid...............0x0003ba000100c708
     port_guid...............0x0003ba000100c70a
     partition_cap...........0x80
     device_id...............0x673C
     revision................0xA0
     port_num................0x2
     vendor_id...............0x2C9
     NodeDescription.........nsn34-39 HCA-1
NodeRecord dump:
     lid.....................0x2
     reserved................0x0
.
.
.
#
```

Note – The output in the example is just a portion of the full output.

Related Information

- `saquery` man page
- "`getboardstat` Command" on page 23

sminfo Command

Queries the InfiniBand SMInfo attribute. Issued on the Linux InfiniBand host.

Syntax

sminfo [-d][-e] -s *state* -p *priority* -a *activity* [-D][-G][-h][-V][-C *ca_name*][-P *ca_port*][-t *timeout*] *smlid*|*smdr_path*

where:

- *state* is the state for the Subnet Manager.
- *priority* is the priority.
- *activity* is the activity count.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *smlid* is the Subnet Manager local identifier.
- *smdr_path* is the directed path for the Subnet Manager.

Description

This InfiniBand software command conducts a query of the Subnet Manager and outputs the information in a human readable format. The target Subnet Manager is identified in the local port information, or it is specified by the *smlid* or *smdr_path*.

Note – Using the sminfo command for other than simple queries might fault the target Subnet Manager.

Options

The following table describes the options to the sminfo command and their purposes:

Option	Purpose
-d	Sets the debug level. Can be used several times to increase the debug level.
-D	Uses the directed path address. The path is a comma delimited sequence of out ports.

Option	Purpose
-e	Displays send and receive errors.
-s	Sets the Subnet Manager state: • 0 – Not active. • 1 – Discovering. • 2 – Standby. • 3 – Master.
-p	Sets the priority, (0–15).
-a	Sets the activity count.
-G	Uses the port GUID address.
-h	Provides help.
-V	Displays the version information.
-C	Uses the specified channel adapter name.
-P	Uses the specified channel adapter port.
-t	Overrides the default timeout.

Example

The following example shows how to display the SMInfo with the `sminfo` command.

```
# sminfo
sminfo: sm lid 58 sm guid 0x3ba000100c70a, activity count 46133 priority 0 state
3 SMINFO_MASTER
#
```

Related Information

- `sminfo` man page
- "`smpdump` Command" on page 195

`smpdump` Command

Dumps the InfiniBand fabric management attributes. Issued on the Linux InfiniBand host.

Syntax

`smpdump [-s][-D][-h][-V][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]`
lid|*dr_path* *attr* `[`*mod*`]`

where:

- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *lid* is the local identifier.
- *dr_path* is the directed path.
- *attr* is the InfiniBand architecture attribute ID for Subnet Manager attribute.
- *mod* is the InfiniBand architecture modifier for Subnet Manager attribute.

Description

This InfiniBand software command is a general purpose SMP utility that returns Subnet Manager attributes from a specified SMA. The output is in hexadecimal.

Options

The following table describes the options to the `smpdump` command and their purposes:

Option	Purpose
`-D`	Uses the directed path address. The path is a comma delimited sequence of out ports.
`-e`	Displays send and receive errors.
`-g`	Shows the GID address only.
`-h`	Provides help.
`-V`	Displays the version information.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.
`-t`	Overrides the default timeout.

Example

The following example shows how to display a raw dump of the InfiniBand management attributes for the directed path through ports 0 to 2, with attribute ID 0x19 with the `smpdump` command.

```
# smpdump D 0,2 0x19 1
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 1f03 0302
7452 0047 5040 0408 0805 f240 0000 0000
0000 0000 0088 0055 00ff ffff 0000 0000
SMP status: 0x8000
#
```

Related Information

- `smpdump` man page
- "`smpquery` Command" on page 197

`smpquery` Command

Queries InfiniBand fabric management attributes. Issued on the Linux InfiniBand host.

Syntax

`smpquery [-d][-D][-e][-G][-h][-v][-V][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`] op lid|dr_path|guid [`*params*`]`

where:

- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *op* is the supported operation.
- *lid* is the local identifier.
- *dr_path* is the directed path.
- *guid* is the global unit identifier.
- *params* is the parameter of the operation.

Description

This InfiniBand software command permits a subset of the standard SMP queries, including the following:

- Node information
- Node description
- Switch information
- Port information

Output is in human-readable format.

Supported operations and parameters are as follows:

- `nodeinfo` *addr*
- `nodedesc` *addr*
- `portinfo` *addr* [*portnum*]
- `switchinfo` *addr*
- `pkeys` *addr* [*portnum*]
- `sl2vl` *addr* [*portnum*]
- `vlarb` *addr* [*portnum*]
- `guids` *addr*

where:

- *addr* is the address.
- *portnum* is the port number.

Options

The following table describes the options to the `smpquery` command and their purposes:

Option	Purpose
`-d`	Sets the debug level. Can be used several times to increase the debug level.
`-D`	Uses the directed path address. The path is a comma delimited sequence of out ports.
`-e`	Displays send and receive errors.
`-G`	Uses the port GUID address.
`-h`	Provides help.
`-v`	Provides verbose output.

Option	Purpose
-V	Displays the version information.
-C	Uses the specified channel adapter name.
-P	Uses the specified channel adapter port.
-t	Overrides the default timeout.

Example

The following example shows how to display node information for LID 15 with the smpquery command.

```
# smpquery nodeinfo 15
# Node info: Lid 15
BaseVers:.....................1
ClassVers:....................1
NodeType:.....................Switch
NumPorts:.....................36
SystemGuid:...................0x0003ba2550282543
Guid:.........................0x0021283a83b710b2
PortGuid:.....................0x0021283a83b710b2
PartCap:......................8
DevId:........................0xbd36
Revision:.....................0x000000a0
LocalPort:....................2
VendorId:.....................0x0002c9
#
```

Related Information

- smpquery man page
- "saquery Command" on page 191
- "smpdump Command" on page 195

vendstat Command

Queries vendor specific functions. Issued on the Linux InfiniBand host.

Syntax

`vendstat [-d][-e][-G][-h][-N][-w][-v][-V][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]` *lid*|*guid*

where:

- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *lid* is the local identifier.
- *guid* is the global unit identifier.

Description

This InfiniBand software command uses vendor specific management datagrams to access vendor functionality beyond the InfiniBand specification.

Note – Currently, only Mellanox provides additional functionality.

Options

The following table describes the options to the `vendstat` command and their purposes:

Option	Purpose
`-d`	Sets the debug level. Can be used several times to increase the debug level.
`-e`	Displays send and receive errors.
`-G`	Uses the port GUID address.
`-h`	Provides help.
`-N`	Shows IS3 general information.
`-w`	Shows IS3 port transmit wait counters.
`-v`	Provides verbose output.
`-V`	Displays the version information.
`-C`	Uses the specified channel adapter name.
`-P`	Uses the specified channel adapter port.

Option	Purpose
-t	Overrides the default timeout.

Example

The following example shows how to display IS3 general information for LID 15 with the vendstat command.

```
# vendstat -N 15
hw_dev_rev:  0x01b3
hw_dev_id:   0x01b3
hw_uptime:   0x00041d6b
fw_version:  07.02.44
fw_build_id: 0x2fb8
fw_date:     06/24/2009
fw_psid:     'SUN_M9_LC001'
fw_ini_ver:  0
sw_version:  00.00.07
#
```

Related Information

- vendstat man page
- "saquery Command" on page 191

Index

A

activate command, 3, 44

C

checkfans command, 4
checklinks command, 5
checkpowers command, 7
checkpwrfault command, 8
checkswitches command, 9
checkvoltages command, 10
clearboardstat command, 11
clia commands, 42
command
  clia, 42
    activate, 44
    deactivate, 45
    exit, 46
    fru, 47
    frucontrol, 48
    fruinfo, 49
    getfruledstate, 51
    getipmbstate, 52
    getlanconfig, 54
    help, 57
    ipmc, 58
    localaddress, 60
    poll, 60
    quit, 61
    sel, 62
    sensordata, 63
    session, 65
    setextracted, 66
    setfruledstate, 67
    setipmbstate, 68
    setlanconfig, 69
    setlocked, 70
    shelf, 71
    shelfaddress, 75
    shmstatus, 76
    showunhealthy, 77
    switchover, 78
    terminate, 78
    user, 79
    user add, 81
    user channel, 82
    user delete, 83
    user enable, 84
    user name, 85
    user passwd, 86
    version, 87
  InfiniBand software, 88
    ib_read_bw, 91
    ib_read_lat, 94
    ib_send_bw, 97
    ib_send_lat, 100
    ib_write_bw, 102
    ib_write_lat, 105
    ibaddr, 108
    ibcheckerrors, 110
    ibchecknet, 112
    ibchecknode, 113
    ibcheckport, 115
    ibcheckportstate, 116
    ibcheckportwidth, 118
    ibcheckstate, 119
    ibcheckwidth, 121
    ibclearcounters, 123
    ibclearerrors, 124
    ibdatacounters, 126
    ibdatacounts, 127
    ibdiagnet, 129
    ibdiagpath, 133
    ibhosts, 137
    ibnetdiscover, 139
    ibnodes, 142

ibping, 143
ibportstate, 146
ibroute, 148
ibrouters, 150
ibstat, 152
ibstatus, 154
ibswitches, 155
ibsysstat, 157
ibtracert, 159
ibv_devices, 161
ibv_devinfo, 162
ibv_rc_pingpong, 163
ibv_srq_pingpong, 166
ibv_uc_pingpong, 168
ibv_ud_pingpong, 171
ofed_info, 173
opensm, 174
opensmd, 180
osmtest, 184
perfquery, 187
saquery, 190
sminfo, 193
smpdump, 194
smpquery, 196
vendstat, 198
switch-specific, 1
activate, 3
checkfans, 4
checklinks, 5
checkpowers, 7
checkpwrfault, 8
checkswitches, 9
checkvoltages, 10
clearboardstat, 11
deactivate, 12
disableboard, 13
disablepsu, 14
disablestby, 14
disableswitchport, 15
enableboard, 16
enablehotinsert, 17
enablepsu, 18
enablestby, 19
enableswitchport, 19
env_test, 20
findport, 21
getbaseguid, 22
getboardstat, 23
getfwversion, 25
getportcounters, 26
getportstatus, 27
getpsufwver, 28
getserialnumbers, 29
getsymerr, 30
i2ctest, 31
m9diag, 31
mcmversion, 33
psustatus, 33
resetswitch, 34
selectdebugport, 35
setlinkspeed, 36
showlogs, 37
showpresent, 38
showtemps, 39
showvoltages, 40
command reference, 1

D

deactivate command, 12, 45
disableboard command, 13
disablepsu command, 14
disablestby command, 14
disableswitchport command, 15

E

enableboard command, 16
enablehotinsert command, 17
enablepsu command, 18
enablesm parameters, 180
enablestby command, 19
enableswitchport command, 19
env_test command, 20
exit command, 46

F

findport command, 21
fru command, 47
frucontrol command, 48
fruinfo command, 49

G

getbaseguid command, 22
getboardstat command, 23
getfruledstate command, 51

getfwversion command, 25
getipmbstate command, 52
getlanconfig command, 54
getlanconfig parameters, 56
getportcounters command, 26
getportstatus command, 27
getpsufwver command, 28
getserialnumbers command, 29
getsymerr command, 30

H

help command, 57

I

i2ctest command, 31
ib_read_bw command, 91
ib_read_lat command, 94
ib_send_bw command, 97
ib_send_lat command, 100
ib_write_bw command, 102
ib_write_lat command, 105
ibaddr command, 108
ibcheckerrors command, 110
ibchecknet command, 112
ibchecknode command, 113
ibcheckport command, 115
ibcheckportstate command, 116
ibcheckportwidth command, 118
ibcheckstate command, 119
ibcheckwidth command, 121
ibclearcounters command, 123
ibclearerrors command, 124
ibdatacounters command, 126
ibdatacounts command, 127
ibdiagnet command, 129
  support files, 130
ibdiagpath command, 133
  support files, 134
ibhosts command, 137
ibnetdiscover command, 139
  topology file, 139
ibnodes command, 142
ibping command, 143
ibportstate command, 146
ibroute command, 148
ibrouters command, 150
ibstat command, 152
ibstatus command, 154
ibswitches command, 155
ibsysstat command, 157
ibtracert command, 159
ibv_devices command, 161
ibv_devinfo command, 162
ibv_rc_pingpong command, 163
ibv_srq_pingpong command, 166
ibv_uc_pingpong command, 168
ibv_ud_pingpong command, 171
InfiniBand software commands, 88
ipmc command, 58

L

LAN configuration parameters, 56
localaddress command, 60

M

m9diag command, 31
mcmversion command, 33

O

ofed_info command, 173
opensm command, 174
  options, 176
  support files, 175
opensm parameters, 180
opensmd daemon, 180
osmtest command, 184

P

perfquery command, 187
poll command, 60
psustatus command, 33

Q

quit command, 61

R

resetswitch command, 34

S

saquery command, 190
sel command, 62
selectdebugport command, 35
sensordata command, 63
session command, 65
setextracted command, 66
setfruledstate command, 67
setipmbstate command, 68
setlanconfig command, 69
setlanconfig parameters, 56
setlinkspeed command, 36
setlocked command, 70
shelf command, 71
shelfaddress command, 75
shmstatus command, 76
showlogs command, 37
showpresent command, 38
showtemps command, 39
showunhealthy command, 77
showvoltages command, 40
sminfo command, 193
smpdump command, 194
smpquery command, 196
switchover command, 78
switch-specific commands, 1

T

terminate command, 78

U

user add command, 81
user channel command, 82
user command, 79
user delete command, 83
user enable command, 84
user name command, 85
user passwd command, 86

V

vendstat command, 198

version command, 87