HP BLc Intel® 4X QDR InfiniBand Switch Command Line Interface Reference Guide
for the HP c-Class BladeSystems

Document Revision History	
Revision C, October 2013	
Changes	**Sections Affected**
Updated for Intel branding	All

Contents

1 Overview

This section discusses the usage of the Command Line Interface (CLI) feature for the Intel® BLc 4X QDR InfiniBand Managed Switch.

The CLI lets you perform remote configuration and management tasks, which in many respects mirror the functionality of the Chassis Viewer GUI.

The CLI is accessed via the Out of Band (OOB) management port using Telnet and secure shell (SSH). For a standalone switch, you can telnet to the IP address of the unit. Once connected, the CLI works as any telnet session does.

To access the CLI, a login and password is required. There are two user modes, operator and administrator with the following access privileges:

Operator:

• Read-only access

Administrator:

• Read and write Access
• Reboot access
• Can change operator and administrator passwords
• Can disable user login and passwords. This allows all users admin-level access without the need for a user name or password
• Can view all current user sessions
• Can access all of the commands executed from any open operator session
• Can log out any open user sessions
• Can send messages to the open user sessions

The CLI allows multiple users to be logged in simultaneously. However, some commands will be locked to a user(s) if another user is executing the same command.

Commands and Functional Groups

The list of available commands can be accessed by typing `list`. To keep the list short, the commands are grouped into functional groups, which are:

General:

General commands for user management and CLI configuration.

Chassis:

Provides commands for configuration and management of the chassis.

Network:

Provides commands for configuration and management of the network.

Firmware:

Provides commands for updating the firmware via a File Transfer Protocol (FTP) server or Secure Copy Protocol (SCP) (if using SSH to access the Bridge Module). The switch has the ability to store the location of the firmware files for future upgrades. Additionally, the Firmware functional group includes commands for viewing the current firmware revisions and for changing the boot image.

Subnet Management:

InfiniBand subnet manager configuration and management.

Log:

Provides commands for viewing log files as well as configuring logging parameters.

Key Management:

License key management.

IbSwitchInfo:

Provides commands for displaying InfiniBand statistics for all InfiniBand ports on the switch, as well as for configuring port statistic thresholds.

Time Management:

Provides commands for retrieving and setting the current system time, as well as commands for setting the time zone and Daylight Saving Time parameters.

SNMP:

Provides commands for configuring SNMP trap destinations and security parameters required to access the switch from an SNMP manager.

To list commands within a functional group, enter the functional group name. For example, to list all of the firmware commands, enter `list Firmware`. The system would display the following:

```
-> list Firmware

fwUpdate             Update units firmware
fwListFiles          List the contents of the firmware ramdisk
fwShowUpdateParams   Display firmware default update parameters
fwSetUpdateParams    Configure firmware default update parameters
showCapability       Display the capabilities/features
showLastScpRetCode   Display the return code from the last SCP Firmware or XML
Config Push
fwVersion            Display Firmware revisions
bootQuery            Query boot image information
bootSelect           Change boot selection
```

Online Help

The online help for the CLI provides, for each command, all necessary information to successfully execute the command. For example, typing `help list` (can also type `list help`) displays the following information for the `list` command:

```
NAME
    list
SYNOPSIS
    list [group] [-noprompt] [-verbose]
DESCRIPTION
    List available commands.
OPTIONS
    group      - List the commands in that particular group
    -noprompt - Just list the command groups.
    -verbose  - Print full help for each command, instead of summary.
NOTES
    Specify the group name or use 'all' to list all available commands.
```

Keyboard Shortcuts

- The CLI keeps a history of recently executed commands. This history is available via the **Up** and **Down** arrow keys.
- You can edit the current command with the **Left** and **Right** arrow keys.
- Tab completion: pressing the **Tab** key after typing at least one character either completes a command or lists all the available commands that begin with the characters already typed.

Accessing the CLI

The CLI can be accessed via telnet or SSH. To use telnet follow these steps:

1. Telnet to the IP address of the switch (the default IP address is 192.168.100.9) with the following command:

```
telnet <IP ADDRESS>
```

2. The system prompts for a user name. The CLI has the following default user names:

   Operator access: `operator`

   Administrator access: `admin`

   Type the appropriate user name and press **Enter**.

3. The system prompts for a password. The CLI has the following default passwords:

   Operator access: **operpass**

   Administrator access: **adminpass**

   Type the appropriate password and press **Enter**. The system responds with:

```
Welcome to the <SWITCH> CLI. Type 'list' for the list of commands.
```

Groups

The following lists all CLI functional groups. Commands for all Intel® switches are listed in each group section. Any commands specific to a switch(es) are noted. For more specific information for each functional group, you would execute the `help <GROUP NAME>` command. For more specific command information, you would execute the `help <COMMAND NAME>` command.

- General
- Chassis
- Network
- Firmware
- Subnet Management
- Log
- Key Management
- IbSwitchInfo
- Time Management
- SNMP
- CaptureInfo

2 General

help

Description Displays help information for a specific command.

> **NOTE:** General Help
>
> Type `list` or `?` for the list of commands.
>
> To get help on a particular command enter: `<commandname> help`.
>
> For convenience, you can also enter: `help <commandname>`
>
> Use the **Up** and **Down** arrow keys to browse command history, **Left** and **Right** arrow keys to edit the current command, and the **Tab** key for tab completion of a command.
>
> Two alternate key bindings exist for the **backspace** and **delete** keys. If these keys are not responding as expected use the `swapBsDel` command to swap the bindings.
>
> Commands are grouped into subcategories. To list the commands in a subcategory, enter in the category heading. Category headings are identified by starting with a capital letter. For example, to list all the commands that handle log configuration enter `Log`.
>
> In each help description, items in brackets(e.g. `[<command>]` in this help description) represent optional parameters. User selectable input is indicated by items within the '<' and '>' symbols(e.g. `<command>`). Text outside the '<' and '>' characters is the actual text that needs to be entered. When there is more than one choice, the options are specified within '{' and '}' characters, where the options are separated by '|' characters.

Syntax `help [<command>]`

Operands `<command>`   Name of the command for which help is requested.

Example `-> help list`

```
NAME
       list

SYNOPSIS
       list [group] [-noprompt]

DESCRIPTION
       List available commands.

OPTIONS
       group    - List the commands in that particular group

       -noprompt - Just list the command groups.
```

list

Description Lists all the valid commands.

NOTE:

Use 'list all' to display brief help for all available commands.

Use 'list all -verbose' to display verbose help for all commands.

Use 'list -noprompt' to display the list of command groups.

Syntax `list [{all | <group>}] [-noprompt] [-verbose]`

Operands

`all`	List the commands for all groups.
`<group>`	List the commands in that particular group, see NOTE.
`[-noprompt]`	Displays a list of the command groups only.
`[-verbose]`	Print full help for each command, instead of summary.

Example
```
list
List of Valid Commands:
General            General commands for user management and CLI configuration.
Deprecated         These commands have been deprecated
Chassis            Chassis management commands. (FRU info, fan / power supply
                   state, etc)
Network            Ethernet interface management commands
Firmware           Update firmware and display current revision levels.
SubnetManagement   InfiniBand subnet manager configuration and management
Log                Log file display and configuration
KeyManagement      License key management
IbSwitchInfo       Infiniband port configuration and statistics
TimeManagement     Display and configure the system time
Snmp               Snmp configuration commands
CaptureInfo        Information capture commands for support personnel use.


Type the name of the group you want to list or return to exit
```

NOTE: Specify the group name or use `all` to list all available commands. To get a full list of commands and all help text, enter `list all -verbose`.

---

history

Description	Displays the command history for the CLI session.
Syntax	history
Operands	None
Example	-> history

```
command history [30 max lines]:
  list
  list Firmware
  help list
  list
  history
```

reboot

Description   Reboots the selected device.

NOTE: You may reboot the local (Master) Mgmt Module or the remote (Slave) Mgmt Module with one or multiple arguments.

Non-disruptive reboots will not interfere with switch traffic (if ASIC firmware is not changing).

Providing argument all performs disruptive reboot of all present Mgmt Modules and cards.

If rebooting the local device from telnet, ssh, etc., you will have to reconnect after rebooting.

Default (no arguments) reboots the local device disruptively after prompt.

Syntax   `reboot [now][-m][slot <slot>][-s][-n][all]`

Operands

`[now]`	Does not prompt before rebooting.
`[-m]`	Reboot Master (local) HP BLc Intel® 4X QDR IB Switch (non-disruptive).
`[slot <slot>]`	Reboot a slot (disruptive).
`[-s]`	Reboot Slave (remote) HP BLc Intel® 4X QDR IB Switch (non-disruptive).
`[-n]`	Reboot Slave (remote) management card only (non-disruptive)
`[all]`	Reboot all local devices (excludes `-n/-m/-s`).

Example
```
-> reboot
Disruptive reboot selected
Proceed with reboot? [N]
```

See also   `resetCard, showInventory`

killclisession

Description Terminates a CLI session.

> **NOTE:** This command logs out remote sessions. Use 'who' to obtain the list of active sessions.

Syntax `killclisession <sessionNumber>`

Operands `<sessionNumber>`   Session number that is returned from the 'who' command.

Example `-> killCliSession`
`must supply session number`

who

Description Displays all the active CLI sessions.

NOTE: A session can be 'active', but no user information available (in most cases, this indicates the session is waiting for the user to enter login information).

For each session the following information is displayed:

`user`	: username of the logged in user
`role`	: security role of the user
`index`	: internal session index
`logged in`	: timestamp of when the user logged in
`last cmd`	: timestamp of the users last command
`type`	: method used to connect to the system
`ip address`	: ip address of the user (if applicable)

Syntax `who`

Operands None

Example
```
-> who
user    role  index logged in            last cmd             type    ip address
----------------------------------------------------------------------------------
admin   admin 0      16:29:24 07/24/2009  16:29:24 07/24/2009  serial  0.0.0.0
admin   admin 1      07:22:45 07/29/2009  07:51:22 07/29/2009  ssh     10.32.4.16
```

broadcast

Description Write a message to all active CLI sessions.

NOTE: Writes the supplied message to all other active CLI sessions. The message to be written must be encapsulated in quotes, and nonempty.

Syntax `broadcast "<msg>"`

Operands `<msg>`   Message to be broadcasted (must be non-empty and in quotes " ").

Example `-> broadcast "The system will be rebooted in 5 minutes."`

swapBsDel

Description Toggle the key bindings for the **Backspace** and **Delete** characters.

NOTE: Terminals may bind the backspace and delete key bindings differently. This command swaps two commonly used bindings, which lets you use the **Backspace** and **Delete** keys properly without having to adjust their terminal settings. Backspace and delete swapping is persistently maintained per session (i.e. each login session can have a separate binding).

Syntax `swapBsDel`

Operands None.

Example `-> swapBsDel`

setTermWidth

Description Changes the terminal width for formatting purposes.

NOTE: Allows modification of the terminal width used for text formatting purposes. Note that not all commands adhere to this setting. The minimum width is 20 characters.

Syntax `setTermWidth <width>`

Operands `<width>`   Width of your terminal window. Minimum width is 20 characters.

Example `-> setTermWidth 100`

getTermWidth

Description Displays the terminal width for text formatting purposes.

> **NOTE:** Not all commands adhere to this setting.

Syntax `getTermWidth`

Operands None.

Example `-> getTermWidth`
`Current terminal width: 80 characters.`

prompt

Description   Changes the CLI prompt.

> **NOTE:**   The prompt may not exceed 11 characters and is not saved across reboots. If the prompt contains a space, asterisk, comma, parenthesis, or semicolon it must be enclosed with double-quotes ("). For example, `"*a prompt*"`. Also, if a prompt is not accepted try to enclose it with double-quotes.
>
> In order for some FastFabric Tools to function correctly, the prompt must end in `"-> "`(note trailing space).

Syntax   `prompt <prompt>`

Operands   `<prompt>`   The new prompt.

Example   `-> prompt "9024->"`
`9024->`

case

Description Displays or changes case sensitivity of the command interpreter for the current CLI session.

NOTE: If an option is not specified, the current case sensitivity is displayed; otherwise, the case sensitivity is turned on or off depending on the option specified. When case sensitivity is on, the CLI input must match the exact character case (lower and upper case) as specified in the help text. When case sensitivity is turned off, the input may be any combination of upper and lower case.

Syntax `case [{off | on}]`

Operands `{off | on}`   Turn case sensitivity off or on

Example `-> case off`
`Case sensitivity is now off`

showLastRetCode

Description   Displays the return code from the last command.

NOTE: This allows for automated systems to determine if a command was successful or not.

Syntax   `showLastRetCode [-brief]`

Operands   `-brief`   Displays only the return code numeric value.

Example   `-> showLastRetCode`
`Last Exit Code: 0: Success`

echo

Description Echoes the input parameters back to the output.

NOTE: If there are multiple arguments they are separated by spaces.

Syntax `echo [<text>] ...`

Operands `[<text>] ...`   Text to echo.

Example `-> echo test`
`test`

rlogin

Description   Open a terminal to a device in the chassis.

NOTE: This command allows users to open a terminal to local I/O devices within the chassis/hemisphere. The command `rcmd` is an alias of `rlogin`. If needed, you can escape out of the remote CLI session back to the local CLI by typing `"]~.<Enter>"`.

Syntax   `rlogin <hostName>`

Operands   `<hostName>`   The name of the device to connect to

Example   `-> rlogin slot2`

See also   `hostShow`

rcmd

Description Open a terminal to a device in the chassis.

NOTE: This command allows users to open a terminal to local I/O devices within the chassis/hemisphere. The command `rcmd` is an alias of `rlogin`. If needed, you can escape out of the remote CLI session back to the local CLI by typing `"]~.<Enter>"`.

Syntax `rcmd <hostName>`

Operands `<hostName>` The name of the device to connect to.

Example `-> rcmd Host1`

See also `hostShow`

resetCard

Description Reboots a specific slot in the chassis.

> **NOTE:** If you reset the switch you are connected to you will have to reconnect. You can find out the valid slots for this device with the `showInventory` command.

Syntax `resetCard [<slot>] [now]`

Operands `<slot>`   Chassis slot number.

`now`   No confirmation before resetting the card.

Example `-> resetCard 2 now`

logout

Description Terminates the current CLI session.

Syntax `logout`

Operands None

Example `logout`

user

Description Changes user accounts.

NOTE: Use this command to change to the operator or admin account.

Syntax `user [<username>]`

Operands `<username>`   User account name to change to.

Example `-> user operator`
`User changed to: operator`

passwd

Description   Changes user's password(s).

NOTE: Allows the modification of the user's password.

Syntax   `passwd [<username>] [-r]`

Operands   `<username>`   Name of the user whose password needs to be changed.

`-r`   Reset user account password, only allowed from serial connection.

Example   `-> passwd operator1`
`User password changed successfully`

userAdd

Description Adds a user account.

> **NOTE:** `<role>` is one of the following: `'admin'`, `'operator'` or `'support'`.

Syntax `userAdd <role> <username> [<password>]`

Operands `<role>`   One of the privilege roles, see NOTES.

`<username>`   New user account name. Must be between 4 and 32 characters.

`<password>`   Optional password. If not supplied the default password for that role is used.

Example
```
-> userAdd admin Bob
User added: Bob
Password is set to the default password for this role: admin
```

userRem

Description	Removes a user account.
Syntax	`userRem <username>`
Operands	`<username>` User account name to delete.
Example	`-> userRem Bob` `User deleted: Bob`

userListShow

Description Lists all user accounts for this device.

Syntax `userListShow`

Operands None.

Example
```
-> userListShow
username      role
operator      operator
USERID        admin
Bob           admin
```

sshKey

Description   Displays or modifies the configured set of SSH keys.

NOTE: Users with administrative privileges may use the `-u` option to manage keys for other users. When valid key is present, user can log in without a password.

Syntax   `sshKey [{show | add "<key>" | rem <index> | rem --all} [-u <username>]]`

Operands   `show`   Displays the SSH public keys in the users `authorized_keys` file.

`add "<key>"`   Add key to the users `authorized_keys` file. Must be escaped with double-quotes (`"<key>"`).

`rem <index>`   Remove key at `<index>` for the user

`rem --all`   Remove all keys for the user.

`-u <username>`   Perform the operation on the user `<username>` (for admins only).

Example
```
-> sshKey show
 Index    Key
--------------------------------
    1        "ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEArNiSexu30rZjs1HAXbDBwTgJgcxLF..."
```

loginMode

Description Changes the login authentication mode.

> **NOTE:** This command displays or changes how users are authenticated when connecting to the GUI or CLI. With no parameters, the current login mode is displayed. The login mode can be changed by specifying a single integer parameter, indicating which of the following modes should become active.When usernames are disabled, all users will be logged on as the administrative user. There is no way to change this behavior.

Syntax `loginMode [{0 | 1 | 2 | 3}]`

Operands

`0`	Username and password required
`1`	Password is not required
`2`	Username / password are not required
`3`	LDAP authentication

Example
```
-> loginMode 1
Mode successfully changed to: 1 = Password is not required
```

setldapSvrIpAddr

Description Displays or modifies the LDAP server IP address.

NOTE: This sets the LDAP server IP address used for LDAP user login authentication.If the `<ipaddress>` is not specified, the currently configured `<ipaddress>` is displayed.

Syntax `setldapSvrIpAddr [<ipaddress>]`

Operands `<ipaddress>`   IP address to use for LDAP, in the format "192.168.0.1"

Example `-> setldapSvrIpAddr 192.168.0.29`

setldapSvrPort

Description   Displays or modifies the TCP port number to use for LDAP.

NOTE: This configures the TCP port number used for LDAP user login authentication.If the `<port>` is not specified, the currently configured `<port>` is displayed.

Syntax   `setldapSvrPort [<port>]`

Operands   `<port>`   TCP port number to use for LDAP.

Example   `-> setldapSvrPort 389`

idleTimeoutGet

Description Get the UI idle timeout value.

NOTE: Displays the idle timeouts for the GUI and CLI interfaces to the system. When the `<timeout>` is zero, the timeout for that interface is disabled.

Syntax `idleTimeoutGet [{--all | --cli | --gui}]`

Operands `--all`   Display all timeouts. This is the default if no options are specified.

`--cli`   Display the timeout for CLI sessions.

`--gui`   Display the timeout for GUI sessions.

Example `-> idleTimeoutGet`
`CLI timeout is 600 seconds.`
`GUI timeout is 0 seconds.`

idleTimeoutSet

Description Sets the UI idle timeout value.

NOTE: Timeouts are in number of seconds, with zero meaning the timeout is disabled.

Syntax `idleTimeoutSet [{--all | --cli | --gui}] <timeout>`

Operands `<timeout>`   Timeout value (in seconds).

`--all`   Set the idle timeout for both the CLI and the GUI to the same value..

`--cli`   Set the idle timeout for only the CLI. This is the default if no identifier is specified.

`--gui`   Set the idle timeout for only the GUI.

Example
```
-> idleTimeoutSet --all 700
Timeout is set to 700 seconds.
```

sessionTimeoutDisable

Description Disables the idle timeout for the current CLI session.

NOTE: This timeout does not persist across instances of the session (i.e. each time you log on, it will default back to the system default value).

Syntax `sessionTimeoutDisable`

Operands None.

Example `-> sessionTimeoutDisable`
`Disabled session idle timeout.`

sessionTimeoutEnable

Description   Enables the idle timeout for the current CLI session.

> **NOTE:**   This timeout does not persist across instances of the session (i.e. each time you log on, it will default back to the system default value).

Syntax   `sessionTimeoutEnable`

Operands   None.

Example   `-> sessionTimeoutEnable`
`Enabled session idle timeout.`

loginMsgGet

Description Displays the CLI login message for this device.

NOTE: This message can be customized with the `loginMsgSet` command.

Syntax `loginMsgGet`

Operands None.

Example `-> loginMsgGet`
`Welcome message: Be certain to logout when you are finished using the CLI.`

loginMsgSet

Description   Sets the CLI login message for this device.

NOTE: This command can be used to modify the welcome message displayed when logging onto the CLI. The current message can be viewed with the `loginMsgGet` command.

Syntax   `loginMsgSet {-clear | <string>}`

Operands   `-clear`   Clear welcome message.

                       `<string>`   Set the welcome to `<string>`.

Example   `-> loginMsgSet "Be certain to logout when you are finished using the CLI."`
`Welcome message set successfully`

loginNameGet

Description Displays the text string that is displayed prior to logging in via telnet.

NOTE: The login name is an arbitrary string displayed prior to a user attempting a login to a new CLI session. This command displays the current value of this string.

Syntax `loginNameGet`

Operands None.

Example `-> loginNameGet`
`login-name: Switch1`

loginNameSet

Description Sets the text string that is displayed prior to logging in via telnet.

NOTE: The login name is an arbitrary string displayed prior to a user attempting a login to a new CLI session. This command modifies this string.

Syntax `loginNameSet {-clear | <loginName>}`

Operands `-clear`   Clear the login name.

`<loginName>`   Set the login name to `<loginName>`.

Example `-> loginNameSet Switch1`
`login-name set successfully`

serialAuthGet

Description   Displays the mode setting for serial console authentication.

NOTE: Displays whether user login and authentication is required on the serial console of the system.

Syntax   `serialAuthGet`

Operands   None.

Example   `-> serialAuthGet`
`Serial authentication is currently disabled.`

serialAuthSet

Description Changes the serial console authentication mode setting.

NOTE: This command is used to enable or disable user authentication on the serial console. Use `1` as the only parameter to enable authentication, or use `0` to disable authentication. Exit and re-login for setting to take effect.

Syntax `serialAuthSet {0 | 1}`

Operands `{0 | 1}`   `<enable>` values: 0=disable, 1=enable.

Example `-> serialAuthSet 1`
`Serial authentication set to enabled.`

uiConfig

Description   Displays or modifies the user interface access methods.

NOTE: There are several access methods for this device. With no arguments, this command displays the access methods that can be configured, and whether each method is currently enabled or disabled.

This command can also be used to enable or disable various access methods. Use `-<proto> 0` to disable a protocol, and `-<proto> 1` to enable a protocal. Multiple operations may be specified in a single command.

A reboot is not required for this command to take effect. Depending on the specific method, it may take 5–10 seconds for the change to take effect. Note that existing connections (for example, a Telnet session) will not be affected by disabling the underlying access method.

Any access method supported by the device (but not listed here) cannot be configured and is always enabled.

Syntax   `uiConfig [-telnet {0|1}] [-https {0|1}] [-http {0|1}] [-ftp {0|1}] [-snmp {0|1}]`

Operands   `-telnet {0|1}`   Disable(0) or enable(1) telnet access.

`-https {0|1}`   Disable(0) or enable(1) https access.

`-http {0|1}`   Disable(0) or enable(1) http access.

`-ftp {0|1}`   Disable(0) or enable(1) ftp access.

`-snmp {0|1}`   Disable(0) or enable(1) snmp access.

Example
```
-> uiconfig -telnet 0
Successfully modified configuration.
 Option          Value
 ----------------------
 telnet          0
 http            1
 https           0
 ftp             1
 snmp            1
```

genPost

Description Displays General POST results and runs tests.

Syntax `genPost [{all | <test>}]`

Operands `all`   Run all tests.

`<test>`   Run specific test number.

Example `-> genPost`

```
ID Gen POST Name                    Result   ENA DEM PWR Result-Ext
---------------------------------------------------------------------
*   POST TEST CPU                   NO-RUN    Y   N   Y
*   POST TEST MEM DRAM              NO-RUN    Y   N   Y
*   POST TEST BOOTROM IMG CHKSUM    NO-RUN    Y   N   Y
*   POST TEST RUN-TIME IMG CHKSUM   NO-RUN    Y   N   Y
*   POST TEST JMP RAM               NO-RUN    Y   N   Y
6   POST TEST I2C PROBE             NO-RUN    Y   Y   N
7   POST TEST FAN TRAYS             NO-RUN    Y   Y   N
8   POST TEST PWR SUPPLIES          NO-RUN    Y   Y   N
9   POST TEST REAL-TIME CLOCK       NO-RUN    Y   Y   N
10  POST TEST PROTO                 NO-RUN    Y   Y   N

Enter Test Index (0 to Exit, 'all' for All tests)
```

exit

Description Terminates the current CLI session.

Syntax `exit`

Operands None.

Example `exit`

3 Chassis

hwCheck

Description Checks the hardware status for chassis (or hemisphere), modules, fans and power supplies.

NOTE: Returns OK or provides detailed status/warning/error information. If an error/warning is detected, this command automatically provides verbose information.

Syntax `hwCheck [{-verbose | Verbose output mode.`

Operands `-verbose`   Name of the command for which help is requested.

`{0 | 1}`   0=quiet(default), 1=verbose.

hwMonitor

Description Monitor fans, power supplies, and port state.

NOTE: To exit monitoring, press the enter key.

The <slot> may require a prefix. The prefix for a Leaf is 'L', the prefix for a Spine is 'S', the prefix for a Management Module is 'M', the prefix for a Power Supply is 'P', the prefix for a Fan is 'F' and the prefix for the Chassis is 'C'. The Leaf, Spine and Management Module slots do not require a prefix. The Power Supply, Fan and Chassis slots require a prefix since their slot numbers overlap with the Leaf, Spine and Management Module slot numbers.

The character meanings on the 12800 main monitoring page are as follows:

P column heading notes:

'*' = device is inserted and powered on.

'!' = device is inserted and powered off.

'A' = device is required in the configuration, but is not inserted.

' ' = device is not required in the configuration, and is not inserted.

TMP headings have two columns:

E column heading:

 'X' = a temperature error condition exists on the device.

 ' ' = no temperature error condition on the device.

W column heading:

 'X' = a temperature warning condition exists on the device.

 ' ' = no temperature warning condition on the device.

VLT headings have two columns:

E column heading:

 'X' = a voltage error condition exists on the device.

 ' ' = no voltage error condition on the device.

W column heading:

 'X' = a voltage warning condition exists on the device.

 ' ' = no voltage warning condition on the device.

Syntax `hwMonitor [<slot>] [onepass] [-all]`

Operands

`<slot>`	slot number.
`onepass`	Display current values and exit. Don't monitor continuously.
`-all`	Display all information screens.

showIBNodeDesc

Description Displays the IB Node (SMA) Description (or the Default).

NOTE: Entering this command with no parameters displays the current Node Description. The -d option will display the unit's default value.

Syntax showIBNodeDesc [-d]

Operands -d   Show the default Node Name for this unit.

setIBNodeDesc

Description Set the IB Node (SMA) Description.

NOTE: New node name must be enclosed in quotes. Node name must be no more than 64 characters.

Syntax `setIBNodeDesc <nodeString>`

Operands `<nodeString>`   New node string.

setIBNodeDescFormat

Description Displays or modifies the IB Node (SMA) Description format modifier.

NOTE: If the `<format>` is not specified, then the current format selection value is displayed. When `<format>` is `0`, the SMA node description is in a verbose format. When `<format>` is `1` the format is brief. The brief format is more consistent with the CLI/GUI Port Stat port naming.

Syntax `setIBNodeDescFormat [<format>]`

Operands `<format>`   0 = verbose, 1 = brief format.

fruInfo

Description Displays the field irreplaceable unit information.

Syntax `fruInfo {<slot> | -all}`

Operands `<slot>`   Slot number.

`-all`   Display information for all slots.

chassisQuery

Description Displays information about the line cards in a chassis.

> **NOTE:** Entering this command with no parameters will display all currently occupied slots which support Firmware Update.

Syntax `chassisQuery [-master] [-slave] [<slot>] [-showType] [-type <cardtype>]`
`          [-ignoreInvalidType]`

Operands

`-master`	Display master slot number.
`-slave`	Display slave slot number.
`<slot>`	Slot number.
`-showType`	Display the card type.
`-type <cardtype>`	Display slots that have given card type.
`-ignoreInvalidType`	Do not return an error if an invalid card type is supplied.
	Note: This option is only valid when used with `-type`.

showInventory

Description Displays asset information on all entities in the chassis.

Syntax `showInventory`

Operands None.

setSystemContact

Description Displays or modifies the chassis system contact information.

> **NOTE:** If the system <contact> is not specified, then the current system contact is displayed.

Syntax `setSystemContact [<contact>]`

Operands `<contact>`   System contact string.

setSystemName

Description Displays or modifies the chassis system name.

NOTE: If the system <name> is not specified, then the current system name is displayed.

Syntax `setSystemNameset [<name>]`

Operands `<name>`   System name string.

setSystemLocation

Description Displays or modifies the chassis system location.

> **NOTE:** If the system `<location>` is not specified, then the current system location is displayed.

Syntax setSystemLocation [<location>]

Operands `<location>`   System location string.

case

Description Displays or changes case sensitivity of the command interpreter for the current CLI session.

NOTE: If an option is not specified, the current case sensitivity is displayed; otherwise, the case sensitivity is turned on or off depending on the option specified. When case sensitivity is on, the CLI input must match the exact character case (lower and upper case) as specified in the help text. When case sensitivity is turned off, the input may be any combination of upper and lower case.

Syntax `case [{off | on}]`

Operands `{off | on}`   Turn case sensitivity off or on

Example `-> case off`
`Case sensitivity is now off`

4 Network

ifShow

Description Displays the interface statistics for the OOB management port.

NOTE: This routine displays the attached network interfaces for debugging and diagnostic purposes. If `<ifName>` is given, only the interfaces belonging to that group are displayed. If `<ifName>` is omitted, all attached interfaces are displayed.

Syntax `ifShow [<ifName>]`

Operands `<ifName>`   The network interface name.

routeShow

Description   Displays the interface routes for the OOB management port.

Syntax   `routeShow`

Operands   None.

ping

Description Send Ping packets to a specified IP address.

NOTE: This routine spawns a process to send ping packets to the specified hostname or IP address. If `<packetCount>` is given, the process exits after that number of packets are sent. If `<packetCount>` is omitted, a default the number of packets defaults to 5. `<hostname>` is limited to 64 characters.

Syntax `ping <hostname> or <ipAddress> [<packetCount>]`

Operands

`<hostname> or <ipAddress>`	The network hostname or IP address to ping.
`<packetCount>`	The number of packets to ping the host with (default is 5).

ping6

Description   Sends Ping packets to a specified IPv6 address.

NOTE: This routine spawns a process to send ping packets to the specified IPv6 address. If `<packetCount>` is given, the process exits after that number of packets are sent. If `<packetCount>` is omitted, a default the number of packets defaults to 5.

Syntax   `ping6 [-n] [-q] [-v] [-c <packetCount>] [-I <interface>] <ipv6Address>`

Operands

`-n`	Numeric output only.
`-q`	Only display summary lines.
`-v`	Verbose output.
`-c <packetCount>`	The number of packets to ping the host with (default is 5).
`-I <interface>`	The local interface name to use (useful for link local ping).
`<ipv6Address>`	The network host to ping, scoped address accepted.

showChassisIpAddr

Description Displays the chassis IP address.

Syntax `showChassisIpAddr`

Operands None

setChassisIpv6Addr

Description   Changes the user configured chassis IPv6 address.

NOTE: Changing the chassis IPv6 address may drop the connection to the CLI if not using the console port, and may cause the device to become unreachable.

Syntax   `setChassisIpv6Addr [<ipv6Address> /<maskLen>]`

Operands   `<ipv6Address>`   The new IPv6 address in notation format 'a:b:c:d:e:f:g:h', embedded 0s can be shortened, eg a:b::g:h.

`<maskLen>`   Network mask length. (0-128).

delChassisIpv6Addr

Description   Deletes the user configured chassis IPv6 address.

NOTE: Changing the chassis IPv6 address may drop the connection to the CLI if not using the console port, and may cause the device to become unreachable. Does not affect link-local or auto configured addresses.

Syntax   `delChassisIpv6Addr`

Operands   None

showChassisIpv6Addr

Description Displays the user configured chassis IPv6 address.

NOTE: Only displays the user configured IPv6 address. To see all the IPv6 addresses, including any auto configured addresses, please use the `'ifShow'` command.

Syntax `showChassisIpv6Addr`

Operands None

autoConfIPv6Enable

Description Enables IPv6 address autoconfig.

> **NOTE:** Allows prefix learning from attached routers. Does not affect link-local or manually configured addresses.

Syntax `autoConfIPv6Enable`

Operands None

autoConfIPv6Disable

Description Disables IPv6 address autoconfig.

NOTE: Prevents prefix learning from attached routers. Does not affect link-local or manually configured addresses.

Syntax `autoConfIPv6Disable`

Operands None

autoConfIPv6Show

Description Displays IPv6 address autoconfig setting.

Syntax `autoConfIPv6Show`

Operands None

ndpShow

Description   Displays IPv6 Neighbors table.

Syntax   `ndpShow`

Operands   None

showDefaultRoute

Description Displays the default gateway IP address.

> **NOTE:** This is the IP address for the default gateway to route packets from the OOB management port to an external network.

Syntax showDefaultRoute

Operands None

arpShow

Description Displays the link level address resolution protocol (ARP) table.

Syntax `arpShow`

Operands None

hostShow

Description Displays the host name table.

Syntax `hostShow`

Operands None

dhcpEnable

Description Enables DHCP on the Ethernet interface.

NOTE: Enables the DHCP client subsystem, requests a DHCP lease, and then configures the interface with the lease data from the server.

Syntax dhcpEnable

Operands None

dhcpShow

Description Displays the current DHCP leases.

Syntax `dhcpShow [-verbose] [-v]`

Operands `-verbose`   Verbose output mode.

                  `-v`   verbose display.

dnsParamsShow

Description Displays the stored DNS parameters.

NOTE: This command retrieves the stored configuration parameters used for Domain Name Resolution.

Syntax dnsParamsShow

Operands None

dnsParamsSet

Description Changes the DNS configuration parameters.

NOTE: The DNS resolver cannot be enabled until both the server address and local domain name have been configured. It will be necessary to manually reboot the switch in order to start or stop the DNS resolver.

Syntax `dnsParamsSet [-s <ipaddress>] [-d <domain name>] [-e <1 or 0>]`

Operands `-s <ipaddress>`   DNS server IP address in dotted notation format 'xxx.xxx.xxx.xxx'.

`-d <domain name>`   Local domain name where this switch is installed (limit 32 chars).

`-e <1 or 0>`   1 = enable the DNS resolver, 0 = disable the DNS resolver.

IpoIbConfigShow

Description Displays the IpoIb Enable/Disable setting.

Syntax `IpoIbConfigShow`

Operands None.

Example `-> IpoIbConfigShow`

IpoIbConfigEnable

Description Enables the IPoIB feature.

Syntax `IpoIbConfigEnable`

Operands None

Example `-> IpoIbConfigEnable`

IpoIbConfigDisable

Description Disables IPoIB feature.

Syntax `IpoIbConfigDisable`

Operands None

Example `-> IpoIbConfigDisable`

IpoIbAddressShow

Description   Displays the IpoIb IP address and associated netmask settings.

Syntax   `IpoIbAddressShow`

Operands   None.

Example
```
-> IpoIbAddressShow
IpoIb feature is currently enabled
 IP Address: 10.33.32.206 Net mask: 255.255.255.0
```

IpoIbAddressSet

Description Changes the IpoIb IP address and netmask settings.

Syntax `IpoIbAddressSet -h <ipaddress> [-m <netMask>]`

Operands `-h <ipaddress>`   The new IP address in dotted notation format '`xxx.xxx.xxx.xxx`'.

`-m <netMask>`   Network mask (in hexadecimal format).

Example `-> IpoIbAddressSet -h 123.45.6.789`

IpoIbAddressSetIPv6

Description   Changes the IPoIB IPv6 address and netmask.

NOTE: The <ipv6Address/netMaskLen> is made up of two major components, separated by a '/'. The ipv6Address is in notation format 'a:b:c:d:e:f:g:h', where embedded 0s can be shortened, e.g. 'a:b::g:h'. The netMaskLen is a value between 0 and 128 inclusive.

Syntax   `IpoIbAddressSetIPv6 <ipv6Address/netMaskLen>`

Operands   `<ipv6Address/`   See NOTE.
               `netMaskLen>`

IpoIbAddressShowIPv6

Description Displays the IPoIB IPv6 Address and associated netmask settings.

Syntax `IpoIbAddressShowIPv6`

Operands None

5 Firmware

fwUpdate

Description Updates units firmware.

NOTE: Firmware update works by using ftp to retrieve the firmware file, then writes the file to flash. Omitting the options will cause the user to prompted for this information.

Syntax `fwUpdate [<hostip> <username> <password> <directory> <filename>]`

Operands `<hostip>`   Host ip address where the file resides.

`<username>`   User name for ftp.

`<password>`   User password for ftp.

`<directory>`   Directory to cd to after logging in.

`<filename>`   File name of the firmware file.

Example
```
-> fwUpdate
Enter 1 for FTP, 2 for local file: 1
Ftp Server IP Address:[10.32.2.150] 10.32.2.150
Ftp username:[ftp] ftp
Ftp password:[ftp] ftp
File Directory:[/Integration/Test/] /Integration/Test/
File name:[X.xt3.pkg] X.xt3.pkg
Save changes? [Y] n
Attempting to initiate firmware update
  Product               = Intel_12000_Series
  Version               = 0vbel.040809.1217
  Compressed Image Size = 6380623 bytes
  md5                   = 7e2c72bf9e261fc81ec0454fd954ecb0
  vxWorks Image Type    = loadable
  Computed md5          = 7e2c72bf9e261fc81ec0454fd954ecb0
  md5 values match!
Firmware update initiated successfully
Info, update of all IDB+ boot image data completed successfully.
Image       = /image1
Product     = Intel_12000_Series
BSP         = xt3
Version     = 0vbel.040809.1217
Image Size  = 6380623 bytes
md5         = 7e2c72bf9e261fc81ec0454fd954ecb0
Computed m1ec0454fd954ecb061fc8Flash write completed: - Updating configuration
information
md5 values agree.
Info, update of all IDB+ boot image data completed successfully.

Firmware update completed successfully
```

fwListFiles

Description Lists the contents of the firmware ramdisk.

NOTE: The firmware directory temporarily stores firmware files before they are written to flash.

Syntax `fwListFiles`

Operands None.

Example `-> fwListFiles`

```
Listing Directory /firmware:
[PATH TO FIRMWARE FILE] / [FIRMWARE FILE NAME]
```

fwShowUpdateParams

Description   Displays the firmware default update parameters.

Syntax   fwShowUpdateParams

Operands   None.

Example
```
-> fwShowUpdateParams
Firmware update ftp configuration settings:
 host:[10.33.21.43]
 user:[ftp]
 password:[ftp]
 directory:[/]
 filename management:[X.xt3.pkg]
 filename evic:[]
 filename fvic:[]
```

fwSetUpdateParams

Description Configure firmware default update parameters.

NOTE: Modifies the default update firmware parameters. Except for the `filename` option, all parameters are shared across all card types. The `cardtype` parameter is only required when specifying the filename.

Syntax `fwSetUpdateParams [-c {management | evic | fvic}] [-h <hostname>] [-u <username>] [-p <password>] [-d <directory>] [-f <filename>]`

Operands

`-c {management | evic | fvic}`	Specifies card type.
`-h <hostname>`	The host name or IP address of the ftp server.
`-u <username>`	The username used to access the ftp server.
`-p <password>`	Thepassword used to access the ftp server.
`-d <directory>`	The directory where the firmware file can be found.
`-f <filename>`	The name of the firmware file.

Example `-> fwSetUpdateParams -h mnftpserver -d /fwlocation`
`The changes have been applied.`

showCapability

Description Displays the capabilities/features.

Syntax `showCapability [-key <feature>]`

Operands `-key <feature>`   Display information about a particular feature.

Example `-> showCapability`
`fwPush: 1`

showLastScpRetCode

Description Displays the return code from the last SCP Firmware or XML Config Push.

NOTE: This allows for automated systems to determine if an SCP firmware or XML config push was successful or not.

Syntax `showLastScpRetCode [<slot>] [-all]`

Operands `<slot>`   Slot number.

`-all`   All slots in the chassis.

Example `-> showLastScpRetCode -all`
`SCP: Slot 0 Last Exit Code: 0: Success`

fwVersion

Description Displays the firmware revisions.

Syntax `fwVersion <slot>`

Operands `<slot>`   Slot number.

Example
```
-> fwversion
HP BLc Intel 4X QDR IB Switch Information ----------------
  Firmware Version: 5.0.3.0.8
  Firmware build:   5_0_3_0_8
  Firmware BSP:     xt3
  MBC Version:      None
  Bootrom Version:  5.0.3.0.2
```

bootQuery

Description   Query boot image information.

Syntax   `bootQuery <slot> [-active | -alternate | -all]`

Operands   `<slot>`   Slot number.

         `-active`   Displays the version of the active firmware image.

         `-alternate`   Displays the version of the alternate firmware image.

         `-all`   Displays the versions for the primary and alternate firmware images.

Example
```
-> bootquery 0 -all
Primary firmware version: 5.0.3.0.8
Alternate firmware version: 5.0.3.0.6
Active firmware version: 5.0.3.0.8
```

bootSelect

Description Selects the next boot image to be used.

NOTE: This command allows the user to set the next boot image for the device. A '*' next to the image entry indicates the currently selected bootimage. A #' indicates the currently active bootimage.

Syntax
```
bootSelect <slot> [-i <index>] [-alternate] [-version <version>]
          [-noprompt]
```

Operands

`<slot>`	Slot number.
`-i <index>`	Index of the boot image to boot next.
`-alternate`	Make the alternate image the one to boot next.
`-version <version>`	Make this version the image to boot next.
`-noprompt`	Just show the current configuration.

Example
```
-> bootselect 0 -noprompt
Currently installed firmware versions
index : alias    : version
------------------------
    1  : jeh      : unknown;
    2  : ron      : unknown;
*# 3  : image1   : 5.0.3.0.8;
    4  : image2   : 5.0.3.0.6;

  * - indicates Primary image (will run at next reboot)
  # - indicates Active image

Default boot image index = 3
```

6 Subnet Management

The Subnet Management commands require that the Intel® Fabric Manager be installed and running. The Intel® True Scale Suite Fabric Manager is part of the Intel® True Scale Fabric Suite (IFS) and requires a license to use. Table 1 is a list of the Subnet Management CLI commands. For information on each command, refer to the *Intel® True Scale Fabric Suite Fabric Manager User Guide*.

Table 1 Subnet Management CLI Commands List

Command	Definition
smControl	Control the Subnet Manager.
smConfig	Configure the Subnet Manager.
smForceSweep	Force the subnet manager to sweep (does not change sweep rate).
bmForceSweep	Force the baseboard manager to sweep (does not change sweep rate).
smShowSMParms	Displays the SM parameters switch lifetime, HOQ lifetime, VLStall val, pkt lifetime, and dynamic PLT.
smShowLids	Displays all the fabric LID information as known by the subnet manager.
smShowGroups	Displays the multicast group information from subnet manager.
smShowServices	Displays the subnet administration service records of the subnet manager.
smShowSubscriptions	Displays the event forwarding (subscription) table from subnet manager.
smShowMasterLid	Displays the master SM's LID, which might not be this SM's lid.
smShowLid	Displays this SM's LID.
smShowLidMap	Displays the SM's LID to port GUID map.
smShowMaxLid	Displays the highest LID allocated by the subnet manager.
smShowCounters	Displays various statistics and counters maintained by the SM.
smResetCounters	Resets various statistics and counters maintained by the SM.
pmShowCounters	Displays various statistics and counters about the PM.
pmResetCounters	Resets various statistics and counters about the PM.
pmShowRunningTotals	Displays PM's Running Total Counters for all ports in fabric.
smOptionConfig	Displays support for certain runtime options.
smPriority	Displays the priority of the subnet manager.
bmPriority	Displays the priority of the baseboard manager.
pmPriority	Displays the priority of the performance manager.
smRestorePriority	Relinquishes elevated priorities for subnet, baseboard and performance managers.
smSweepRate	Displays or dynamically sets the sweep rate of the subnet manager.
smMasterLMC	Displays the Master SM's LMC value to be used on CA ports.
smSwitchLifetime	Displays or dynamically sets the default switch lifetime (time a packet can live in a switch) of the subnet manager.
smHoqLife	Displays or dynamically sets the head of queue lifetime of the subnet manager.

Table 1 Subnet Management CLI Commands List (continued)

smVLStall	Displays or dynamically sets the VL Stall value of the subnet manager.
smDynamicPlt	Displays the dynamic packet lifetime table values.
smPKeys	Displays the PKeys.
smInfoKey	Displays the subnet manager's SMInfo key.
smMgmtKey	Displays the subnet manager's management key.
smDefBcGroup	Displays the default broadcast group parameters for the SM.
smGidPrefix	Displays the subnet manager's subnet prefix.
smSubnetSize	Displays the subnet manager's subnet size.
smTopoErrorThresh	Displays the subnet manager's error threshold for a sweep.
smTopoAbandonThresh	Displays the max consecutive times the SM can abandon a sweep.
smMaxRetries	Displays the maximum number of SM receive retries.
smRcvWaitTime	Displays max time to wait for a reply to an SM packet in millisecs.
smNonRespDropTime	Displays seconds to wait before dropping a non-responsive node.
smNonRespDropSweeps	Displays sweeps to wait before dropping a non-responsive node.
smLogLevel	Displays or dynamically sets the Subnet Manager logging level.
smLogMode	Displays or dynamically sets the Subnet Manager logging mode.
smLogMask	Displays or dynamically sets the Subnet Manager logging mask for a specific subsystem.
smMcLidTableCap	Displays limit of multicast lids available for allocation.
smMasterPingInterval	Displays master SM ping interval.
smMasterPingFailures	Displays number of master ping failures allowed.
smDbSyncInterval	Displays how often (in minutes) a Master SM performs a full sync with standby SMs.
sm1xLinkMode	Displays how the SM handles links that come up at 1x.
smTrapThreshold	Displays the urgent trap threshold before auto-disabling ports.
smAppearanceMsgThresh	Displays threshold for appearance & disappearance log messages.
smPmBmStart	Controls start of PM, BM and FE.
smShowRemovedPorts	Displays ports that have been automatically removed from the fabric.
smResetConfig	Resets the XML config to factory defaults.
smShowConfig	Displays the XML config file.
smMcastCheck	Enables or disables the Common MTU and rate for multicast groups.
smLooptestStart	Starts the SM Loop Test in normal mode.
smLooptestFastModeStart	Starts the SM Loop Test in fast mode.
smLooptestStop	Stops the SM Loop Test.
smLooptestInjectPackets	Injects packets into the SM Loop Test.
smLooptestInjectAtNode	Injects packets to a specific switch node for the SM Loop Test.

Table 1 Subnet Management CLI Commands List (continued)

`smLooptestInjectEachSweep`	Injects packets on each sweep for the SM Loop Test.
`smLooptestPathLength`	Sets the loop path length for the SM Loop Test.
`smLooptestMinISLRedundancy`	Sets the minimum ISL redundancy in fast mode for the SM Loop Test.
`smLooptestShowLoopPaths`	Displays the loop paths for the SM Loop Test.
`smLooptestShowSwitchLft`	Displays an LFT for the SM Loop Test of a switch.
`smLooptestShowTopology`	Displays the topology for the SM Loop Test.
`smLooptestShowConfig`	Displays the configuration for the SM Loop Test.

7 Log

logShow

Description Displays the log file.

NOTE: This command will display the log file as it is contained in RAM.

Syntax `logShow`

Operands None.

Example `-> logshow`

```
W|2006/10/04 20:26:31.176U: Thread "Log" (0x8fdab3b0)
        Log: Unable to Send Trap: 523:Bad:65535
W|2006/10/04 20:26:31.176U: Thread "CPU1" (0x1)
        Fcpi: Target Device 1 (2e7 p1) NPort Id 0x0106d1: Connection Restored
W|2006/10/04 20:26:31.196U: Thread "Log" (0x8fdab3b0)
        Log: Unable to Send Trap: 523:Bad:65535
W|2006/10/04 20:26:31.216U: Thread "CPU1" (0x1)
        Fcpi: Target Device 2 (b31 p2) NPort Id 0x0106d2: Connection Restored
W|2006/10/04 20:26:31.236U: Thread "CPU1" (0x1)
        Fcpi: Target Device 3 (d94 p3) NPort Id 0x0106d3: Connection Restored
W|2006/10/04 20:26:31.246U: Thread "Log" (0x8fdab3b0)
        Log: Unable to Send Trap: 523:Bad:65535
W|2006/10/04 20:26:31.256U: Thread "CPU1" (0x1)
        Fcpi: Target Device 6 (ac6 p6) NPort Id 0x0106d6: Connection Restored
W|2006/10/04 20:26:31.276U: Thread "Log" (0x8fdab3b0)
        Log: Unable to Send Trap: 523:Bad:65535
W|2006/10/04 20:26:31.326U: Thread "Log" (0x8fdab3b0)
        Log: Unable to Send Trap: 523:Bad:65535
W|2006/10/04 20:28:29.912U: Thread "tTelnetd" (0x8fe143e0)
        Osa: telnetd: connection requested by 192.168.0.107
W|2006/10/04 20:46:26.113U: Thread "tTelnetd" (0x8fe143e0)
        Osa: telnetd: connection requested by 192.168.0.107
W|2006/10/05 19:37:08.727U: Thread "tTelnetd" (0x8fe143e0)
        Osa: telnetd: connection requested by 192.168.0.46
W|2006/10/05 20:19:20.101U: Thread "tTelnetd" (0x8fe143e0)
```

logClear

Description Clears the log file.

NOTE: This command will delete all entries in the log file.

Syntax `logClear [-noprompt]`

Operands `-noprompt`   Delete all log messages without prompting the user.

Example `-> logclear`
This command will delete all log messages currently stored in RAM on HP BLc
Intel 4X QDR IB Switch.
Proceed with log clear? [N] y
RAM Log messages cleared

logConfigure

Description Configures the log settings.

NOTE: This is an interactive command to configure log settings. This involves setting which log levels are active.

Definitions:

- Preset: Enable or disable each log level that may be generated on the system.
- Device: Enables a device to display or process log messages of each level.
- Syslog: Configures the syslog host IP address and port.

Syntax `logConfigure`

Operands None.

Example
```
-> logconfigure
Type Q or X to exit.
Please enter the number corresponding to what you want to configure.
index : name            : description
-------------------------
  1   : Device          : Logging device. (IE. Ram, syslog, etc)
  2   : Preset          : General log filter.

Select: 1
Configurable devices
index : name      : |D|F|E|A|W|P|C|I|P|N|1|2|3|4|5|
-------------------------------------------------------
  1   : Ram       : |X|X|X|X|X| | | | |X| | | | | |
  3   : Console   : |X|X|X|X|X| | | | |X| | | | | |
  5   : Syslog    : |X|X|X|X|X| | | | |X| | | | | |
Type Q or X to exit

Enter the device index you wish to configure: 1
Level: Dump [1]
Level: Fatal [1]
Level: Error [1]
Level: Alarm [1]
Level: Warning [1]
Level: Partial [0]
Level: Config [0]
Level: Info [0]
Level: Periodic [0]
Level: Notice [1]
Level: Debug1 [0]
Level: Debug2 [0]
Level: Debug3 [0]
Level: Debug4 [0]
Level: Debug5 [0]
Log device configuration changed
```

logResetToDefaults

Description Restores the log file default settings.

Syntax `logResetToDefaults [-noprompt]`

Operands `-noprompt`   Restore the defaults without prompting the user.

Example `-> logResetToDefaults`
`This command will reset all log configurations for HP BLc Intel 4X QDR IB Switch`
`to default levels? (SysLog, Presets, Modules & Output Devices)`
`Proceed with log defaults reset? [N]`

logSyslogConfig

Description   Configure the syslog host IP address.

NOTE: The device can forward its log messages to a syslog host if configured. This command allows a user to configure the host and port to send messages to and the facility to use in the messages. The facility value assignments are:

0 kern

1 user

2 mail

3 daemon

4 auth

5 syslog

6 lpr

7 news

8 uucp

9 cron

10 authpriv

11 ftp

12 ntp

13 audit

14 alert

15 clock

16 local0

17 local1

18 local2

19 local3

20 local4

21 local5

22 local6

23 local7

Additional configuration may be necessary to fully configure the log system.

Syntax   `logSyslogConfig [-h <hostname> or <ip_address>] [-p <port>] [-f <facility>] [-m <mode>]`

Operands   `-h <hostname> or <ip_address>`   set the syslog server hostname or IP address.

`-p <port>`   The host port number that the syslog server is listening on.

`-f <facility>`   The syslog facility to use in the messages.

`-m <mode>`   Determines whether syslog is to be put into a special OEM mode.

Example   `-> logSyslogConfig -h 172.26.0.202`
`Successfully configured the syslog host`

logShowConfig

Description Displays the current log configuration.

Syntax `logShowConfig`

Operands None.

Example `-> logShowConfig`

```
Log Configuration for: HP BLc Intel 4X QDR IB Switch
-----------------------------
Configurable devices
index : name      : |D|F|E|A|W|P|C|I|P|N|1|2|3|4|5|
----------------------------------------------------
  1   : Ram       : |X|X|X|X|X| | | | |X| | | | | |
  3   : Console   : |X|X|X|X|X| | | | |X| | | | | |
  5   : Syslog    : |X|X|X|X|X| | | | |X| | | | | |

Configurable presets
index : name      : state
-----------------------------
  1   : Dump      : Enabled
  2   : Fatal     : Enabled
  3   : Error     : Enabled
  4   : Alarm     : Enabled
  5   : Warning   : Enabled
  6   : Partial   : Disabled
  7   : Config    : Disabled
  8   : Info      : Disabled
  9   : Periodic  : Disabled
  15  : Notice    : Enabled
  10  : Debug1    : Disabled
  11  : Debug2    : Disabled
  12  : Debug3    : Disabled
  13  : Debug4    : Disabled
  14  : Debug5    : Disabled
```

logSyslogTest

Description Tests the Syslog configuration.

NOTE: This command tests the Syslog configuration by sending CSM message to registered Syslog servers.

Syntax `logSyslogTest {-e | -w | -n}`

Operands `-e`   Send Error severity CSM test message to Syslog.

`-w`   Send Warning severity CSM test message to Syslog.

`-n`   Send Notice severity CSM test message to Syslog.

Example `-> logSyslogTest -e`
`Currently configured Syslog host is: 0.0.0.0 port 514 facility 22`
`Syslog configuration has been tested`

8 Key Management

showKeys

Description   Displays the list of installed license keys.

NOTE: License keys unlock various software features of the product.

Syntax   `showKeys`

Operands   None.

Example
```
-> showKeys
-------------------------------------------------------
Key number:  1
Key:         XXXXX-2V9H6F-946QS3-3SCEV5-YZMZ5R-X
Description: License Key
Status:      Active
```

addKey

Description	Install a license key.
Syntax	`addKey <key>`
Operands	`<key>` The license key string to add.
Example	`-> addKey XXxxx-YYYYY-ZZZZZ-11111-222222-3`

removeKey

Description Removes a license key.

> **NOTE:** After entering this command a list of available keys will be displayed. The keys are shown along with an associated number. At the prompt, enter the number of the key you wish to remove.

Syntax `removeKey [<index>]`

Operands `<index>`   Index of license key to remove.

Example
```
-> removeKey

-------------------------------------------------
Key number:  1
Key:         XX7P91-XXXXX-946QS3-3SCEV5-XXXXX-X
Description: License Key
Status:      Active

Please enter the key number you wish to remove:
```

9 IbSwitchInfo

ismPortStats

Description Displays link error information associated with each switch port. These statistics include errors, dropped packets, discarded packets, and invalid packets.

NOTE: The Port statistic descriptions are as follows:
- PhysState   - The physical state of the port.
- PortState   - The state of the link on this port.
- LinkWidth   - The currently active link width on this port.
- LinkSpeed   - The currently active link speed on this port. Port speed is LinkWidth * LinkSpeed.
- SymblErrors - The number of times a 8B10B encoding violation, or a disparity violation was detected. If multiple errors are detected simultaneously (in more than one lane), the counter only increments by one.
- ErrRecovery - The number of times the link error recovery process happened successfully.
- LinkDowned - The number of times the link error recovery process failed.
- RcvErrors   - Number of errors received on the port.
- RmtPhysErr - Number of remote physical errors received on the port.
- TxDiscards - Number of port transmit discards.
- InPKeyViol - Number of times PKey inbound invalid.
- OutPKeyViol - Number of times PKey outbound invalid.
- InRawViol   - Number of times raw inbound packet discarded.
- OutRawViol - Number of times raw outbound packet was discarded.
- LLIntegrity - Number of local link integrity errors.
- ExcesBufOvr - Number of excessive buffer overrun errors.

Syntax `ismPortStats [-clear] [-noprompt] [-cols <columns>] [-port <port>]`

Operands

`-clear`   Clear the statistics. The statistics are first displayed, then cleared.

`-noprompt`   Don't prompt to continue output.

`-cols <columns>`   Set the number of columns to display per line.

`-port <port>`   Specify a port to display.

ismPortStats (continued)

Example

```
-> ismPortStats
Cable Ports:
Name            Cable01      Cable02      Cable03      Cable04
PhysState          Pol          Pol          Pol          Pol
PortState          Dwn          Dwn          Dwn          Dwn
LinkWidth           4X           4X           4X           4X
LinkSpeed      10.0Gbps     10.0Gbps     10.0Gbps     10.0Gbps
SymblErrors          0            0            0            0
ErrRecovery          0            0            0            0
LinkDowned           0            0            0            0
RcvErrors            0            0            0            0
RmtPhysErr           0            0            0            0
TxDiscards           0            0            0            0
InPKeyViol           0            0            0            0
OutPKeyViol          0            0            0            0
InRawViol            0            0            0            0
OutRawViol           0            0            0            0
LLIntegrity          0            0            0            0
ExcesBufOvr          0            0            0            0
Continue? [Y]

Name            Cable05      Cable06      Cable07      Cable08
PhysState          Pol          Pol          Pol          Pol
PortState          Dwn          Dwn          Dwn          Dwn
LinkWidth           4X           4X           4X           4X
LinkSpeed      10.0Gbps     10.0Gbps     10.0Gbps     10.0Gbps
SymblErrors          0            0            0            0
ErrRecovery          0            0            0            0
LinkDowned           0            0            0            0
RcvErrors            0            0            0            0
RmtPhysErr           0            0            0            0
TxDiscards           0            0            0            0
InPKeyViol           0            0            0            0
OutPKeyViol          0            0            0            0
InRawViol            0            0            0            0
OutRawViol           0            0            0            0
LLIntegrity          0            0            0            0
ExcesBufOvr          0            0            0            0
Continue? [Y]
```

ismPortCounters

Description Displays a table comparison of transmit, receive and error counters corresponding to each port of the switch. Optionally displays link error statistics associated with each port of the switch. These statistics include errors, dropped packets, discarded packets, and invalid packets.

NOTE:

Port counter descriptions:
- Transmit, Packets - The number of packets transmitted by the port, not including flow control packets.
- Transmit, Words - The number of data words transmitted by the port, not including flow control and VCRC data.
- Transmit, Discard - The number of transmit packets discarded by the port due to congestion or errors.
- Receive, Packets - The number of data packets received by the port, not including flow control packets.
- Receive, Words - The number of data words received by the port, not including flow control and VCRC data.
- Symbol, Errors - Number of symbol errors received on the port.
- Active, Speed - The active link speed of the port.
- Active, Width - The active link width of the port.
- Active, Potential - The utilization the port based on its maximum supported link speed and maximum supported link width.

Optional Statistics
- PhysState - The physical state of the port.
- PortState - The state of the link on this port.
- LinkWidth - The currently active link width on this port.
- LinkSpeed - The currently active link speed on this port.
- SymblErrors - The number of 8B10B encoding or disparity violations.
- ErrRecovery - The number of link error recovery process successes.
- LinkDowned - The number of link error recovery process failures.
- RcvErrors - Number of errors received on the port.
- RmtPhysErr - Number of remote physical errors received on the port.
- TxDiscards - Number of port transmit discards.
- InPKeyViol - Number of times PKey inbound invalid.
- OutPKeyViol - Number of times PKey outbound invalid.
- InRawViol - Number of times raw inbound packet discarded.
- OutRawViol - Number of times raw outbound packet was discarded.
- LLIntegrity - Number of local link integrity errors.
- ExcesBufOvr - Number of excessive buffer overrun errors.

The user is prompted to continue the output after each group of 24 ports are displayed. This command is best displayed with a terminal width of at least 120 columns.

Syntax `ismPortCounters [-clear] [-active] [-errors] [-potential | -stats] [-noprompt]`

Operands `-clear`   Clear the counters, the counters are first displayed, then cleared.

`-active`   Only show counters for ports in the active state.

ismPortCounters (continued)

-errors	Only show counters for ports with receive symbol errors.	
-potential	Only show counters for ports with active link or width under their maximum supported value.	
-stats	Show the optional link error counters associated with each switch port.	
-noprompt	Don't prompt to continue output.	

Example
```
-> ismPortCounters -errors
No ports with symbol errors.
```

ismLinearFwdb

Description Displays or sets Linear Forwarding Database for switch.

Syntax `ismLinearFwdb [<switch>]`

Operands `<switch>`   Switch index.

Example `-> ismLinearFwdb`

```
Switch Switch 1 Linear Fwdb (LFTTOP = 0x1):
 LID :: Port
0001      0 ( )
```

ismMultiFwdb

Description Displays the Multicast Forwarding Database for switch.

NOTE: This command is best displayed with a terminal width of at least 120 columns.

Syntax ismMultiFwdb <switch>

Operands <switch>   Switch index.

Example -> ismMultiFwdb

```
Switch Switch 1 Multicast Fwdb:
No entries found.
```

ismAutoClearConf

Description Configure Port stat counter auto-clear options.

NOTE: This feature will log every time a counter has reached its maximum capacity. This may be useful for diagnostics purposes, specifically for bad cables.

This feature is only available on certain switch hardware platforms.

Syntax `ismAutoClearConf {enable | disable} [warn] [log_first]`

Operands `{enable | disable}`   Enable or disable the auto-clear feature

   `warn`   Generate warning log messages instead of the default info messages.

   `log_first`   Log first clear (otherwise the first clear is not logged).

Example `-> ismAutoClearConf enable`
`Auto clear is enabled`

ismPortSetWidth

Description Displays or sets the supported link width for an IB port.

> **NOTE:** This command ONLY works on platforms which support 8X link aggregation. This command will set link aggregation so that switching will be at the specified width. Calling this function with only the <portName> will display its current values.
>
> Calling this function with the -bounce option disruptively brings active links down and back up so the links use the new setting immediately. Otherwise the new setting will be used the next time links retrain.
>
> The set of valid <portName>(s) depends on the chassis type. Use the command ismPortStats to see the valid <portName>(s).
>
> On non-12800 systems, an example cable <portName> is: Cable01 (i.e. Cable Port 1)
>
> On 12800 systems, an example cable <portName> is: L101P01 (i.e. Leaf 101 Port 1)
>
> On 12800 systems, an example interswitch link name is: S105AP18L104P36 (i.e. Spine 105 chip A port 18).

Syntax `ismPortSetWidth <portName> [<linkWidth>] [-bounce] [-verbose]`

Operands

`<portName>`	See the NOTE for this command.
`<linkWidth>`	1=1X, 2=4X, 3=1X/4X, 4=8X, 5=1X/8X, 6=4X/8X, 7=1X/4X/8X
`-bounce`	Brings active links down and back up if a new value is set.
`-verbose`	Verbose output mode.

Example
```
-> ismPortSetWidth Cable01
Cable01 link width supported: 4X
```

ismChassisSetWidth

Description   Displays or sets the supported link width for all chassis IB ports.

NOTE: This command ONLY works on chassis' which support 8X link aggregation. This command will set link aggregation so that Internal AND External switching will be at the specified width. Each 8X port is 2 aggregated 4X ports (i.e. a DUO), where one will be reported as a 8X port, and the other port is reported as disabled. Each of the two aggregated 4X ports must be connected to another enabled 8X Duo to establish a 8X link. To force links to only operate at 8X, use the '8X' option.

Calling this function without a parameter will display the ports' current values.

Calling this function with the -bounce option disruptively brings active links down and back up so the links use the new setting immediately. Otherwise the new setting will be used the next time links retrain.

Syntax   `ismChassisSetWidth [<linkWidth>] [-bounce] [-verbose]`

Operands   `<linkWidth>`   1=1X, 2=4X, 3=1X/4X, 4=8X, 5=1X/8X, 6=4X/8X, 7=1X/4X/8X

   `-bounce`   Brings active links down and back up if a new value is set.

   `-verbose`   Verbose output mode.

ismModuleSetWidth

Description Displays or sets the supported link width for all module IB ports.

NOTE: This command only works on modules that support 8X link aggregation. This command will set link aggregation so that Cable switching will be at the specified width. Each 8X port is 2 aggregated 4X ports (i.e. a Duo), where one will be reported as an 8X port, and the other port is reported as disabled. Each of the two aggregated 4X ports must be connected to another enabled 8X Duo to establish a 8X link. To force links to only operate at 8X, use the `8X` option.

Calling this function without a parameter will display the ports' current values.

Calling this function with the -bounce option disruptively brings active links down and back up so the links use the new setting immediately. Otherwise the new setting will be used the next time links retrain.

Syntax `ismModuleSetWidth [linkWidth] [-bounce]`

Operands `linkWidth`   1=1X, 2=4X, 3=1X/4X, 4=8X, 5=1X/8X, 6=4X/8X, 7=1X/4X/8X

`-bounce`   Brings active links down and back up if a new value is set.

Example `-> ismModuleSetWidth 7`

ismPortSet12x

Description Displays or sets the supported link width for an IB port.

NOTE: This command is included for backwards compatibility. The `ismPortSetWidth` command is prefered. This command only works on platforms that support 8X link aggregation. To force links to only operate at 8X, use the 8X only option.

Calling this command without the enable8xBit will display the ports' current values.

Calling this function with the `-bounce` option disruptively brings active links down and back up so the links use the new setting immediately. Otherwise the new setting will be used the next time links retrain.

The set of valid <portName>(s) depends on the chassis type. Use the command ismPortStats to see the valid <portName>(s).

On non-12800 systems, an example cable <portName> is: Cable01 (i.e. Cable Port 1)

On 12800 systems, an example cable <portName> is: L101P01 (i.e. Leaf 101 Port 1)

On 12800 systems, an example interswitch link name is: S105AP18L104P36 (i.e. Spine 105 chip A port 18).

Syntax `ismPortSet12x <portName> [<enable>] [-bounce]`

Operands `<portName>`   See the NOTE for this command.

`<enable>`   2=enable(8X only) 1=enable(AUTO 1X/4X/8X), 0=disable

`-bounce`   Brings active links down and back up if a new value is set.

Example `-> ismPortSet12x Cable01, 2`

ismChassisSet12x

Description Displays or sets the supported link width for all chassis IB ports.

NOTE: This command is included for backwards compatibility.

The `ismChassisSetWidth` command is preferred.

This command only works on chassis' which support 8X link aggregation.

This command will enable 8X link aggregation so that internal and external switching will be at 8X. Each 8X port is 2 aggregated 4X ports (i.e. a DUO), where one will be reported as an 8x port, and the other port is reported as disabled. Each of the two aggregated 4x ports must be connected to another enabled 8X Duo to establish an 8X link. To force links to only operate at 8X, use the '8X only' option.

Calling this command without the `enable8xBit` option will display the ports' current values.

Calling this function with the `-bounce` option disruptively brings active links down and back up so the links use the new setting immediately. Otherwise the new setting will be used the next time links retrain.

Syntax `ismChassisSet12x [<enable8xBit>] [-bounce]`

Operands `<enable12xBit>`   2=enable(8X only) 1=enable(AUTO 1X/4X/8X), 0=disable.

`-bounce`   Brings active links down and back up if a new value is set.

Example `->  ismChassisSet12x`
```
Cable01 link width supported: 8X
Cable02 link width supported: 4X
Cable03 link width supported: 4X
Cable04 link width supported: 4X
Cable05 link width supported: 4X
Cable06 link width supported: 4X
Cable07 link width supported: 4X
Cable08 link width supported: 4X
Cable09 link width supported: 4X
Cable10 link width supported: 4X
Cable11 link width supported: 4X
Cable12 link width supported: 4X
Cable13 link width supported: 4X
Cable14 link width supported: 4X
Cable15 link width supported: 4X
Cable16 link width supported: 4X
Bay01 link width supported: 4X
Bay02 link width supported: 4X
Bay03 link width supported: 4X
Bay04 link width supported: 4X
Bay05 link width supported: 4X
Bay06 link width supported: 4X
Bay07 link width supported: 4X
Bay08 link width supported: 4X
Bay09 link width supported: 4X
Bay10 link width supported: 4X
Bay11 link width supported: 4X
Bay12 link width supported: 4X
Bay13 link width supported: 4X
Bay14 link width supported: 4X
Bay15 link width supported: 4X
Bay16 link width supported: 4X
```

ismChassisSetMtu

Description Displays or sets MTU and VL capabilities all chassis IB ports.

NOTE: This command will set the mtu capability for both Internal AND External switching. If the {vlCap} is not specified it defaults to the maximum VL(s) for the selected mtuCap. Calling this function without an mtuCap designator will display the current value for each port.

Calling this function with the -bounce option disruptively brings active links down and back up so the links use the new setting immediately. Otherwise the new setting will be used the next time links retrain.

Syntax `ismChassisSetMtu [<mtuCap> [<vlCap>]] [-bounce]`

Operands `<mtuCap>`   4=2048 bytes, 5=4096 bytes.

`<vlCap>`   1=VL0, 2=VL0-VL1, 3=VL0-VL3, 4=VL0-VL7

`-bounce`   Brings active links down and back up if a new value is set.

ismModuleSetMtu

Description   Displays or sets MTU and VL capabilities all module IB ports.

NOTE: This command will set the MTU capability for both cable and bay switching. Calling this function without an `mtuCap` designator will display the current value for each port.

Calling this function with the `-bounce` option disruptively brings active links down and back up so the links use the new setting immediately. Otherwise the new setting will be used the next time links retrain.

Syntax   `ismModuleSetMtu [<mtuCap> [<vlCap>]] [-bounce]`

Operands   `<mtuCap>`   4=2048 bytes, 5=4096 bytes.

   `<vlCap>`   1=VL0, 2=VL0,VL1, 3=VL0-VL3, 4=VL0-VL7

   `-bounce`   Brings active links down and back up if a new value is set.

Example
```
-> ismModuleSetMtu
Cable01 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Cable02 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Cable03 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Cable04 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Cable05 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Cable06 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Cable07 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Cable08 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Cable09 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Cable10 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Cable11 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Cable12 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Cable13 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Cable14 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Cable15 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Cable16 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Bay01 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Bay02 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Bay03 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Bay04 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Bay05 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Bay06 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Bay07 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Bay08 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Bay09 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Bay10 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Bay11 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Bay12 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Bay13 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Bay14 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Bay15 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
Bay16 MTUCap=4(2048 bytes) VLCap=4(8 VLs)
```

ismChassisSetBCDM

Description Displays or sets the Buffer Credit Distribution Method for all chassis IB ports.

NOTE: Calling this function without an <Method> designator will display the current value for each port. The default Buffer Credit Distribution Method is 4.

Calling this function with the `-bounce` option disruptively brings active links down and back up so the links use the new setting immediately. Otherwise the new setting will be used the next time links retrain.

Syntax `ismChassisSetBCDM [<Method>] [-bounce]`

Operands `<Method>`   Buffer Credit Distribution Method 0-4.

`-bounce`   Brings active links down and back up if a new value is set.

ismPortEnable

Description Enables an IB port.

NOTE: The set of valid <portName>(s) depends on the chassis type. Use the command ismPortStats to see the valid <portName>(s).

On non-12800 systems, an example cable <portName> is: Cable01 (i.e. Cable Port 1)

On 12800 systems, an example cable <portName> is: L101P01 (i.e. Leaf 101 Port 1)

On 12800 systems, an example interswitch link name is: S105AP18L104P36 (i.e. Spine 105 chip A port 18).

Syntax `ismPortEnable <portName> [-verbose]`

Operands `<portname>`   See the NOTE for this command.

`-verbose`   Verbose output mode.

Example `-> ismPortEnable Cable01`

ismChassisSetEnable

Description Displays or modifies the port enable setting for each IB port in the chassis.

NOTE: Calling this function without an enable will display ports current values.
When disabling ports only cable ports will be disabled.

Syntax `ismChassisSetEnable [{0 | 1}]`

Operands `{0 | 1}`   Values: 0=disable, 1=enable.

ismModuleSetEnable

Description Displays or modifies the port enable setting for each IB port on the module.

NOTE: Calling this function without an enable will display ports current values.

Syntax `ismModuleSetEnable [{0 | 1}]`

Operands `{0 | 1}`   Values: 0=disable, 1=enable

Example
```
-> ismModuleSetEnable
Cable01 is ENABLED
Cable02 is ENABLED
Cable03 is ENABLED
Cable04 is ENABLED
Cable05 is ENABLED
Cable06 is ENABLED
Cable07 is ENABLED
Cable08 is ENABLED
Cable09 is ENABLED
Cable10 is ENABLED
Cable11 is ENABLED
Cable12 is ENABLED
Cable13 is ENABLED
Cable14 is ENABLED
Cable15 is ENABLED
Cable16 is ENABLED
Bay01 is ENABLED
Bay02 is ENABLED
Bay03 is ENABLED
Bay04 is ENABLED
Bay05 is ENABLED
Bay06 is ENABLED
Bay07 is ENABLED
Bay08 is ENABLED
Bay09 is ENABLED
Bay10 is ENABLED
Bay11 is ENABLED
Bay12 is ENABLED
Bay13 is ENABLED
Bay14 is ENABLED
Bay15 is ENABLED
Bay16 is ENABLED
```

ismPortDisable

Description   Disables an IB port.

NOTE: The set of valid <portName>(s) depends on the chassis type. Use the command ismPortStats to see the valid <portName>(s).

On non-12800 systems, an example cable <portName> is: Cable01 (i.e. Cable Port 1)

On 12800 systems, an example cable <portName> is: L101P01 (i.e. Leaf 101 Port 1)

On 12800 systems, an example interswitch link name is: S105AP18L104P36 (i.e. Spine 105 chip A port 18).

Syntax   `ismPortDisable <portName> [-verbose]`

Operands   `<portname>`   See the NOTE for this command.

`-verbose`   Verbose output mode.

Example   `-> ismPortDisable Cable01`

ismChassisSetSpeed

Description Displays or sets the support link speeds for all IB ports in the chassis.

NOTE: This command can be used to set the supported link speed on each of the chassis ports. This command does not cause the associated ports to change the active or enabled speed of the port. The active and enabled speed is changed the next time the port transitions down and back up. Each external port must be connected to another port similarly configured to establish a link.

Calling this function without a speed designator will display the ports' current values.

Calling this function with the `-bounce` option disruptively brings active links down and back up so the links use the new setting immediately. Otherwise the new setting will be used the next time links retrain.

Syntax `ismChassisSetSpeed [<linkSpeed>] [-bounce]`

Operands `<linkSpeed>`   1=SDR 2=DDR 3=SDR/DDR 4=QDR 5=SDR/QDR 6=DDR/QDR 7=SDR/DDR/QDR

`-bounce`   Brings active links down and back up if a new value is set.

ismModuleSetSpeed

Description Displays or sets the support link speeds for all IB ports in the module.

NOTE: DDR and QDR links are only available on modules that support these links.

This command will set link speeds for both cable and bay switching. Each cable port must be connected to another port similarly configured to establish a link.

Calling this function without a speed designator will display the ports' current values.

Calling this function with the [-bounce] option disruptively brings active links down and back up so the links use the new setting immediately. Otherwise the new setting will be used the next time links retrain.

Syntax `ismModuleSetSpeed [<linkSpeed>] [-bounce]`

Operands `<linkSpeed>` 1=SDR 2=DDR 3=SDR/DDR 4=QDR 5=SDR/QDR 6=DDR/QDR 7=SDR/DDR/QDR

`-bounce` Brings active links down and back up if a new value is set.

Example
```
-> ismModuleSetSpeed
Cable01 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Cable02 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Cable03 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Cable04 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Cable05 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Cable06 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Cable07 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Cable08 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Cable09 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Cable10 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Cable11 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Cable12 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Cable13 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Cable14 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Cable15 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Cable16 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay01 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay02 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay03 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay04 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay05 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay06 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay07 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay08 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay09 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay10 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay11 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay12 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay13 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay14 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay15 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay16 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
```

ismMidplaneSetSpeed

Description Displays or sets the support link speeds for all bay ports.

NOTE: This command only works on modules that support DDR or QDR links. This command will set link speeds so that bay switching will be at the specified speed.

Calling this function without a parameter will display the ports' current values.

Calling this function with the `-bounce` option disruptively brings active links down and back up so the links use the new setting immediately. Otherwise the new setting will be used the next time links retrain.

Syntax `ismMidplaneSetSpeed [<linkSpeed>] [-bounce]`

Operands `<linkSpeed>`   1=SDR 2=DDR 3=SDR/DDR 4=QDR 5=SDR/QDR 6=DDR/QDR 7=SDR/DDR/QDR

          `-bounce`   Brings active links down and back up if a new value is set.

Example
```
-> ismMidplaneSetSpeed
Bay01 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay02 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay03 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay04 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay05 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay06 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay07 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay08 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay09 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay10 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay11 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay12 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay13 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay14 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay15 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
Bay16 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE
```

ismPortSetSpeed

Description Displays or sets the speed of an IB port.

NOTE: This command can be used to set the supported link speed on the port.

This command does not cause the port to change the active or enabled speed of the port. The active and enabled speed is changed the next time the port transitions down and back up.

Calling this function with only the `<portName>` will display its current values.

Calling this function with the `-bounce` option disruptively brings active links down and back up so the links use the new setting immediately. Otherwise the new setting will be used the next time links retrain.

The set of valid `<portName>`(s) depends on the chassis type. Use the command ismPortStats to see the valid `<portName>`(s).

On non-12800 systems, an example cable `<portName>` is: Cable01 (i.e. Cable Port 1)

On 12800 systems, an example cable `<portName>` is: L101P01 (i.e. Leaf 101 Port 1)

On 12800 systems, an example interswitch link name is: S105AP18L104P36 (i.e. Spine 105 chip A port 18).

Syntax `ismPortSetSpeed <portName> [<linkSpeed>] [-bounce]`

Operands `<portName>`   See the NOTE for this command.

`<linkSpeed>`   1=SDR 2=DDR 3=SDR/DDR 4=QDR 5=SDR/QDR 6=DDR/QDR 7=SDR/DDR/QDR

`-bounce`   Brings active links down and back up if a new value is set.

Example `-> ismPortSetSpeed Cable01`
`Cable01 link speed supported is SDR/DDR/QDR AUTO_NEGOTIATE`

ismPortSetBeacon

Description   Displays or modifying the LED beaconing indicator for an IB port.

NOTE: REBOOT of this chassis is REQUIRED to activate changes.

double-quotes (e.g., `ismPortSetBeacon "Cable 1"`). Using this command with only the `port` option displays its current values. A reboot is required to activate changes made with this command.

Syntax   `ismPortSetBeacon <portName> [{0 | 1}] [-verbose]`

Operands   `<portName>`   Port name.

`{0 | 1}`   `<enable>` values: 0=disable, 1=enable.

`-verbose`   Verbose output mode.

Example   `-> ismPortSetBeacon Cable01`
`Cable01 beacon is off`

ismPortQsfpInfo

Description   Displays information from the installed QSFPs.

NOTE: QSFP information may not be available for DDR hybrid cable assemblies. For 0.5 meter cables, the length may be reported as "0 meter".

Syntax   `ismPortQsfpInfo [{all | L<xxx> | L<xxx>P<yy> | Cable<yy>]} [-verbose]`

Operands   `all`   Display info for all installed QSFPs in chassis.

`L<xxx>`   Display info for all installed QSFPs on leaf xxx.

`L<xxx>P<yy>`   Display info for QSFP installed on leaf <xxx>, port <yy>.

`Cable<yy>`   display info for QSFP installed on port yy (Viper, Edge only).

`-verbose`   Verbose output mode.

ismChassisBounce

Description Bounce enabled ports for entire chassis.

Syntax `ismChassisBounce [-verbose]`

Operands `-verbose`   Verbose output mode.

ismModuleBounce

Description Bounce enabled ports for entire module.

Syntax `ismModuleBounce [-verbose]`

Operands `-verbose` Verbose output mode.

ismMidplaneBounce

Description Bounce enabled bay ports only.

Syntax `ismMidplaneBounce [-verbose]`

Operands `-verbose`   Verbose output mode.

ismPortBounce

Description Bounce port if enabled.

NOTE: The set of valid <portName>(s) depends on the chassis type. Use the command ismPortStats to see the valid <portName>(s).

On non-12800 systems, an example cable <portName> is: Cable01 (i.e. Cable Port 1)

On 12800 systems, an example cable <portName> is: L101P01 (i.e. Leaf 101 Port 1)

On 12800 systems, an example interswitch link name is: S105AP18L104P36 (i.e. Spine 105 chip A port 18).

Syntax `ismPortBounce <portName> [-verbose]`

Operands `<portName>`   See the NOTE for this command.

`-verbose`   Verbose output mode.

ismShowPStatThresh

Description Displays the port statistics thresholds.

NOTE: The following fields are the available port statics thresholds:

- `portXmitData`: 32-bit data words transmitted
- `portRecvData`: 32-bit data words received
- `portXmitPkts`: data packets transmitted
- `portRecvPkts`: data packets received
- `portSymbolErr`: a 8B10B encoding violation, or a displarity violation was detected
- `portLinkErrRecv`: link error recovery process happened successfully
- `portLinkDowned`: link error recovery process failed
- `portRecvErr`: errors received
- `portRecvRemPhysErr`: remote physical errors received
- `portXmitDiscard`: port transmit discards
- `portPKeyViolIn`: PKey inbound was invalid
- `portPKeyViolOut`: PKey outbound was invalid
- `portRawViolIn`: raw inbound packet discarded
- `portRawViolOut`: raw outbound packet discarded
- `portLocalLinkInteg`: link integrity errors
- `portExcBufferOverrun`: excessive buffer overrun errors
- `portRelayedVL15Dropped`: remote VL15 packet was dropped
- `portLocalVL15Dropped`: local VL15 packet was dropped
- `portNonSMPDropped`: non SMP packet was dropped

Syntax `ismShowPStatThresh`

Operands None.

Example
```
-> ismShowPStatThresh
            Field          Threshold  Time Unit
            -----          ---------  ---------
    1          portXmitData:      0   (Percent of Max)
    2          portRecvData:      0   (Percent of Max)
    3          portXmitPkts:      0   (Percent of Max)
    4          portRecvPkts:      0   (Percent of Max)
    5          portXmitWait:      0   (1 Second)
    6         portSymbolErr:      0   (1 Second)
    7       portLinkErrRecv:      0   (1 Second)
    8        portLinkDowned:      0   (1 Second)
    9           portRecvErr:      0   (1 Second)
   10    portRecvRemPhysErr:      0   (1 Second)
   12       portXmitDiscard:      0   (1 Second)
   13        portPKeyViolIn:      0   (1 Second)
   14       portPKeyViolOut:      0   (1 Second)
   15         portRawViolIn:      0   (1 Second)
   16        portRawViolOut:      0   (1 Second)
   17    portLocalLinkInteg:      0   (1 Second)
   18  portExcBufferOverrun:      0   (1 Second)
   19 portRelayedVL15Dropped:     0   (1 Second)
   20    portLocalVL15Dropped:    0   (1 Second)
   21       portNonSMPDropped:    0   (1 Second)
```

ismSetPStatThresh

Description Changes the port statistics thresholds.

> **NOTE:** The available thresholds are:
> - `portXmitData`
> - `portRecvData`
> - `portXmitPkts`
> - `portRecvPkts`
> - `portSymbolErr`
> - `portLinkErrRecv`
> - `portLinkDowned`
> - `portRecvErr`
> - `portRecvRemPhysErr`
> - `portXmitDiscard`
> - `portPKeyViolIn`
> - `portPKeyViolOut`
> - `portRawViolIn`
> - `portRawViolOut`
> - `portLocalLinkInteg`
> - `portExcBufferOverrun`
> - `portRelayedVL15Dropped`
> - `portLocalVL15Dropped`
> - `portNonSMPDropped`

Syntax `ismSetPStatThresh <field> <threshold>`

Operands `<field>`   The name of the port status threshold field, see NOTE.

`<threshold>`   The numeric threshold value.

Example `-> ismSetPStatThresh  portXmitData 0`
`successfully set port stat threshold portXmitDat`

ismRemoveStateDump

Description Removes switch ASIC state dump files.

Syntax `ismRemoveStateDump`

Operands None.

Example `-> ismRemoveStateDump`

ismShowStateDump

Description Displays the switch ASIC state dump files.

Syntax `ismShowStateDump`

Operands None.

Example `-> ismShowStateDump`

10 Time Management

time

Description Configure the time on the device.

NOTE: Time can be configured locally (using a local clock) or set to be updated by an SNTP server. Please note that if you set the time locally, the unit will unconfigure the NTP server IP address if set. If no options are passed in, the current system time will be printed out. NTP server hostname length is limited to 64 characters.

Syntax `time [{-S <hostname or IP address> | -T <hhmmss>[<mmddyyyy>]}]`

Operands `-S <hostname or IP address>` NTP server hostname or IP address.

`-T <hhmmss>[<mmddyyyy>]` Set the local clock time hour minutes seconds [month day year].

Example `-> time`
`10:34:32  07/30/2009`
`Configured to use NTP server IP address: 10.20.33.35`

timeZoneConf

Description Displays and configures the time zone setting.

NOTE: The offset parameter specifies a time zone the system should use when setting the time. In the U.S. the following time zones are in effect: Eastern Standard Time = GMT -5; Central Standard Time = GMT -6; Mountain Standard Time = GMT -7; Pacific Standard Time = GMT -8. GMT = Greenwich Mean Time.

Syntax `timeZoneConf [<offset>]`

Operands `<offset>`   Time offset in relation to GMT.

Example `-> timeZoneConf -5`
`Timezone offset successfully configured`
`Current time zone offset is: -5`

timeDSTConf

Description Configure the daylight savings time settings.

> **NOTE:** To set the daylight savings time to start on the 1st Sunday of April and end on the 4th Sunday of October the command would be: `'timeDSTConf 1 1 4 4 1 10'`.

Syntax `timeDSTConf [<sw> <sd> <sm> <ew> <ed> <em>]`

Operands

`<sw>`	Start week, valid values: 1 = 1st, 2 = 2nd, 3 = 3rd, 4 = 4th, 5 = 5th
`<sd>`	Start day, valid values: 1 = Sunday, 2 = Monday, 3 = Tuesday, 4 = Wednesday, 5 = Thursday, 6 = Friday, 7 = Saturday
`<sm>`	Start month, valid values: 3 = March, 4 = April, 5 = May, 6 = June, 7 = July, 8 = August, 9 = September, 10 = October, 11 = November
`<ew>`	End week, valid values: 1 = 1st, 2 = 2nd, 3 = 3rd, 4 = 4th, 5 = 5th
`<ed>`	End day, valid values: 1 = Sunday, 2 = Monday, 3 = Tuesday, 4 = Wednesday, 5 = Thursday, 6 = Friday, 7 = Saturday
`<em>`	End month, valid values: 3 = March, 4 = April, 5 = May, 6 = June, 7 = July, 8 = August, 9 = September, 10 = October, 11 = November

Example
```
-> timeDSTConf 2 1 3 1 1 11
Timezone offset successfully configured
Current DST = Start: 2'nd Sunday of March End: 1'st Sunday of November
```

timeNtpTimeout

Description Configure the NTP timeout setting.

NOTE: With no arguments, this command displays the current NTP timeout settings. This is the amount of time (in seconds) for the system to wait for a response from the NTP server. This setting can be configured by using the same command with the new timeout value (in whole seconds) as the only argument. The default settings is 2 seconds. The NTP timeout value is not used on linecards or slave CMUs.

Syntax `timeNtpTimeout [<seconds>]`

Operands `<seconds>`   New timeout setting in seconds.

Example
```
-> timeNtpTimeout
Current NTP timeout value: 2 seconds
-> timeNtpTimeout 3
Current NTP timeout changed to 3 seconds
```

timeNtpRefreshTime

Description   Configure the delay between time updates via NTP.

NOTE: With no arguments this command displays the current NTP refresh time, which is the delay in seconds between attempts to sync the clock via NTP. This value can be configured by using the same command with the new refresh time (in whole seconds) as the only argument. The NTP refresh time is not used on linecards or slave CMUs.

Syntax   `timeNtpRefreshTime [<seconds>]`

Operands   `<seconds>`   Refresh delay in seconds.

Example   `-> timeNtpRefreshTime`
`Current NTP refresh delay value: 60 seconds`
`-> timeNtpRefreshTime 50`
`Current NTP refresh delay changed to 50 seconds.`

11 SNMP

snmpCommunityConf

Description Configure the SNMP community strings.

NOTE: If no options are entered the current SNMP read-only community string and read-write community string is displayed. If options are entered then the corresponding community string is changed.

Syntax snmpCommunityConf [-r <readonly_comm_str>] [-w <read_write_comm_strstring>]

Operands -r <readonly_comm_   A read-only community string.
str>

-w <read_write_   A read/write community string.
comm_strstring>

Example -> snmpCommunityConf -r public
Read Only Community String Was Set To: public

snmpTargetAddr

Description Configure the SNMP trap targets.

NOTE: The output is in the form: `mib : mib_object : table_index : value`. For more details on the `snmpTargetAddrTable`, see *SNMP-TARGET-MIB, RFC 2573*.

Syntax
```
snmpTargetAddr {show | add | edit | delete} -n [-a <address>] [-p <port>] [-t
<timeout>] [-r <retry_count>] [-l <tag_list>] [-v <parameters>] [-s
<storage_type>] [-i <status>]
```

Operands

`show`	Show the contents of the `snmpTargetAddrTable`.
`add`	Add a row to the `snmpTargetAddrTable`.
`edit`	Modify an existing row in the `snmpTargetAddrTable`.
`delete`	Remove an existing row of the `snmpTargetAddrTable`.
`-n`	Name. A unique name used to identify a row.
`-a <address>`	Address, target machine IP address in decimal form.
`-p <port>`	Port, the target port to send traps/informs.
`-t <timeout>`	Timeout, time to wait for an inform response.
`-r <retry_count>`	Retry count, number of re-send attempts for inform.
`-l <tag_list>`	Tag list, indicates the trap/informs that are sent.
`-v <parameters>`	Parameters, maps to an entry in the `snmpTargetAddrTable`.
`-s <storage_type>`	Storage type, determines whether the entry is saved in flash. Must be string 'volatile' or 'nonVolatile'.
`-i <status>`	Status, 1=Active, 2=Not In Service, 3=Not Ready.

snmpTargetAddr (continued)

Example
```
-> snmpTargetAddr
mib:mib_object                          : Name    : Value
-------------------------------------------------------------------------------
-
rfc2573t:snmpTargetAddrTDomain      : nms v1 : 1.3.6.1.6.1.1
rfc2573t:snmpTargetAddrTAddress     : nms v1 : IpAddr:0.0.0.0 Port:162
rfc2573t:snmpTargetAddrTimeout      : nms v1 : 1500
rfc2573t:snmpTargetAddrRetryCount   : nms v1 : 3
rfc2573t:snmpTargetAddrTagList      : nms v1 : rfc1493 rfc1757 rfc1907 rfc2233
tmscom
rfc2573t:snmpTargetAddrParams       : nms v1 : v1 params
rfc2573t:snmpTargetAddrStorageType  : nms v1 : nonVolatile
rfc2573t:snmpTargetAddrRowStatus    : nms v1 : Not Ready

rfc2573t:snmpTargetAddrTDomain      : nms v2 : 1.3.6.1.6.1.1
rfc2573t:snmpTargetAddrTAddress     : nms v2 : IpAddr:0.0.0.0 Port:162
rfc2573t:snmpTargetAddrTimeout      : nms v2 : 1500
rfc2573t:snmpTargetAddrRetryCount   : nms v2 : 3
rfc2573t:snmpTargetAddrTagList      : nms v2 : rfc1493 rfc1757 rfc1907 rfc2233
tmscom
rfc2573t:snmpTargetAddrParams       : nms v2 : v2 params
rfc2573t:snmpTargetAddrStorageType  : nms v2 : nonVolatile
rfc2573t:snmpTargetAddrRowStatus    : nms v2 : Not Ready

rfc2573t:snmpTargetAddrTDomain      : nms v3 : 1.3.6.1.6.1.1
rfc2573t:snmpTargetAddrTAddress     : nms v3 : IpAddr:0.0.0.0 Port:162
rfc2573t:snmpTargetAddrTimeout      : nms v3 : 1500
rfc2573t:snmpTargetAddrRetryCount   : nms v3 : 3
rfc2573t:snmpTargetAddrTagList      : nms v3 : rfc1493 rfc1757 rfc1907 rfc2233
tmscom
rfc2573t:snmpTargetAddrParams       : nms v3 : v3 params
rfc2573t:snmpTargetAddrStorageType  : nms v3 : nonVolatile
rfc2573t:snmpTargetAddrRowStatus    : nms v3 : Not Ready
```

snmpTargetParams

Description Access the snmpTargetParamsTable.

> **NOTE:** The output is in the form: `mib : mib_object : table_index : value`. For more details on the `snmpTargetParamsTable`, see *SNMP-TARGET-MIB, RFC-2573*.

Syntax `snmpTargetParams`

Operands None

Example
```
-> snmpTargetParams
rfc2573t:snmpTargetParamsMPModel: v1 params : 0
rfc2573t:snmpTargetParamsMPModel: v2 params : 1
rfc2573t:snmpTargetParamsMPModel: v3 params : 3
rfc2573t:snmpTargetParamsSecurityModel: v1 params : 1
rfc2573t:snmpTargetParamsSecurityModel: v2 params : 2
rfc2573t:snmpTargetParamsSecurityModel: v3 params : 3
rfc2573t:snmpTargetParamsSecurityName: v1 params : public
rfc2573t:snmpTargetParamsSecurityName: v2 params : public
rfc2573t:snmpTargetParamsSecurityName: v3 params : initialnone
rfc2573t:snmpTargetParamsSecurityLevel: v1 params : 1
rfc2573t:snmpTargetParamsSecurityLevel: v2 params : 1
rfc2573t:snmpTargetParamsSecurityLevel: v3 params : 1
rfc2573t:snmpTargetParamsStorageType: v1 params : 3
rfc2573t:snmpTargetParamsStorageType: v2 params : 3
rfc2573t:snmpTargetParamsStorageType: v3 params : 3
rfc2573t:snmpTargetParamsRowStatus: v1 params : 1
rfc2573t:snmpTargetParamsRowStatus: v2 params : 1
rfc2573t:snmpTargetParamsRowStatus: v3 params : 1
```

snmpNotifyProfile

Description Access the snmpNotifyFilterProfileTable.

NOTE: The output is in the form: `mib : mib_object : table_index : value`. For more details on the `snmpNotifyFilterProfileTable`, see *SNMP-NOTIFICATION-MIB, RFC-2573*.

Syntax `snmpNotifyProfile`

Operands None

Example
```
-> snmpNotifyProfile
rfc2573n:snmpNotifyFilterProfileName: v1 params : v1 params
rfc2573n:snmpNotifyFilterProfileName: v2 params : v2 params
rfc2573n:snmpNotifyFilterProfileName: v3 params : v3 params
rfc2573n:snmpNotifyFilterProfileStorType: v1 params : 3
rfc2573n:snmpNotifyFilterProfileStorType: v2 params : 3
rfc2573n:snmpNotifyFilterProfileStorType: v3 params : 3
rfc2573n:snmpNotifyFilterProfileRowStatus: v1 params : 1
rfc2573n:snmpNotifyFilterProfileRowStatus: v2 params : 1
rfc2573n:snmpNotifyFilterProfileRowStatus: v3 params : 1
```

snmpNotifyFilter

Description Access the snmpNotifyFilterTable.

NOTE: The output is in the form: `mib : mib_object : table_index : value`. For more details on the `snmpNotifyFilterTable`, see *SNMP-NOTIFICATION-MIB, RFC-2573*.

Syntax `snmpNotifyFilter`

Operands None

Example
```
-> snmpNotifyFilter
rfc2573n:snmpNotifyFilterMask:  v1 params : 0
rfc2573n:snmpNotifyFilterMask:  v2 params : 0
rfc2573n:snmpNotifyFilterMask:  v3 params : 0
rfc2573n:snmpNotifyFilterType:  v1 params : 1
rfc2573n:snmpNotifyFilterType:  v2 params : 1
rfc2573n:snmpNotifyFilterType:  v3 params : 1
rfc2573n:snmpNotifyFilterStorageType:   v1 params : 3
rfc2573n:snmpNotifyFilterStorageType:   v2 params : 3
rfc2573n:snmpNotifyFilterStorageType:   v3 params : 3
rfc2573n:snmpNotifyFilterRowStatus:     v1 params : 1
rfc2573n:snmpNotifyFilterRowStatus:     v2 params : 1
rfc2573n:snmpNotifyFilterRowStatus:     v3 params : 1
```

snmpNotify

Description Access the snmpNotifyTable.

> **NOTE:** The output is in the form: `mib : mib_object : table_index : value`. For more details on the `snmpNotifyTable`, see *SNMP-NOTIFICATION-MIB, RFC-2573*.

Syntax `snmpNotify`

Operands None.

Example
```
-> snmpNotify
rfc2573n:snmpNotifyTag: bridge : rfc1493
rfc2573n:snmpNotifyTag: interfaces : rfc2233
rfc2573n:snmpNotifyTag: rmon : rfc1757
rfc2573n:snmpNotifyTag: snmp : rfc1907
rfc2573n:snmpNotifyTag: tms : tmscom
rfc2573n:snmpNotifyType: bridge : 1
rfc2573n:snmpNotifyType: interfaces : 1
rfc2573n:snmpNotifyType: rmon : 1
rfc2573n:snmpNotifyType: snmp : 1
rfc2573n:snmpNotifyType: tms : 1
rfc2573n:snmpNotifyStorageType: bridge : 3
rfc2573n:snmpNotifyStorageType: interfaces : 3
rfc2573n:snmpNotifyStorageType: rmon : 3
rfc2573n:snmpNotifyStorageType: snmp : 3
rfc2573n:snmpNotifyStorageType: tms : 3
rfc2573n:snmpNotifyRowStatus: bridge : 1
rfc2573n:snmpNotifyRowStatus: interfaces : 1
rfc2573n:snmpNotifyRowStatus: rmon : 1
rfc2573n:snmpNotifyRowStatus: snmp : 1
rfc2573n:snmpNotifyRowStatus: tms : 1
```

snmpSystem

Description Configure the SNMP system info.

NOTE: The output is in the form: `mib : mib_object : table_index : value.`

Syntax `snmpSystem [show | edit] [-n <sysName>] [-c <sysContact>] [-l <sysLocation>]`

Operands

`show`	Show the contents of the `snmpTargetAddrTable`.
`edit`	Modify an existing row in the `snmpTargetAddrTable`
`-n` <sysName>	Specify system name information.
`-c` <sysContact>	Specify system contact information.
`-l` <sysLocation>	Specify system location information.

Example
```
-> snmpSystem
rfc1907:sysDescr: Ý : 20.28.4D.61.73.74.65.72.29
rfc1907:sysObjectID:  : 1.3.6.1.4.1.10222.7.1.2
rfc1907:sysUpTime:  : 0 Day(s), 18 Hour(s), 10 Minute(s), 44 Second(s)
rfc1907:sysContact: Ý : {no value}
rfc1907:sysName: Ý : 2D.32.35.33
rfc1907:sysLocation: Ý : Main Chassis Unit, Slot 253
rfc1907:sysServices:  : 79
```

snmpUsrSec

Description Configure SNMP V3 users.

NOTE: Handles configuration and display of SNMP v3 users. Supported authentication algorithms are: NONE, MD5, and SHA. A passphrase is required for all except the NONE algorithm. No privacy algorithms are currently supported.

Syntax `snmpUsrSec {add | show | edit | delete} -a {MD5 <key> | SHA <key> | NONE}`

Operands

add	Add an entry to the V3 user table.
show	Show entries in the V3 user table.
edit	Modify an entry in the V3 user table.
delete	Remove an entry in the V3 user table.
-a {MD5 <key>	SHA <key>	NONE}	Authentication algorithm and key. Options: MD5, SHA, NONE. Passphrase is required unless NONE is specified.

Example
```
-> snmpUsrSec
User     : initialmd5
Auth     : MD5
Auth Key: 0x047b473f93211a17813ce5fff290066b
Priv     : NONE

User     : initialsha
Auth     : SHA
Auth Key: 0x1c8cbd687fb0f0a22ddd24315db0d84c09eb5ff3
Priv     : NONE

User     : initialnone
Auth     : NONE
Priv     : NONE
```

12 CaptureInfo

capture

Description Displays all the information for this device.

Syntax `capture`

Operands None.

captureFw

Description   Displays all the firmware information for this device.

NOTE: This command is intended for support personnel.

Syntax   `captureFw`

Operands   None.

captureSm

Description Displays all the manager information for this device.

> **NOTE:** This command is intended for support personnel.

Syntax captureSm

Operands None

captureIsm

Description   Displays all the Infiniband switch information for this device.

NOTE: This command is intended for support personnel.

Syntax   captureIsm

Operands   None

captureChassis

Description Displays all the chassis information for this device.

NOTE: This command is intended for support personnel.

Syntax `captureChassis`

Operands None

captureNetwork

Description Displays all the network information for this device.

NOTE: This command is intended for support personnel.

Syntax `captureNetwork`

Operands None.

captureLog

Description   Displays all the log information for this device.

> **NOTE:**   This command is intended for support personnel.

Syntax   captureLog

Operands   None.

captureMisc

Description Displays all the miscellaneous information for this device.

NOTE: This command is intended for support personnel.

Syntax captureMisc

Operands None.

captureSnmp

Description Displays all the miscellaneous information for this device.

> **NOTE:** This command is intended for support personnel.

Syntax `captureSnmp`

Operands None.

captureShell

Description Displays the pertinent shell command information for this device.

NOTE: This command is intended for support personnel.

Syntax `captureShell`

Operands None.