



Broadcom NetXtreme Controller 5741x(Wh+) /5750x (Thor) /5760x (Thor2)

NICCLI User Guide

Broadcom Limited

Web: www.broadcom.com

Corporate Headquarters: San Jose, CA

Revision History

Rev.	Date	Notes
01	09/16/2024	Initial release

TABLE OF CONTENTS

[Broadcom NetXtreme Controller 5741x\(Wh+\) /5750x \(Thor\) /5760x \(Thor2\)](#)

[NICCLI User Guide](#)

[Broadcom Limited](#)

[Revision History](#)

[TABLE OF CONTENTS](#)

[1. Introduction](#)

[2. Supported Platforms](#)

[3. Installing the NICCLI Configuration Utility](#)

[3.1 Installing the NICCLI package in Linux](#)

[3.1.1 Executing the NICCLI package in Linux](#)

[3.2 Executing the NICCLI Configuration Utility in Windows](#)

[3.3 Installing the NICCLI Configuration Utility in VMware](#)

[3.4 Executing the NICCLI Configuration Utility in FreeBSD](#)

[3.5 Executing the NICCLI Configuration Utility in UEFI](#)

[4. NICCLI Configuration Utility Usage and Commands](#)

[4.1 NICCLI Configuration Utility Interface and Usage](#)

[4.1.1 NICCLI Configuration and Usage on VMWare 8.x](#)

[4.2 Hardware Interface Group of Arguments](#)

[4.3 NICCLI Configuration Utility Commands](#)

[4.3.1 Oneline Mode](#)

[4.3.2 Interactive Mode](#)

[4.3.3 Batch Mode](#)

[5. NICCLI Logging](#)

[6. Known Limitations/issues & usage guidelines](#)

[6.1 VMWare Limitations](#)

[6.2 Linux inbox limitations in secure kernel](#)

[7. Command Line Options](#)

1. Introduction

The NICCLI configuration utility sets the nonvolatile configuration elements of the Broadcom Ethernet network adapter, such as enabling or disabling RoCE, SR-IOV, and other options. The NICCLI configuration utility can also perform firmware upgrades. The NICCLI configuration utility uses the L2 driver in Linux, VMWare and FreeBSD and Windows. In the UEFI environment, the NICCLI configuration interacts with the PCIe hardware. The NICCLI configuration utility supports both the BCM9574XX, BCM95750X, and BCM957608 family of devices.

2. Supported Platforms

1. Linux :
 - x86_64
 - aarch64
2. Windows :
 - x86_64
3. ESXI 7/8 onwards
 - x86
 - x86_64
4. FreeBSD :
 - x86_64
5. UEFI :
 - x86_64
 - aarch64

3. Installing the NICCLI Configuration Utility

This section provides information on installing/executing the NICCLI configuration Utility:

3.1 Installing the NICCLI package in Linux

- Using NICCLI RPM:
`sudo rpm -i niccli -<version>.rpm`
- Using the NICCLI deb package:
`sudo dpkg -i niccli-<version>.deb`

Broadcom Proprietary and Confidential. © 2024 Broadcom. All rights reserved.

3.1.1 Executing the NICCLI package in Linux

- Users can execute the niccli binary by using the niccli-233.xxx-linux_<arch>.tar.gz package without installing the rpms/deb packages. User has to unzip the file and execute the niccli.<arch> on the OS.

3.2 Executing the NICCLI Configuration Utility in Windows

To execute the NICCLI configuration utility, unzip the provided Windows package file and use the niccli.exe file to run it on windows OS.

3.3 Installing the NICCLI Configuration Utility in VMware

To install the NICCLI configuration utility using a vib package:

```
esxcli software vib install -v <VIB package> --no-sig-check
```

NOTE:

The vib packaging is under the process of VMware signing.

To install the NICCLI configuration utility using a signed .zip bundle:

```
esxcli software vib install -d <zip package>
```

3.4 Executing the NICCLI Configuration Utility in FreeBSD

To execute the NICCLI configuration Utility, unzip the provided FreeBSD package file and use the niccli.freebsd executable to run it on FreeBSD OS.

3.5 Executing the NICCLI Configuration Utility in UEFI

To execute the NICCLI configuration utility, unzip the provided uefi package file in the uefi environment and use the niccli<arch>.efi executable file to run on the UEFI shell.

4. NICCLI Configuration Utility Usage and Commands

Provides information on PCI and operational inband communication using NICCLI. The NICCLI Configuration Utility is a management tool that is used to perform operations on Broadcom Ethernet network adapters. This utility provides support for PCI and operational inband communication. The utility also accepts arguments to select the communication interface or the specified device in which to communicate from the device list.

4.1 NICCLI Configuration Utility Interface and Usage

The NICCLI configuration utilities provide three different types of interfaces. By default, the utility starts with the interactive interface. The utility accepts three groups of command arguments based on the existing CLI standards.

```
<niccli> <HW i/f argument> [util arguments] [Target command]
```

4.1.1 NICCLI Configuration and Usage on VMWare 8.x

The following syntax is used when the signed bundle of NICCLI is installed:

```
esxcli niccli <command> -c <connection_type> -v <connection_type_value> [command options]
```

-c : Indicates connection type connection_type : Value for connection type. Supported values are [dev|i|pci]

-v : Indicates connection type value connection_type_value : Supported values are index_number, PF MAC Address and PCI Address

Examples:

1. esxcli niccli list
2. esxcli niccli debug -c dev -v 1 --coredump
3. esxcli niccli link -c dev -v BC:97:E1:70:14:10 --status
4. esxcli niccli debug -c pci -v 0000:86:00.00 --coredump

4.2 Hardware Interface Group of Arguments

The interface arguments depend on the hardware connection type and its specified dependent arguments. The NICCLI configuration utility supports the -pci interface which

takes the PCI Bus/Device/Location of the device. Alternatively, the utility also offers to list all the available Ethernet network adapter PCI devices in the system along with the appropriate Ethernet/Network interface names.

The NICCLI configuration utility has the `-i/-dev` index support which can select when more than one device is found within the host.

NICCLI Configuration Utility Arguments

The utility arguments are optional. These are specific to the NICCLI configuration utility itself. E.g. convert all the output into JSON or increase the logger verbosity, and so forth.

Target Command

The target is nothing but the NICCLI configuration utility connected device. These targets offer a specific set of commands depending on the connected interface/device. The target-specific commands are executed upon acquiring the connection with the target.

4.3 NICCLI Configuration Utility Commands

All the commands that are provided are case-sensitive and operate with any of the interface modes. The following rules are for the newly defined NICCLI configuration utility syntax. The commands use a specific syntax as follows:

- `< >` mandates user to specify the value
- `[]` is an optional parameter.
- Parameter syntaxes can also be combined such as `[-i <index value>]` optional `-i` index argument but mandatory index value, if `-i` switch specified.
- The NICCLI configuration utility provides "help" commands with brief information for every command.
- The NICCLI configuration utility shall accept combinations such as `-h`, `-?`, `'--help'` to display the help.
- Every command also has a detailed help description.
- The NICCLI configuration utility also displays the supported commands and/or valid command syntax when the user executes an invalid command. The NICCLI configuration supports the user command line argument as follows:

```
./niccli -i <index> <command line>
```

```
./niccli --pci <domain:bus:device.function> <command line>
```

NICCLI Configuration Utility Help

To access the NICCLI configuration utility help, use the following command:

```
./niccli [--help | -h]
```

Example:

```
./niccli -h
```

The utility provides three modes of execution:

4.3.1 Online Mode

Execute the NICCLI configuration utility on a per-target command basis. In this mode, specify the hardware interface and target command with appropriate arguments. The NICCLI configuration utility connects to the target, executes the target command, and exits from the application. The return status of the command is the exit status of the NICCLI configuration utility.

To list the available targets for Online mode use the following command:

```
niccli --list
```

Use the following command to display the list of available commands for Online Mode:

```
niccli [-i <index of the target> | --pci <NIC pci address>] --help
```

Use the following command to display the help for a specific command:

```
niccli [-i [<index of the target> | <mac addr> | <NIC pci address>]] --help  
<command>
```

Example:

```
niccli -i 1 help nvm
```

4.3.2 Interactive Mode

The NICCLI configuration utility starts in interactive console mode if no target command is provided. The interface starts with the target prompt upon a successful connection with the target.

- 1) The NICCLI configuration utility provides a help command to list all the available or supported commands.
- 2) The NICCLI configuration utility supports help [command] to display detailed help for the specific command. Upon execution of the given user command, the prompt is shown again for the next command.

NOTE:

This mode is best suited for connecting to the target and executing multiple operations/commands without having to disconnect from the target. This improves performance and time in establishing a connection with the target each time while executing a command. This is only for interactive usage and is not designed or meant for the scripting.

To launch in Interactive Mode, use the following command:

```
<NIC CLI executable> [-i <index of the target> | --pci <NIC pci address>]
```

Use the following command to display the list of available commands for Interactive Mode:

```
'help'
```

4.3.3 Batch Mode

Write the list of commands into a flat text file and execute them in the NICCLI configuration utility without disconnecting. This combines Interactive and OneLine modes without disconnecting the target. If any one of the commands fails, the NICCLI configuration utility exits and shall not continue to execute the rest of the commands from the script.

To launch in Batch Mode, use the following command:

```
<NIC CLI executable> [-i <index of the target> | --pci <NIC pci address>] --batch  
<batch file>
```

NOTE:

Batch mode requires a flat text file with utility-supported commands. Supported commands can be listed using OneLine mode or Interactive mode. Upon failure of any commands, the utility exits without continuing with other commands.

5. NICCLI Logging

1) 'scrutiny.ini' file captures the logs for the scrutiny library and to capture the logs for the CLI layer alone, user has to specify any of the following options in the cli command,

- A) niccli --verbose NULL <command>
- Prints the verbose logs on the console.

B) `niccli --debug NULL <command>`

- Prints the debug logs on the console.

C) `niccli --verbose file.txt <command>`

- Redirects the verbose logs into the file specified.

D) `niccli --debug file.txt <command>`

- Redirects the debug logs into the file specified.

2) Running any `niccli` command with 'scrutiny.ini' file being present in the same directory as `niccli` executable, will affect the overall completion time for the command. This is because debug logging will get enabled when 'scrutiny.ini' file is present. It is recommended to use only for debugging purposes.

6. Known Limitations/issues & usage guidelines

1) NICCLI framework does not support running multiple instances in parallel, sometimes it may lead to system crash and this scenario should be avoided.

2) White space characters other than plain space like tab etc. are not supported as argument separators in the interactive mode.

3) Interrupting `niccli` during the middle of some operations may result in unknown/undefined/unexpected behavior.

4) DCB commands i.e. `pfc`, `apptlv`, `up2tc`, `getqos`, `ets`, `listmap`, `dscp2prio` and `tcrlmt` to work, user has to disable the following nvm options "`lldp_nearest_bridge`", "`lldp_nearest_non_tpmr_bridge`" and "`dcbx_mode`".

5) The error codes will be supported only for the online mode, the interactive mode and the batch mode will not have error codes displayed.

6) If User encounters below error while running the `niccli` executable on Linux Systems, Please follow the below steps to avoid the below error:

`"/opt/niccli/niccli.x86_64: /lib/x86_64-linux-gnu/libnl-3.so.200: no version information available (required by /opt/niccli/niccli.x86_64)"`

A). Identify the `libnl` version installed on your Build Systems.

B). Install 3.2.28 version of `libnl3` (`libnl3` and `libnl-devel`)

- a. Source code can be found at below any one of the links
 - <https://www.infradead.org/~tgr/libnl/files/>
 - <https://snapshot.debian.org/package/libnl3/3.2.21-1/>
 - <https://snapshot.debian.org/package/libnl3/>
 - <https://www.linuxfromscratch.org/blfs/view/7.10/basicnet/libnl.html>
 - b. Untar the source code and run the below commands
 - `./configure --prefix=/usr --sysconfdir=/etc --disable-static`
 - `make`
 - `make install`.
 - c. Run `export LD_LIBRARY_PATH=<installed library path>`
 - Check the make install logs for the location where the library is installed.
 - d. Then run the `niccli`
- C). If the user is fine with ignoring the “no version information available”, then they can skip #B.

7) Unless `-all` option is explicitly specified, `resmgmt` command works based on its default mode (for a selected single PF or for all PFs/entire NIC device) which is printed when it's executed or its help is displayed. `-all` option performs the given command/operation for all the PFs of the given NIC device.

8) NICCLI / NICCLI in the Linux secure boot environment displays the "Firmware Reset Counter" and "Error Recovery Counter" fields in the `show` command always as zero as the `mmap` fails due to security considerations.

6.1 VMWare Limitations

1) In interactive mode, editing the command by moving the cursor using the left/right arrow will not work. The user has to re-issue the command.

2) Separate loggings make debugging easier for the developers.

Since it supports Storage, switch, expanders, NIC products through different ways of communication interfaces, we cannot change the existing logging mechanism design/functionality alone for NIC.

3) This design is well approved by all architects.

4) Interactive mode will not be supported in `niccli` Plugin model.

5) User has to provide an option `-s` (silent mode) for the command named `"restorefactorydefaults"` which requires user confirmation 'Yes/No'.

6.2 Linux inbox limitations in secure kernel

- 1) In Linux OS, When the secure boot is enabled the adapter configuration/query commands using niccli will not work as the mapping to the PCI BAR is not allowed by the OS.
- 2) Running multiple instances of niccli at the same time can result in the unexpected outputs and command timeouts.
- 3) In multihost environment, Race conditions can occur if more than one host attempts to utilize the USHI channel at the same time and this can result in the corruption of the control and data registers, timeouts, etc.
- 4) When the kernel configuration parameter CONFIG_IO_STRICT_DEVMEM=y is enabled and inbox bnxt_en driver is loaded, niccli adapter configuration/query commands will not work. This is because from the user space niccli cannot map the PCI BAR to access the hardware. Below are the two work-arounds for this issue.
 - A) Unbind the L2 driver from the PF.
 - B) Enable the iomem=relaxed in the grub and reboot the server.
- 5) niccli adapter configuration/query commands in the guest OS or VM can cause the unexpected outputs and command timeouts when the guest OS is loaded with the inbox bnxt_en driver and the PF is binded to the vfio-pci driver in the hypervisor and is attached to the guest OS or VM. In this case, guest OS or VM should be loaded with the out-of-box bnxt_en driver.

7. Command Line Options

To get a current list of supported commands by the niccli utility, use the niccli help option.

```
niccli help
```

Commands sets - Generic/Offline/Interface not required

Command	Description
devid	Query Broadcom device id's.
pkgver	Display FW PKG version installed on the device.

verify	Verify FW packages & NVM NVRAM Option Management
decode	Decodes the PTrace captured data
quit	Quits from the application (Applicable in interactive mode only)

Commands sets - BCM57xxx Commands

show	Shows NIC specific device information
nvm	Query or Configure device NVM
fw	Firmware manager
qos	Query or configure device QOS parameters
linkdiag	Link Diagnostics
serdes	Plots the SERDES eyes values
vf	Performs VF operations
cable	Display the cable information
link	Link Operations
timesync	Peer to Peer related operations
counters	Display and clear the PCIe port counters
tunnel	Performs Custom, GRE Tunnel and RSS(receive side scaling) operations
msix	Query or configure MSIX vector of VF's for each PF
mh	Modify and retrieve the PF count for each PCIe endpoint
resmgmt	Query and Configure resources of the device
ccparams	Query or configure the congestion control(cc) parameters for RoCE.
debug	Dumps device internal configuration registers

fwmanager commands

=====

DESCRIPTION :

Performs Firmware operations

SYNTAX :

```
fw <-u|--update> -f <package file> [--force] [--online] [-r|--recovery] [-n|--no_id_check]
[-y|--yes]
fw <--reset> [--cfa]
fw <-l|--livepatch> <--show>
```

```
fw <-l|--livepatch> <-a|--activate> [target_fw]
fw <-l|--livepatch> <-d|--deactivate> [target_fw]
fw <-l|--livepatch> <-p|--patch_update> [target_fw] -f <patch file>
```

OPTIONS :

```
-u|--update      : Perform the firmware install/update.
--force         : Forces the installation of the package file.
--reset         : This option is to perform the reset operations.
-y|--yes        : Answer as "yes" in prompts.
--cfa           : This option is to reset the OVS offload flows of the device.
-l|--livepatch  : Perform the firmware live patch operations.
--show          : Show the livepatch target firmware versions.
-a|--activate   : Activate the firmware livepatch from the NVM.
-d|--deactivate : Deactivate the firmware livepatch from the NVM.
-p|--patch_update : Update the patch file directly to the device i.e. without installing it in NVM
target_fw       : Target firmware is an optional parameter to active/deactivate the livepatch.
                  By default the tool updates all the supported target firmwares.
                  target_fw strings supported are "common_fw" or "secure_fw"
                  on BCM95750x devices. "chimp_fw" string is supported on BCM9574x
```

devices.

```
--online        : Fetch firmware image online from Broadcom web server & perform update.
-r|--recovery    : Recovers the adapter and updates the package file.
-n|--no_id_check : Recovers the adapter by ignoring the adapter PCI ID check.
                  This option is applicable only with the --recovery option.
```

EXAMPLES :

```
To perform the firmware update
  niccli -i 3 fw --update -f FW.pkg --yes
  niccli -i 3 fw --update --online --yes
  niccli -i 3 fw --update -f FW.pkg --recovery
  niccli -i 3 fw --update -f FW.pkg --recovery --no_id_check

To reset the device
  niccli -i 1 fw --reset

To query the livepatch firmware versions
  niccli -i 2 fw --livepatch --show

To activate the livepatch firmware from NVM
  niccli -i 7 fw --livepatch --activate

To deactivate the livepatch firmware from NVM
  niccli -i 1 fw --livepatch --deactivate

To update the patch file directly to the device
  niccli -i 2 fw --livepatch --patch_update -f patch.pkg
```

Config commands

=====

DESCRIPTION :

NVM configuration option of a device

This command provides :

- Display the current settings of the NVM configuration.
- Configure the current settings of the NVM configuration.
- Save the current settings of the NVM configuration.

SYNTAX :

```
nvm --view [-V] [-f <firmware package file name>] [-t|--type <nvm directory name>]
nvm -l|--list [-V] [-f <firmware package file name>]
nvm --verify [-V] [-f <firmware package file name>]
nvm -n|--sync
nvm -F|--restore_factory_defaults [--silent]
nvm -r|--dir_read -f <file name> -t|--type <nvm directory name>
nvm -w|--dir_write -f <file name> -t|--type <nvm directory name>
nvm -S|--saveoptions -f <file name>
nvm -O|--optionhelp <option name>
nvm -g|--getoption <option name> [--scope <scope index>]
nvm -s|--setoption <option names with comma seperated>
    -v|--value <option value with comma seperated> [--scope <scope index>]
nvm -L|--listoptions --diff
```

OPTIONS :

--view	: View the NVM item data.
-l --list	: Display the NVM components and its associated version details.
--verify	: Verify packages & NVM.
-F --restore_factory_defaults	: Restores NVM configuration to factory defaults.
-n --sync	: Synchronize SBI, SRT and CRT Primary and Secondary FW images. Supported on BCM9575xxx and BCM9576xxx devices.
-r --dir_read	: Read the NVM item data and write its contents to a file.
-w --dir_write	: Create or overwrite NVM data item with a file.
-S --saveoptions	: Save NVM configuration options on the device to a file. Only the end user access NVM configuration options are saved.
-O --optionhelp	: Detailed help for the NVM configuration option.
-g --getoption	: Get NVM configuration option of a device.
--setoption	: Set NVM configuration option of a device.
-v --value	: The value for the specified option. Value can be in hex or decimal format
--scope	: The scope can be either of 'function' or 'port' index.

-L|--listoptions : Displays current and default NVM configuration options of a device.
 --diff : Displays the difference between current and default NVM configuration options of a device.
 --silent : Silent option. Do not prompt for user message.
 -V : Verbosity.
 -f : Input file name.
 -t|--type : Input NVM directory name string.

EXAMPLE :

To view the NVM directory entries in detail:

```
niccli -i 2 nvm --view -V
```

To list the NVM directory entries:

```
niccli -i 2 nvm -l -f BCM957504-N1100GD.pkg
```

To verify the NVM directory entries:

```
niccli -i 2 nvm --verify
```

To Sync the primary and secondary firmware images:

```
niccli -i 2 nvm -n
```

To read NVM directory:

```
niccli -i 2 nvm -r -t pkglog -f data.txt
```

To write NVM directory:

```
niccli -i 2 nvm -w -f data.txt -t pkglog
```

To restore the config to factory defaults:

```
niccli -i 2 nvm --restore_factory_defaults --silent
```

To save the NVM config to a file:

```
niccli -i 2 nvm --saveoptions -f output.txt
```

To display the current settings for the NVM option:

```
niccli -i 2 nvm --getoption mac_address --scope 0
```

To configure the current settings of the NVM configuration:

```
niccli -i 2 nvm --setoption an_protocol --value 1 --scope 0
```

```
niccli -i 2 nvm -s 1,1 --scope 0,1 -v B0:26:28:99:88:14,B0:26:28:99:88:15
```

To list the current and default config settings:

```
niccli -i 2 nvm -L --diff
```

To get the help of each NVM config:

```
niccli -i 2 nvm --optionhelp an_protocol
```

qos commands

=====

DESCRIPTION :

This command can be used to query and configure the various QOS parameters such as ETS, priority to traffic class, application TLV's, receive rate control, tx and rx rate limits transmit(egress) and receive(ingress) buffer threshold input parameters.

SYNTAX :

```

qos <-E|--ets> --show
qos <-E|--ets --tsa <tc[0-7]:[ets|strict], ...> --up2tc <priority[0-7]:tc>, ...> --tcbw <list>
qos --pfc --enable <pfc list>
qos --up2tc --pri <priority[0-7]:tc, ...>
qos --apptlv <-a|--add> [<-d|--del>] --app <priority,selector,protocol>
qos <-D|--dscp2prio>
qos <-l|--listmap> --pri2cos
qos --tc --set <-T|--rate_limit> <list of rate limit>
qos <-r|--rx_port_rate_limit> --set --max <value> [-p]---persistent]
qos <-R|--rx_rate_limit> --show [-p]---persistent]
qos <-X|--rx_ep_rate_limit> --set --ep0 <value> [--ep1 <value>] [--ep2 <value>] [--ep3
<value>] [-p]---persistent]
qos <-t|--tx_partition_rate_limit> --show [-p]---persistent]
qos <-t|--tx_partition_rate_limit> --set --max <value> [-p]---persistent]
qos <-P|--tx_port_rate_limit> --show
qos <-P|--tx_port_rate_limit> --set --max <value>
qos <-x|--tx_ep_rate_limit> --show
qos <-x|--tx_ep_rate_limit> --set <--port> <port number> --ep0 <value> [--ep1 <value>] [--ep2
<value>] [--ep3 <value>] [-p]---persistent]
qos <-n|--ingress> --cosq --show [-p]---persistent]
qos <-n|--ingress> --cosq --set --state <value> [--mode <value>] [-p]---persistent]
qos <-e|--egress> --cosq --show [-p]---persistent]
qos <-e|--egress> --cosq --set --state <value> [-p]---persistent]

```

OPTIONS :

-E|--ets : Query or Configure enhanced transmission selection, priority to traffic class, traffic class bandwidths and the list of configured application tlvs.

--tsa : Transmission selection algorithm, sets a comma separated list of traffic classes to the corresponding selection algorithm. Valid algorithms include "ets" and "strict".

--up2tc : Comma separated list mapping user priorities to traffic classes.

--tcbw : Comma separated list of bandwidths for each traffic class the first value being assigned to traffic class 0 and the second to traffic class 1 and so on.

--pfc : Enable priority based flow control on a given priority.

-apptlv : Configure the priority of the application TLV.

-a|--add : Add the priority of the application TLV.

-d|--del : Delete the priority of the application TLV.

--app : Key to provide the priority, selector, protocol for configuring the application TLV.

-D|--dscp2prio : query the dscp to priority mapping.

-l|--listmap : List the priority mapping and related queue id for a given physical function.

--pri2cos : List the priority to traffic class mapping.

--tc : Command to set the rate limit for each traffic class.

-T|--rate_limit : Option to provide the comma separated percentage limit for each TC.

-r|--rx_port_rate_limit : Configure the receive side port rate limit

--max : The max option specifies an 8-bit rate limit as a percentage of total link bandwidth with a range of 0 to 100 percent. A value of 0 indicates no rate limit and deletes the previously configured rate limit.

-p|--persistent : Option to write the configuration to NVRAM, but it does not take effect immediately.

-R|--rx_rate_limit : Query the configured receive side rate control parameters.

-X|--rx_ep_rate_limit : Configure the receive side rate control parameters for a given endpoint.

-t|--tx_partition_rate_limit : Query and Configure the transmit side partition. Rate limit applies to traffic sent from a partition, which is one PF and all of its child VFs.

-P|--tx_port_rate_limit : Query and Configure the transmit side port rate limit.

--port : Specify the index of the external port of the device.

--ep(x) : Specify the Tx or Rx endpoints rate limit values.

-x|--tx_ep_rate_limit : Query and Configure the PCIe endpoint transmit rate control.

-n|--ingress : Query and configure the QoS dynamically to receive buffer thresholds by configuring different input parameters.

-e|--egress : Query and configure the QoS dynamically at transmit buffer thresholds by configuring different input parameters.

--cosq : This option is used to query or set the cosq parameter i.e. cosq state and the mode.

`--state` : Bitmask field indicating which traffic classes are enabled or disabled. Each bit represents a specific traffic class, where bit 0 represents traffic class 0 and so on. A value of 0 indicates that the traffic class is not enabled.

`--mode` : Bitmask field indicating which traffic class are lossy or lossless. Each bit represents a specific traffic class, where bit 0 represents traffic class 0 and so on. A value of 0 indicates that the traffic class is lossy and value 1 indicates that the traffic class is lossless.

EXAMPLES :

To show the ETS(enhanced transmission selection) configuration

```
niccli -i 5 qos --ets --show
```

To configure the ETS(enhanced transmission selection)

```
niccli -i 5 qos --ets --tsa 0:ets,1:ets,2:strict,3:strict,4:strict,5:strict,6:strict,7:strict --up2tc 0:0,1:0,2:0,3:0,4:0,5:1,6:0,7:0 --tcbw 70,30
```

To enable priority based flow control on a given priority

```
niccli -i 5 qos --pfc --enable 5,6
```

```
niccli -i 5 qos --pfc --enable 0xFF
```

To set the user priorities to traffic classes

```
niccli -i 3 qos --up2tc 0:0,1:0,2:0,3:0,4:0,5:1,6:0,7:0
```

To add the priority of the application TLV

```
niccli -i 3 qos --apptlv --add --app 5,1,35093
```

To delete the priority of the application TLV

```
niccli -i 5 qos --apptlv --del --app 5,1,35093
```

To query the dscp to priority mapping

```
niccli -i 5 qos --dscp2prio
```

To list the priority to traffic class mapping

```
niccli -i 5 qos --listmap --pri2cos
```

To set the rate limit for each traffic class in units of percentage

```
niccli -i 5 qos --tc --set --rate_limit 10,20,30
```

```
niccli -i 5 qos --tc --set --rate_limit 10
```

To configure receive rate control that applies to all traffic in a receive CoS queue group

```
niccli -i 2 qos --rx_port_rate_limit --set --max 40
```

```
niccli -i 2 qos --rx_port_rate_limit --set --max 70 -persistent
```

To show the receive side rate limits

```
niccli -i 1 qos --rx_rate_limit --show
```

```
niccli -i 1 qos --rx_rate_limit --show --persistent
```

To configure endpoint rate limit for all endpoints from one host

```
niccli -i 1 qos --rx_ep_rate_limit --set --ep0 0
```

```
niccli -i 1 qos --rx_ep_rate_limit --set --ep0 0 --persistent
```

To show the Tx partition rate limit

```
niccli -i 1 qos --tx_partition_rate_limit --show
```

To configure the Tx partition rate limit

```

niccli -i 1 qos --tx_partition_rate_limit --set --max 2
To show the transmit side port rate limit
niccli -i 1 qos --tx_port_rate_limit --show
To show the transmit side port rate limit
niccli -i 2 qos --tx_port_rate_limit --show
To configure the transmit side port rate limit
niccli -i 2 qos --tx_port_rate_limit --set --max 2
To query the PCIe endpoint transmit rate control
niccli -i 2 qos --tx_ep_rate_limit --port 0 --show
To configure the PCIe endpoint transmit rate control for two endpoints
niccli -i 5 qos --tx_ep_rate_limit --set --port 0 --ep0 50 --ep2 40
To query the ingress cosq parameters
niccli -i 5 qos --ingress --cosq --show
To enable all the 8 traffic classes and mode lossless(1) is configured on traffic class 4
niccli -i 5 qos --ingress --cosq --set --state 255 --mode 16
To query the egress cosq parameters
niccli -i 5 qos --egress --cosq --show
To configure the egress cosq parameters. Below example enables all the 8 queues
niccli -i 5 qos --egress --cosq --set --state 255

```

linkdiag command

=====

DESCRIPTION :

This command is used to perform the link diagnostic tests like PRBS, loopback, DSCDump and TXFIR settings.

SYNTAX :

```

linkdiag -T|--txfir --show <-M|--modulation_type> <mod_type> <-l|--lane> <lane_number>
linkdiag -T|--txfir --set <-M|--modulation_type> <mod_type> <-l|--lane> <lane_mask>
    --pre1 <value> --pre2 <value> [--pre3 <value>] --main <value> --post1 <value>
    --post2 <value> [--post3 <value>] --amp <value> --nlcl <value> --nlcu <value>]

linkdiag -T|--txfir --show <-M|--modulation_type> <mod_type> <-l|--lane> <lane_number>

linkdiag -F|--fdrstat [--start] [--stop] [--clear] [--counters]

linkdiag -D|--dsdump -l|--lane <lane_number> [-a|--diag_level <level>]

linkdiag -L|--loopback --show

```

```
linkdiag -L|--loopback [<-P|--phy_remote> | <-p|--phy_local> | <-m|--mac_local> |
<-d|--disable>]
```

```
linkdiag -L|--loopback [--external] [--RJ45]
```

```
linkdiag -P|--prbs_test <-e|--enable | -d|--disable> [--mode <mode_value>]
[<-r|--rx_lane_mask> <value>]
[<-t|--tx_lane_mask> <value>] [<-s|--duration> <value in seconds>] [--tcode]]
```

OPTIONS :

-T|--txfir : This option is used to query and configure the TX FIR (transmitter finite impulse response)

-F|--fdrstat : This option is used for FDR(Flight Data Recorder) to collect the FEC Performance data.

-D|--dscdump : This option is used to retrieve DSC dump data from a device

-L|--loopback : This option is used to query and configure the different loopback Modes i.e. phy loopback, mac loopback and external loopback.

-P|--prbs_test : This option is used in port interface debugging to analyze the quality of the link. The test can be run on a port or per lane with a specific polynomial.

Note:

- 1) The interface(s) should be fully initialized prior to the testing.
- 2) During the testing, there should not be any queries or configs sent to the card. "ifdown" the interface(s) is recommended.

-M|--modulation_type : Modulation types of TxFIR. Supported values are 'NRZ','PAM4','C2MNRZ','C2MPAM4', 'PAM4-112','C2MPAM4-112G' and 'LPOPAM4-112G' of the device. The modulation types 'PAM4-112','C2MPAM4-112G' and 'LPOPAM4-112G' are only supported on BCM5760x devices.

The modulation types 'PAM4' and 'C2MPAM4' are supported on BCM5750x and BCM5760x devices.

-l|--lane : TXFIR show command takes the MRS lane number and lane mask for the TXFIR configuration. To collect the dsc dump on all the supported lanes, please provide a value of 65535. DSC dump on all lanes is supported only on BCM9576xx devices.

--pre1 : This is a mandatory parameter to configure the TXFIR settings. This parameter is supported for all the modulation types and the valid range is from -32768 to 32767.

--pre2 : This is a mandatory parameter to configure the TXFIR settings. This parameter is supported for all the modulation types and the valid range is from -32768 to 32767.

--main : This is a mandatory parameter to configure the TXFIR settings. This

parameter is supported for all the modulation types and the valid range is from -32768 to 32767.

--post1 : This is a mandatory parameter to configure the TXFIR settings. This parameter is supported for all the modulation types and the valid range is from -32768 to 32767.

--post2 : This is a mandatory parameter to configure the TXFIR settings. This parameter is supported for all the modulation types and the valid range is from -32768 to 32767.

--pre3 : This is an optional parameter to configure the TXFIR settings. This parameter is supported on following modulation types i.e, 'LPOPAM4-112G', 'PAM4-112', 'C2MPAM4' and 'C2MPAM4-112G'. The valid range is from -32768 to 32767.

--post3 : This is an optional parameter to configure the TXFIR settings. This parameter is supported on the following modulation types i.e, 'NRZ', 'PAM4' and 'C2MNRZ'. The valid range is from -32768 to 32767.

--amp : This is an optional parameter to configure the TXFIR settings. This parameter is supported on following modulation types i.e, 'NRZ', 'PAM4', 'C2MNRZ', 'PAM4-112', 'C2MPAM4' and 'C2MPAM4-112G'. The valid range is from -32768 to 32767.

--nlcu : This is an optional parameter to configure the TXFIR settings. This parameter is supported only on 'LPOPAM4-112G' modulation type. The valid range is from -100 to 100.

--nlcl : This is an optional parameter to configure the TXFIR settings. This parameter is supported only on 'LPOPAM4-112G' modulation type. The valid value is 0.

--start : This is an optional parameter to start the fdrstat.

--stop : This is an optional parameter to stop the fdrstat.

--clear : This is an optional parameter to clear the fdrstat.

--counters : This is an optional parameter to pull the fdrstat counters information.

-a|--diag_level : This is an optional parameter. If the user does not specify this parameter by default DSC dump will be collected on all the supported diag levels.

Supported diag levels are as follows:

- 0 = diag lane
- 1 = diag core
- 2 = diag event
- 3 = diag eye
- 4 = diag reg core

5 = diag reg lane
 6 = diag uc core
 7 = diag uc lane
 8 = diag lane debug
 9 = diag ber vert
 10 = diag ber horz
 11 = diag event safe
 12 = diag timestamp

-R|--phy_remote : This option enables loopback of local PHY Rx to peer PHY Tx. The packets transmitted by peer are looped back to the peer at the PHY. No packets will reach the host. Host will see a link down.

-p|--phy_local : This option enables a loopback of local TX to local RX at the PHY. Any packets transmitted from the host are looped back to the host. No packets will be transmitted on the line. If any peer is connected, the peer should ignore the link status from the host.

-m|--mac_local : This option enables a loopback of local TX to local RX at the MAC. Any packets transmitted from the host are looped back to the host. No packets will be transmitted on the line. If any peer is connected, the peer should ignore the link status from host.

-d|--disable : This option disables the current loopback settings. In case of prbs_test this option disables the PRBS test.

--external : This option prepares the PHY from external loopback and suppresses NONCE generation for auto negotiation to work with an external loopback. This option should only be used if the same port external loopback dongle or equivalent is used. Without this option and AN enabled, the host may not see a link up.

--RJ45 : This test is designed for dual port 10GBase-T where the PRBS test is not applied. A compliant UTP cable is needed to connect between the two ports of the controller because auto-negotiation is required to establish a link, traffic is initiated from one port. The PHY of the other port is put into a remote loopback mode, essentially serving as a loopback plug, to bounce packets back to the initiating port. This test requires the production Linux driver (bnxt_en) to be loaded in order to execute. Packets are sent and received through the Linux OS network stack

-e|--enable : Enable the PRBS test

--mode : Specify the supported modes. And the supported modes are 'PRBS31','PRBS7','PRBS9','PRBS11','PRBS15','PRBS23','PRBS58','PRBS49','PRBS10','PRBS20' and 'PRBS13'. The default mode value is PRBS31

-r|--rx_lane_mask : Receiver lane mask value.

-t|--tx_lane_mask : Transmitter lane mask value.
 -s|--duration : Duration to run the prbs test. Default time is 10 seconds

--tcode : If this option was provided. The prbs test will run on t-code project as well.

EXAMPLES :

To set the TXFIR settings

```
niccli -i 1 linkdiag -T --set --modulation_type LPOPAM4-112G --lane 1 --pre1 1 --pre2 -2
--pre3 10 --main 12 --post1 -10 --post2 15 --nlcl 0 --nlcu 1
niccli -i 1 linkdiag -T --set --modulation_type PAM4 --lane 1 --pre1 1 --pre2 -2 --main 12
--amp 10 --post1 -10 --post2 15 --post3 10
niccli -i 1 linkdiag -T --set --modulation_type NRZ --lane 1 --pre1 1 --pre2 -2 --main 12
--amp 10 --post1 -10 --post2 15 --post3 10
niccli -i 1 linkdiag -T --set --modulation_type PAM4-112 --lane 1 --pre1 1 --pre2 -2 --pre3 5
--main 12 --amp 10 --post1 -10 --post2 15
```

To get the TXFIR settings

```
niccli -i 2 linkdiag -T --show --modulation_type LPOPAM4-112G --lane 0
niccli -i 2 linkdiag -T --show --modulation_type PAM4 --lane 0
niccli -i 2 linkdiag -T --show --modulation_type NRZ --lane 0
niccli -i 2 linkdiag -T --show --modulation_type PAM4-112 --lane 0
```

To Start the fdrstat

```
niccli -i 4 linkdiag --fdrstat --start
```

To Stop the fdrstat

```
niccli -i 4 linkdiag --fdrstat --stop
```

To Clear the fdrstat

```
niccli -i 4 linkdiag --fdrstat --clear
```

To Pull the fdrstat counters

```
niccli -i 4 linkdiag --fdrstat --counters
```

To get the DSC Dump

```
niccli -i 3 linkdiag -D --lane 0
niccli -i 3 linkdiag -D --lane 0 --diag_level 2
```

To get loopback status

```
niccli -i 4 linkdiag --loopback --show
```

To disable the loopback mode

```
niccli -i 4 linkdiag --loopback --disable
```

To enable the mac_local loopback mode

```
niccli -i 5 linkdiag --loopback --mac_local
```

To enable the phy_local loopback mode

```
niccli -i 5 linkdiag --loopback --phy_local
```

To enable the phy_remote loopback mode


```

    niccli -i 5 linkdiag --loopback --phy_remote
To enable the external loopback mode
    niccli -i 6 linkdiag --loopback --external
To enable the external RJ45 loopback mode
    niccli -i 6 linkdiag --loopback --external --RJ45
To enable the PRBS Test with default
    niccli -i 3 linkdiag --prbs_test --enable
To disable the PRBS Test
    niccli -i 3 linkdiag --prbs_test --disable
To run the PRBS Test with user provided params
    niccli -i 2 linkdiag --prbs_test --enable --mode PRBS31 --rx_lane_mask 255
--tx_lane_mask 255 --duration 10

```

serdes command

=====

DESCRIPTION :

This command is used to plot the serdes ethernet eye, PCI eye scope and margin values of the eye.

Note:

1. While plotting the serdes ethernet eye the link toggling is expected.
2. Serdes ethernet and pci eye shares the resources of the NIC. Therefore, these commands cannot be run concurrently. If pci eye is running and you attempt to run ethernet eye tool will return failure.

SYNTAX :

```

serdes --eye -e|--ethernet [-l|--lane <ethernet lane number>] [-P|--plot]
serdes --eye -p|--pci [-l|--lane <pci_lane_number>] [-P|--plot] [-t|--target_ber <value>]
serdes --eye -p|--pci -s|--stop

```

OPTIONS :

--eye : This option is used to plot the serdes pci and ethernet eye.

-e|--ethernet : This option is used to plot the serdes ethernet eye.

By default this options displays only horizontal and vertical margin values, including the test result.

-p|--pci : This option is used to plot the serdes pci eye.

-l|--lane : This option is used to specify the lane number. For pcie serdes test, the maximum value is the device pcie lane width minus 1. Valid values are from 0 to 15. For ethernet serdes test the valid range is from 0 to 7.

-P|--plot : This is an optional parameter. When user specifies this option, will plot the eye and displays the horizontal and vertical margin values, including the test result.

-t|--targetber : This option is used to specify the target bit error rate. By default serdes pci eye is plotted with BER "1e-8". This option is only supported on BCM9575xxx and above devices. The supported target BER values are "1e-8", "1e-9", "1e-10" and "1e-11"

-s|--stop : This option is used to stop the running serdes pci eye plotting. This option is only supported on BCM9575xxx and above devices.

EXAMPLES :

To plot the ethernet serdes eye

```
niccli -i 4 serdes --eye -e -P
```

```
niccli -i 4 serdes --eye --ethernet --plot
```

To plot the ethernet serdes eye with specific lane number

```
niccli -i 2 serdes --eye -e -l 0 -P
```

```
niccli -i 2 serdes --eye --ethernet --lane 0 --plot
```

To get the PCI serdes eye margins and Rx settings with specific lane number

```
niccli -i 3 serdes --eye --pci --lane 0
```

```
niccli -i 3 serdes --eye -p -l 0
```

To plot the PCI serdes eye with specific lane number

```
niccli -i 2 serdes --eye --pci --lane 0 --plot
```

```
niccli -i 2 serdes --eye -p -l 0 -P
```

To plot the PCI serdes eye with specific lane number and target BER

```
niccli -i 1 serdes --eye --pci --lane 0 --plot --target_ber 1e-8
```

```
niccli -i 1 serdes --eye -p -l 0 -P -t 1e-8
```

To stop the PCI serdes eye test

```
niccli -i 7 serdes --eye --pci --stop
```

```
niccli -i 7 serdes --eye -p -s
```

cable command

=====

DESCRIPTION :

Query/Decode Module EEPROM information and optical diagnostics.

SYNTAX :

```
cable -m|--module_info --show
```

```
cable -r|--read_module_eeprom [-p|--page_number <page number> -o|--offset <byte offset>
```

```
-l|--length <number of bytes> -b|--bank <bank number> -i|--i2c_address <i2c addr>]
```

```
cable -w|--write_module_eeprom -p|--page_number <page number> -o|--offset <byte offset>
```

```
-v|--value <bytes>
```

cable -M|--module_loopback -t|--loopback_type <type> [--lane <module_lane_number>]

OPTIONS:

- m|--module_info : Get the module information
- r|--read_module_eeprom : Read the module EEPROM in hex format
- w|--write_module_eeprom : Write the bytes into the module EEPROM
- i|--i2c_address : I2C address of a page. Value less than 0x7f expected
Most of the EEPROMs use 0x50 or 0x51
- p|--page_number : The page number that is being accessed over I2C
- l|--length : Length of EEPROM data to read or write
- o|--offset : Offset within the page that is being accessed over I2C
- b|--bank : The bank number of the page that is being accessed over I2C
- v|--value : Bytes to write into module EEPROM
- M|--module_loopback : This option is used to perform the various module loopback. This operation is supported only on CMIS 4.0 and above supported modules.
- t|--loopback_type : This option is used to ran the module loopback on the specified loopback type.

The supported module loopback types are as below:

- 1 - Media Side Output Loopback
- 2 - Media Side Input Loopback
- 3 - Host Side Output Loopback
- 4 - Host Side Input Loopback

--lane : This is an optional parameter. If user specify this option with a valid lane number, then the module loopback will be run on specified lane.

--show : Displays the information in detailed.

EXAMPLES:

To show the module information

```
niccli -i 1 cable -m --show
```

```
niccli -i 1 cable --module_info --show
```

To Read the module eeprom

```
niccli -i 2 cable -r -p 0 -o 0 -l 128 -b 0 -i 0x50
```

```
niccli -i 2 cable --read_module_eeprom --page_number 0 --offset 0 --length 128 --bank 0  
--i2c_address 0x50
```

To Write to module eeprom

```
niccli -i 3 cable -w -p 0 -o 0 -v 20
```

```
niccli -i 3 cable --write_module_eeprom --page_number 0 --offset 0 --value 20
```

To run the 'Media Side Output Loopback'

```
niccli -i 3 cable -M -t 1 -l 0
```

```
niccli -i 3 cable --module_loopback --loopback_type 1 --lane 0
```

To run the 'Media Side Input Loopback'

```
niccli -i 2 cable -M -t 2 -l 0
```

```
niccli -i 2 cable --module_loopback --loopback_type 2 --lane 0
```

To run the 'Host Side Output Loopback'

```
niccli -i 2 cable -M -t 3 -l 0
```

```
niccli -i 2 cable --module_loopback --loopback_type 3 --lane 0
```

To run the 'Host Side Input Loopback'

```
niccli -i 1 cable -M -t 4 -l 0
```

```
niccli -i 1 cable --module_loopback --loopback_type 4 --lane 0
```

link command

=====

DESCRIPTION :

This command is used to query the link status, BER information, physical counters and configure the port state.

SYNTAX :

```
link -s|--status
```

```
link -c|--counters --show
```

```
link -p|--port_state <port state value>
```

OPTIONS :

-s|--status : Shows the link status and related information

-c|--counters : Show physical counters and BER Info

-p|--port_state : Configures the portstate. The valid values are 0(Down), 1(UP) and 2(Toggle)

EXAMPLES :

To get the link information

```
niccli -i 1 link -s
```

```
niccli -i 1 link --status
```

To get the physical counters and BER information

```
niccli -i 2 link -c --show
```

```
niccli -i 2 link --counters --show
```

To configure the port state to down

```
niccli -i 2 link -p 0
```

```
niccli -i 2 link --port_state 0
```

To configure the port state to up

```
niccli -i 3 link -p 1
```

```
niccli -i 3 link --port_state 1
```

To toggle the port state

```
niccli -i 1 link -p 2
niccli -i 2 link --port_state 2
```

timesync command

=====

DESCRIPTION :

Timesync operations

Timesync command provides the user to:

- To set duty cycle on TSIO outgoing signal.
- To set the DLL source for PHC.
- Set PTP extended parameters operation. All the parameters are optional.
- Configure and display the synchronous ethernet frequency profile, primary and secondary clock state.
- TSIO operations for the requested PF/VF

SYNTAX :

```
timesync <-d | --dutycycle> <--period> <value> --up <value>
```

```
timesync <--dll> <-s | --source> <value> <-q | --frequency> <value>
```

```
timesync <--ptp> --show
```

```
timesync <--ptp> --set <-p | --primary_pf> <pid> [<-v | --primary_vf> <vfid>]
```

```
[<-P | --secondary_pf> <pfid>] [<-V | --secondary_vf> <vfid>]
```

```
timesync <--synce> --show
```

```
timesync <--synce> --set <-Q | --frequency_profile> <value> [<-c | --primary_clock_state>
<value>] [<-C | --secondary_clock_state> <value>]
```

```
timesync <--tsio> <-t | --tsio_function_pin> <idx> <-u | --pin_usage_string> <value>
<--state> <value>
```

OPTIONS :

- d|--dutycycle : To set duty cycle on TSIO outgoing signal.
- period : value for period will be treated as in nanoseconds.
- up : Up flag is used to set the duty cycle and should be lesser than period value.
- dll : To set the DLL source for PHC.
- s | --source : The valid values range is 0 to 4.
- q | --frequency : The valid values range is 0 to 3
- p | --ptp : PTP extended parameters operation.
- show : Displays timesync operation.

--set : Configures timesync operation.
 -p | --primary_pf : Primary physical function ID.
 -v | --primary_vf : Primary virtual function ID belongs to primary PF ID.
 -P | --secondary_pf : Secondary physical function ID.
 -V | --secondary_vf : secondary virtual function ID belongs to secondary PF ID.

--synce : Configure and display the synchronous ethernet frequency profile.
 primary and secondary clock state. This command is supported only on BCM9575xxx devices.

-Q | --frequency_profile : Frequency profile for SyncE recovered clock. Supported profiles are "25MHz"

-c | --primary_clock_state : Enable or disable primary clock for PF or port, overriding previous primary clock setting.

-C | --secondary_clock_state : Enable or disable secondary clock for PF or port, overriding previous secondary clock setting.

--tsio : Displays or Configures tsio function capability on the pin

-t | --tsio_function_pin : Pin Index. Valid Index 0 to 3
 -u | --pin_usage_string : Pin usage string.
 --state : Enable/Disable function capability on the pin

EXAMPLES :

Setting duty cycle on TSIO outgoing signal

```
niccli -i 2 timesync -d --period 1 --up 0
niccli -i 2 timesync --dutycycle --period 1 --up 0
```

To configure the DLL source for PHC

```
niccli -i 2 timesync --dll -s 1 -q 3
niccli -i 2 timesync --dll --source 1 --frequency 3
```

To perform PTP extended parameters operation

```
niccli -i 1 timesync --synce --show
niccli -i 1 timesync --synce --set -Q 25MHz
niccli -i 3 timesync --synce --set --frequency_profile 25MHz
```

To perform TSIO function capability on the pin

```
niccli -i 1 timesync --tsio -t 3 -u 1 --state 1
niccli -i 1 timesync --tsio --tsio_function_pin 3 --pin_usage_string 1 --state 1
```

To perform PTP extended parameters operation.

```
niccli -i 4 timesync --ptp --show
```

```
niccli -i 4 timesync --ptp --set -p 1 -v 1
```

```
niccli -i 6 timesync --ptp --set --primary_pf 1 --secondary_pf 1
```

Counters command

=====

DESCRIPTION :

This command is used to display and clear the PCIe port counters

SYNTAX :

```
counters -p | --pcie
```

```
counters -c | --clear
```

OPTIONS :

```
-p | --pcie : Display Pcie Counters.
```

```
-c | --clear : Clear the port counters
```

EXAMPLES :

To display the PCIe port counters

```
niccli -i 1 counters -p
```

```
niccli -i 1 counters --pcie
```

To clear the port counters

```
niccli -i 3 counters -c
```

```
niccli -i 2 counters --clear
```

vf configuration commands

=====

DESCRIPTION :

Performs VF operations

SYNTAX :

```
vf <-t|--trust> --set <-v|--vf_index> <idx> <--state> <enable/disable>
```

```
vf <-t|--trust> --show <-v|--vf_index> <idx>
```

```
vf <-a|--add_ntuple_filter> <-m|--macaddress> <value> <-p|--dest_port> <value>
```

```
<-P|--dst_port_mask> <value> <-v|--vf_index> <idx> <-T|--ip_type> <value>
```

```
vf <-d|--free_ntuple_filter> <-l|--filter_id> <value>
```

```
vf <-M|--peer_mem_map> <--hpa> <list of values> <--gpa> <list of values>
```

```
<--size> <list of values>
```

OPTIONS :

- t|--trust : Perform the trusted VF operations.
- v|--vf_index : Provide the VF index.
- state : Option to enable or disable the trusted VF.
- a|--add_ntuple_filter : Option to add the ntuple flow filter.
- d|--free_ntuple_filter : Option to free the ntuple flow filter.
- m|--macaddress : MAC address in format xx:xx:xx:xx:xx:xx.
- p|--dest_port : Option to provide the destination port.
- P|--dst_port_mask : Option to provide the destination port mask.
- T|--ip_type : Option to provide the IP type. The valid values are:
1 - IPV4, 2 - IPV6, 3 - ARP-REPLY
- M|--peer_mem_map : Option to configure the GPU host and guest physical address mapping.
- hpa : This option is used to specify the list of host physical addresses.
Max 8 entries are supported. User has to provide the list with a comma separated for each host physical address. The value should be in the hex-decimal.
- gpa : This option is used to specify the list of guest physical addresses.
Max 8 entries are supported. User has to provide the list with a comma separated for each guest physical address. The value should be in the hex-decimal.
- size : This option is a comma separated list in kilobytes for each mapping.
Max 8 entries are supported. The value should be in the hex-decimal.

EXAMPLES :

```

To query trusted vf state
  niccli -i 4 vf --trust --show --vf_index 1
To enable trusted vf state
  niccli -i 2 vf --trust --set --vf_index 1 --state enable
To add ntuple flow filter for the specified MAC and destination port
  niccli -i 1 vf --add_ntuple_filter --macaddress 00:01:02:03:04:a3 -p 1023 -P 0xFFFF -v 1 -T
1
To free ntuple flow filter for the specified filter id
  niccli -i 1 vf --free_ntuple_filter --filter_id F06C0000D6C22414
To configure the GPU host and guest physical address mapping
  niccli -i 5 vf --peer_mem_map --hpa
0x1FFFFFFFF,0x2FFFFFFFF,0x3FFFFFFFF,0x4FFFFFFFF
--gpa 0x9FFFFFFFF,0xAFFFFFFFF,0xBFFFFFFFF,0xCFFFFFFFF
--size 0x10000,0x10000,0x8000,0x8000

```


cfgtunnel commands

=====

DESCRIPTION :

Performs Custom, GRE Tunnel and RSS(receive side scaling) operations

SYNTAX :

```
tunnel --cfg --vxlan <-t|--type> <ipv4|ipv6> --show
tunnel --cfg --vxlan <--add> <-t|--type> <ipv4|ipv6> <-p|--dest_port> <value>
tunnel --cfg --vxlan <--del> <-t|--type> <ipv4|ipv6> <-p|--dest_port> <value>
tunnel --cfg <--rss> --show
tunnel --cfg <--rss> --set <--mode> <inner/outer>
```

OPTIONS :

--cfg : Perform Custom, GRE Tunnel and RSS(receive side scaling) operations.
 --vxlan : Option to query or configure vxlan type.
 -t|--type : Option to provide the IP type. The valid values are "ipv4" and "ipv6".
 -p|--dest_port : Option to provide the destination port. Valid range is 0-65535.
 --rss : Option to query and configure RSS(receive side scaling).
 --mode : Option to configure the RSS mode. The valid values are "inner" and "outer".
 -g|--gre_tunnel_offload : Option to query and configure the custom GRE tunnel offload.
 --state : Option to enable or disable the non udp port based GRE tunnel offload.

EXAMPLES :

To show the custom tunnel configuration
 niccli -i 2 tunnel --cfg --vxlan --type ipv4 --show
 To add the custom tunnel configuration on destination port
 niccli -i 2 tunnel --cfg --vxlan --add --type ipv4 --dest_port 1024
 To delete the custom tunnel configuration on destination port
 niccli -i 2 tunnel --cfg --vxlan --del --type ipv4 --dest_port 1024
 To show the RSS mode configuration
 niccli -i 3 tunnel --cfg --rss --show
 To configure the RSS inner mode
 niccli -i 1 tunnel --cfg --rss --set --mode inner
 To show the state of GRE tunnel offload
 niccli -i 1 tunnel --cfg --gre_tunnel_offload --show
 To enable the non udp port based GRE tunnel offload
 niccli -i 1 tunnel --cfg --gre_tunnel_offload --set --state enable

msix commands

=====

DESCRIPTION :

Query and configure the number of MSI-X max vectors values for VF's per each PF

SYNTAX :

```
msix -m|--max_vectors --show [--all | --pf <pf number>]
msix -m|--max_vectors --set [--pf <pf number>]
```

OPTIONS :

```
-m|--max_vectors : Retrieves the register dump and crashdump from the firmware.
--pf             : PF Number to query the table of 8 rows for msix max vectors.
--show           : Get the msix max vectors for PF.
--set            : Configure the msix max vectors for PF.
--all            : Displays msix max vectors for all the PF's.
```

EXAMPLES :

```
To display the MSI-X max vectors values
  niccli -i 1 msix --max_vectors --show --pf 0
To display all the values of MSI-X max vectors
  niccli -i 2 msix --max_vectors --show --all
To configure the MSI-X max vectors values for PF 0
  niccli -i 3 msix --max_vectors --set --pf 0
```

mh command

=====

DESCRIPTION :

Query and configure the Broadcom Multi-Host PF information.

SYNTAX :

```
mh -p|--pf_alloc --show
mh -p|--pf_alloc --set --ep0 <pf_cnt> --ep1 <pf_cnt> --ep2 <pf_cnt> --ep3 <pf_cnt>
```

OPTIONS :

```
-p|--pf_alloc : PF Number to query Multi-Host PF information
--show        : Query Multi-Host PF information
--set         : Configure the Multi-Host PF information
--ep0         : number of PF to be written on EP0
--ep1         : number of PF to be written on EP1
--ep2         : number of PF to be written on EP2
--ep3         : number of PF to be written on EP3
```

NOTE: The number of non-zero values can be for 2 endpoints or 4 endpoints.

NOTE: The sum of all the endpoints should be less than or equal to 16.

EXAMPLES :

To display the Multi-Host PF information

```
niccli -i 1 mh --pf_alloc --show
```

To configure the Multi-Host PF information

```
niccli -i 6 mh --pf_alloc --set --ep0 0 --ep1 0 --ep2 0 --ep3 0
```

ring resource command

=====

DESCRIPTION :

Query and Configure resources of the device.

SYNTAX :

```
resmgmt [--pf/--all] [<-p|--profile> | < --min > | < --max > | <-r | --roce_max> | < -m |  
--max_completion_rings> | < -s | --strategy >] --show
```

```
resmgmt [--pf/--all] --set [<-p|--profile> | < --min > | < --max > | <-r | --roce_max> | < -m |  
--max_completion_rings>] [< --bw <bandwidth of each pf with comma separated>] [< -s |  
--strategy > < minimal/maximal/minimal-static>]
```

OPTIONS :

-p|--profile : Active profile of the device

-r|--roce_max : RoCE enabled PF's on the device(supported only for Stratus & Cumulus-B/WHP)

-m|--max_completion_rings : maximum completion rings for each active PF

-s|--strategy : strategy associated with each active PF.

--min : Minimum bandwidth associated with each active PF

--max : Maximum bandwidth associated with each active PF

--show : Displays resources of the device

--set : Configures resources of the device

--bw : Bandwidth to configure for PF N. This is applicable for min, max, roce_max and max_cmpl

EXAMPLES :

To query resources of the device

```
niccli -i 7 resmgmt --all --profile --show
```

To configure resources of the device

```
niccli -i 7 resmgmt --pf --set --bw 1,2,3 --strategy minimal
```

Congestion Control command

=====

DESCRIPTION :

This command is used to query and configure the congestion control(cc) parameters for RoCE.

SYNTAX :

```
ccparams -d|--dump
ccparams --set -f <configuration file>
```

OPTIONS :

```
-d|--dump      : Dumps the congestion control parameters into a configuration <xxx>.CFG file.
The file will be generated in the same directory where the executable is running.
--set          : To configure the congestion control parameters using a <xxx>.CFG file.
-f            : Configuration file to configure the congestion control (cc) parameters.
```

EXAMPLES :

To dump the congestion control params

```
niccli -i 2 ccparams -d
niccli -i 2 ccparams --dump
```

To configure the congestion control params

```
niccli -i 3 ccparams --set -f
```

BCM957608-P2200GQF00_congestion_control_20250210_025627_19427.CFG

Nic Information commands

=====

DESCRIPTION :

This command will display all the basic details of the device

SYNTAX :

```
niccli --list
niccli -l | --list_devices
niccli -e | --list_ethernet
show -d | --device_info
show -p | --pkg_ver [-f <firmware package file(s)>]
show -D | --device_pci_ids [-f <firmware package file(s)>]
show -c | --certificate [<-s | --slot number>]
show -n | --nvm_measurement
show --all
show --health
show -g | --pcb_gen2_otp
```

OPTIONS :

- p | --pkg_ver : Display firmware package version installed on the device or in the package file.
- f : Display firmware package file information.
- D | --device_pci_ids : Display Broadcom device id information.
- d | --device_info : Display the basic details of the device
- all : Display all the details of the device.
- c | --certificate : Display the imported certificate chain on the device.
This command is supported on BCM9575xxx and BCM9576xxx devices.
- s | --slot number : Slot number is where the certificate chain on the device is imported. The valid values are from 0 to 7. The default value is 0.
- n | --nvm_measurement : Display whether the NVM configuration is active in the system.

has been changed or not.To facilitate this, a hash is generated based on nvm configuration.The hash represents the measurement of the configuration.This command is supported on BCM9575xxx and BCM9576xxx devices.
- health : Display the device health.
- g | --pcb_gen2_otp : Display the PCB gen2 device OTP. This command is supported on BCM9574xxx devices.

EXAMPLES :

To list all the supported devices with target indexes

```
niccli --list
```

To list all the supported devices with basic information

```
niccli -l
```

```
niccli --list_devices
```

To list all the supported device interface names

```
niccli -e
```

```
niccli --list_ethernet
```

To display firmware package version installed on the device or in the package file

```
niccli -i 2 show -p -f BCM957508-N2100G.pkg
```

```
niccli -i 2 show --pkg_ver
```

To display the basic details of the device

```
niccli -i 1 show -d
```

```
niccli -i 1 show --device_info
```

To display the device health of the interface

```
niccli -i 2 show --health
```

To display the imported certificate chain on the device

```
niccli -i 1 show -c -s 1
```

```
niccli -i 1 show --certificate --slot 0
```

To display whether the NVM configuration that is active in the system

```
niccli -i 5 show -n
```

```
niccli -i 5 show --nvmmeasurement
```

To display Broadcom device id information

```
niccli -i 1 show -D -f BCM957508-N2100G.pkg
```

```
niccli -i 1 show --device_pci_ids -f BCM957608-P2200GQF00.pkg
```

To display all the details of the device

```
niccli -i 3 show --all
```

To display the PCB gen2 device OTP.

```
niccli -i 3 show -g
```

```
niccli -i 3 show --pcb_gen2_otp
```

/* End of file */