ibdiagnet InfiniBand Fabric Diagnostic Tool User Manual v2.21

Table of contents

Overview

ibdiagnet utility is one of the basic tools for InfiniBand fabric discovery, error detection and diagnostic. The output files of the ibdiagnet include error reporting, switch and HCA configuration dumps, various counters reported by the switches and the HCAs. Parameters of various devices such as switch fans, switch PSUs, cables, and PCI lanes are reported by the ibdiagnet as well. ibdiagnet also performs Unicast Routing, Adaptive Routing and Multicast Routing validation for correctness and credit-loop free routing.

ibdiagnet is distributed as part of ibutils2 package which is part of MLNX_OFED and UFM software packages. It is also available from the website as part of InfiniBand management package for Inbox customers.

Software Download

Please visit InfiniBand Management Tools page.

ibdiagnet Functionality

Functionality	Description
Fabric Discovery	Sweeps the InfiniBand fabric and collects information from the following InfiniBand devices: • Switches • HCAs • Routers • Aggregation Nodes • Gateways
Duplicated GUIDs Detection	Checks and reports duplicated Node and Port GUIDs in the fabric.
Duplicate Node Description Detection	Checks and warns regarding duplicated node description of switches or HCAs.
Alias GUIDs Check	Performs alias GUID checks (Relevant for ConnectX-3 devices only)
LIDs Check	Performs correct LID assignment and duplicated LID check for InfiniBand devices.
Links in INIT State and Unresponsive Nodes Detection	Reports links in INIT logical state. Additionally, it reports unresponsive devices and Direct Route to such devices.
Split Cables Support	Reports split cables in ibdiagnet dumps and logs.
Counters Fetch	Fetches various counters from InfiniBand devices, including standard and extended port

Functionality	Description
	counters, diagnostic counters, phy counters, etc.
Error Counters Check	Checks errors counters crossing thresholds between counter snapshots.
Routing Fetch and Checks	Performs correctness check of the switch routing tables as well as checking of credit-loop free routing.
Link Width and Speed Checks	Checks that fabric links are operating at maximum supported speed and width.
Dumping Virtualization Information	Dumps virtualization information from channel adaptors.
Dumping SHARP Trees Structures and SHARP Counters	
Dumping SHIELD Configuration and Counters	
Topology Matching	Performs matching of fabric topology with previously stored one.
Fast Discovery	Avoids rediscovering the fabric by using previously cached fabric data.
Support IB Security	Using MKEY, VSKEY, AMKEY and CC KEY for dumping fabric configuration.
Partition Checks	Dumps and validates HCA and switch partition tables.
BER Test	Reports links with high Bit Error Rate (BER).
Dump PCI Data from Servers	
Dump Cable Info	
Dump PHY Info	

ibdiagnet Commands

The chapter contains the following sections:

Basic Commands

Command	Description
--aguid	Run Alias GUID stage.
--am_key <am_key>	Specifies constant SHARP am_key for the fabric.
--am_key_file <path_to_am_key_file>	Specifies the path to the SHARP am_key_file: guid2am_key
--back_compat_db <version.sub_version>	Shows ports section in "ibdiagnet2.db_csv" according to given version. Default version 2.0. (0 - latest version)
--ber_test	Provides a BER test for each port. Calculate BER for each port and check no BER value has exceeded the BER threshold. (default threshold="10^-12"). This option applies for SwitchX/ConnectX-4/ConnectX-3 devices only. For later devices use --get_phy_info for BER validation.
--ber_thresh <value>	Specifies the threshold value for the BER test. The reciprocal number of the BER should be provided. Example: for 10^-12

Command	Description
	then value needs to be 1000000000000 or 0xe8d4a51000 (10^12). If threshold given is 0 then all BER values for all ports will be reported. This option applies for SwitchX/ConnectX-4/ConnectX-3 devices only
-c | --create_config_file <config-file>	Creates template configuration file.
--clear_congestion_counters	Displays Congestion Counters and clear them - this option also activate congestion_control option.
--fast_recovery	Display Fast Recovery info.
--config_file <config-file>	Configuration file.
--congestion_control	Displays Congestion Control info.
--congestion_counters	Displays Congestion Counters - this option also activate congestion_control option.
--dbg_levels	Verbosity levels to be applied on the debug log file. Possible values are: • 0x01 - Error • 0x02 - Info • 0x04 - MAD • 0x08 - Discover • 0x10 - Debug • 0x20 - Funcs • 0x80 - Sys • 0xff - ALL
--dbg_modules	Comma separated Module's names to be added to the debug log file. Possible values are:

Command	Description
	• IBIS, IBDIAG, IBDM, IBDIAGNET, ALL
--dfp	Provides a report of the fabric Dragonfly+ analysis.
--dfp_opt <max_cas=num>	Coma separated Dragonfly+ options (if --dfp option selected): • max_cas: maximal number of CAs on a switch to be counted as Dragonfly+ spine. This parameter is mutually exclusive with -smdb

--enable_output <files types list | csv section name>	Enables output for files and csv sections.
	• CSV section should have prefix 'csv:'
	  ◦ Examples of csv sections see in '.db_csv' file
	• Examples type of files (by file extensions):
	  ◦ lst|sm|pm|nodes_info|fdbs|mcfdbs|debug|pkey|aguid|slvl|vl2vl|plft|ar|far|rn|rnc|rnc2|mlnx_cntrs|net_dump|vports|vports_pkey|sharp|sharp_an_info|sharp_pm|cables|port_attr|net_dump_ext|db_csv
	• Specific reserved types:
	  ◦ <default|csv:default> : Will disabled by default for types wasn't set.
	  ◦ <all|csv:all> : Will disabled for all, ignore any specified value for file or csv section.

Command	Description
--discovery_only	Dumps only db_csv output file with discovery
--disable_output <files types list|csv section name>	Disables output for files and csv sections. • CSV section should have prefix 'csv:' ◦ Examples of csv sections see in '.db_csv' file • Examples type of files (by file extensions): ◦ lst|sm|pm|nodes_info|fdbs|mcfdbs|debug|pkey|aguid|slvl|vl2vl|plft|ar|far|rn|rnc|rnc2|mlnx_cntrs|net_dump|vports|vports_pkey|sharp|sharp_an_info|sharp_pm|cables|port_attr|net_dump_ext|db_csv • Specific reserved types: ◦ <default|csv:default> : Will enabled by default for types wasn't set. ◦ <all|csv:all> : Will enabled for all, ignore any specified value for file or csv section.
--enable_spst	Skips switch down ports while discover the fabric - use Switch Port State Table of the switch (enabled by default) - Deprecated
--enable_switch_dup_guid	Enables duplicated switch GUIDs detection while discover the fabric
--exclude_scope <file.guid>	The file with a list of Node-GUIDs and their ports to be excluded from the scope. The ibdiagnet2.ibnetdiscover file will not be generated.

Command	Description
--extended_speeds <dev-type>	Collects and tests port extended speeds counters. dev-type: • sw | all | none
-f | --load_from_file <path to ibdiagnet2.db_csv file>	Loads ibdiagnet.db_csv from external file. Use this option to skip discovery stage.
--ft	Provides a report of the fabric Fat-Tree analysis.
--ft_roots_regex_opt	Regular expression to select Fat-Tree root nodes. To be applied to a node's description.
-g | --guid <GUID in hex>	Specifies the local port GUID value of the port used to connect to the IB fabric. If GUID given is 0 then ibdiagnet displays a list of possible port GUIDs and waits for user input.
--gmp_window <num>	Max gmp MADs on wire. (default=128).
-h | --help	Prints help information (including plugins help if exists).
-i | --device <dev-name>	Specifies the name of the device of the port used to connect to the IB fabric (in case of multiple devices on the local system).
--llr_active_cell <0|64|128>	Specifies the LLR active cell size for BER test, when LLR is active in the fabric. (0 - not specified). This option applies for SwitchX/ConnectX-4/ConnectX-3 devices only
--ls <0|2.5|5|10|14|25|50|100|200|FDR10>	Specifies the expected link speed. (0 - disable expected link speed)
--lw <0|1x|2x|4x|8x|12x>	Specifies the expected link width. (0 - disable expected link width)

Command	Description
-m | --map <map-file>	Specifies mapping file, that maps node guid to name (format: 0x[0-9a-fA-F]+ "name"). Mapping file can also be specified by environment variable "IBUTILS_NODE_NAME_MAP_FILE_PATH".
--security_keys <path to key file>	Specifies the path to the directory with the key files (guid2lid, guid2mkey, neighbors, guid2vskey, guid2cckey, guid2_m2n_key, guid2_pm_key).
--m_key <m_key>	Specifies constant m_key for the fabric.
--mads_retries <mads-retries>	Specifies the number of retries for every timeout mad. (default=2).
--mads_timeout <mads-timeout>	Specifies the timeout (in milliseconds) for sent and received mads. (default=500).
--max_hops <max-hops>	Specifies the maximum hops for the discovery process. (default=64).
-o | --output_path <directory>	Specifies the directory where the output files will be placed.
--out_ibnl_dir <directory>	The topology file custom system definitions (ibnl) directory.
-P | --counter <cntr=threshold,... | all=threshold>	If any of the provided counter is greater than its provided value, then print it. If 'all' used - all counters get same threshold (0 by default).
-p | --port <port-num>	Specifies the local device's port number used to connect to the IB fabric.
--path <files types list>=<path>	Sets custom path for specific files. • Specific reserved types:

Command	Description
	• <default> : Will set path by default for types wasn't set. <all> : Will set path for all, ignore any specified value for file or csv section.
--pc	Resets all fabric IB spec compliant port counters (PortCounters and PortCountersExtended) RN, AR and HBF counters.
--per_slvl_cntrs	Provides a report of all per sl/vl port counters
--pm_pause_time <seconds>	Specifies the seconds to wait between first counters sample and second counters sample. If seconds given is 0 then no second counters sample will be done. (default=1).
--pm_per_lane	Lists all counters per lane (when available).
--pm_get_all	Get all PM counters. Activate the following flags: --per_slvl_cntrs --sc --extended_speeds all --pm_per_lane
--pm_clear_all	Clear all PM counters. Activate the following flags: --scr --pc
--qos	Displays qos config sl.
-r | --routing	Provides a report of the fabric qualities.
--r_opt	Comma separated routing options: (if -r option is selected) • skip_vs: Skip collect and check vendor specific routing settings like AR and PLFT.

Command	Description
	• skip_far: Deprecated! please use '--disable_output far' to skip dumping full ar tables data to file.Skip dump full ar tables data to file. • sl=<sl_num>: SL number to be used for ar connectivity and credit loop check. • check_sl: Check all SL2VL tables. SL should not be mapped to VL15. • mcast: Multicast credit loop check. It is recommended to use this option with sa_dump. • dump_only: Dump routing configuration files and skip routing checks. • dump_only_skip_routing_tables: Dump routing data and skip routing tables (LFTs) retrieving. • static_ca2ca: Run also static CA to CA routing check even if AR enabled.
--rail_validation	Checks topology being rail optimized (default - disabled).
--rail_validation_opt <regex='regular expression'>	Comma separated Rail Optimized Validation options (if --rail_validation option selected): • regex: regular expression to filter HCA nodes from reports. To be applied to HCAs node descriptions.
--read_capability <file name>	Specifies capability masks configuration file, giving capability mask configuration for the fabric. ibdiagnet will use this mapping for Vendor Specific MADs sending.

Command	Description
--sa_dump <file>	Specifies opensm-sa.dump file path, multicast groups definition generated by SM. used for mcast credit loop check (if -r option selected and r_opt=mcast).
--sc	Provides a report of Mellanox counters
--scope <file.guid>	The file with a list of Node-GUIDs and their ports to be left in the scope. The ibdiagnet2.ibnetdiscover file will not be generated.
--scr	Resets all the Mellanox counters (if -sc option selected).
--screen_num_errs <num>	Specifies the threshold for printing errors to screen. (default=5).
--sharp	Collects SHARP configuration. Check and dump to file.
--sharp_control_version < 0|1|2 >	Checks and dumps only SHARP nodes with the specified version (default 0 - all nodes).
--sharp_opt <[csc][dsc][dscp][ad_hoc][trees]>	Comma separated sharp options: (if --sharp option selected) • csc: Clear sharp counters. • dsc: Dump sharp performance counters to db_csv file. This option is for debug • dscp: Dump sharp HBA performance counters per port to db_csv file. This option is for debug • ad_hoc: Indicates that SHARP support ad-hoc trees, avoid warnings for tree_id duplication in the fabric

Command	Description
	• trees: Perform sharp trees validation
--pm_key <pm_key>	Specifies constant pm_key for the fabric.
--vs_key <vs_key>	Specifies constant vs_key for the fabric.
--cc_key <cc_key>	Specifies constant cc_key for the fabric.
--m2n_key	Specifies constant m2n_key for the fabric.
--skip <stage>	Skips the executions of the given stage. Applicable skip stages: • dup_guids | dup_node_desc | lids | sm | nodes_info | pkey | aguid | links | pm | speed_width_check | temp_sensing | virt | hca_cfg_check | all.
--skip_plugin <library name>	Skip the load of the given library name. Applicable skip plugins: • libibdiagnet_cable_diag_plugin-2.1.1 • libibdiagnet_phy_diag_plugin-2.1.1.
--sl <sl>	Specifies the SL to be used for QP1 MADs. (default=0).
--smdb <path to SMDB file>	Loads Routing Engine and Ranks from the User Subnet Manager SMDB file. Used for: • Adaptive Routing validation (if -r option selected)

Command	Description
	• Dragonfly+ validation (if --dfp option selected) • Fat-Tree validation (if –ft option selected)
--smp_window <num>	Max smp MADs on wire. (default=8).
-t | --topo_file <file>	Specifies the topology file name.
-V | --version	Prints the version of the tool.
--ppcc <filename|path|pattern>	Enables fetching PPCC (Port Programable Congestion Control) counters. Possible values: 1. File path - ibdiagnet loads from file PPCC Algorithms 2. Folder path - ibdiagnet loads all files from the directory 3. Wildcard - ibdiagnet loads files according to the wildcard matching (**Note:** In this case quotation marks must be used!). For more information on the supported wildcard syntax refer to the manual page by typing 'man 7 glob'
--vlr <file>	Specifies opensm-path-records.dump file path, src-dst to SL mapping generated by SM plugin. ibdiagnet will use this mapping for MADs sending and credit loop check (if -r option is selected).
-w | --write_topo_file <file name>	Writes out a topology file for the discovered topology.
--write_capability <file name>	Writes out an example file for capability masks configuration, and also the default capability masks for some devices.

Command	Description
--host_file <file name>	Specifies a file of hostnames to be applied for a scope.

Cable Diagnostic (Plugin)

> **Note**
>
> The plugin is deprecated and will be entirely removed in the next version.

Command	Description
--get_cable_info	Indicates to query all QSFP cables for cable information. Cable information will be stored in "ibdiagnet2.cables".
--cable_info_disconnected	Get cable info on disconnected ports. used only with get_cable_info flag.

Phy Diagnostic (Plugin)

Command	Description
--get_phy_info	Query all ports for phy information.
--get_ppamp	Query all ports for PPAMP (Port Phy opAMP data), works with --get_phy_info.
--show_cap_reg	Dump capability registers (hidden by default), works with --get_phy_info.

Command	Description
--reset_phy_info	Query and clear all ports phy information.
--get_p_info	Query HCA ports for PCI information.
--reset_p_info	Query and clear HCA ports PCI information.

ibdiagnet Features Overview

The chapter contains the following sections:

- Fat-Tree Topology Validation
- IBDIAGNET Output Control
- Write and Read Capability Files (Deprecated)
- Useful Options
- Debug Mode
- Validation of SM configuration for HCAs
- Security Keys

Getting ibdiagnet Version

The following command should be used to obtain ibdiagnet version:

Command:

```
ibdiagnet --version
```

or

```
ibdiagnet -V
```

Output:

```
I IBDIAGNET 2.5.0.MLNX20201108.8d5c3c9
I Dependencies: "IBDIAG 2.1.1.8d5c3c9","IBDM 2.1.1.8d5c3c9","IBIS
2.1.1.f10f7e1"
```

Running ibdiagnet without Parameters

Running ibdiagnet without command line parameters will perform the following fabric diagnostics:

- Fabric Discovery
- Duplicated GUIDs check
- Duplicated Node Description Check
- LID Check
- Links Check
- Subnet Managers Check
- Port Counters Snapshot/Checks in One Sec Period
- Nodes Information Check (Uniform firmware versions across the fabric. etc)
- Speed/Width Check
- Dump Virtualization Information
- Partition Keys Checks
- Dump Temperature Sensing
- Create Network Dump File Similar to the ibnetdiscover Format

Using Configuration File

All ibdiagnet options can be specified in configuration file prior to running ibdiagnet command. Default configuration file is located under the following: `- /etc/ibutils2/ibdiag.conf.` If the `"--config_file"` option is not specified, but the configuration file exists in the default location, the configuration options defined in the file will be applied if not overridden by specific options in the ibdiagnet command line.

Example:

`/etc/ibutils2/ibdiag.conf` includes the following line:

```
map = /etc/opensm/ib-node-name-map
```

Running ibdiagnet without the command line parameters will assume the `"--map"` option parameter is taken from the default configuration file.

Parameter	Description
--config_file <filename>	Loads configuration from file.
-c | --create_config_file <filename>	Creates a template configuration file.

Selecting InfiniBand Interface

This section explains how to select specific IB interface and port for running ibdiagnet on it. If the options below are not specified, first active IB interface will be used by ibdiagnet.

Parameter	Description
-i | -device <dev-name>	Specifies the name of the device of the port used to connect to the InfiniBand fabric (in case of multiple devices on the local system).
-p | -port <port-num>	Specifies the local device's port number used to connect to the InfiniBand fabric.
-g | -guid <GUID in hex>	Specifies the local port GUID value of the port used to connect to the InfiniBand fabric. If GUID given is 0, then ibdiagnet displays a list of possible port GUIDs and waits for user input.

Example:

```
ibdiagnet --i mlx5_2 --p 1
```

InfiniBand MAD-Specific Parameters

This section specifies InfiniBand parameters related to MAD sending.

Parameter	Description
--mads_timeout <mads-timeout>	Specifies the timeout (in milliseconds) for r waiting mad responses (default=500).
--mads_retries <mads-retries>	Specifies the number of mad retries in case of mad timeout (default=2).
--smp_window <num>	Specifies maximum number of QP0 MADs on wire (default=8).
--gmp_window <num>	Specifies maximum number of QP1 MADs on wire (default=128).
--max_hops <max-hops>	Specifies the maximum hops for the discovery process (default=64).
--sl <value>	Specifies the SL to be used for QP1 MADs (default=0).

Example:

```
ibdiagnet --mads_timeout 1000 --mads_retries 3 --gmp_window 128 --gmp_window 4096 --max_hops 64
```

Fabric Discovery

The section specifies additional options applicable to the fabric discovery stage of ibdiagnet.

Parameter	Description	Example
--enable_switch_dup_guid	By default, ibdiagnet does network discovery without checking duplicated switch GUIDs. To allow duplicated switch GUIDs check, this option should be specified.	`ibdiagnet --enable_switch_dup_guid`
--enable_spst	**Deprecated.** Speeds up fabric discovery by detecting down ports from switch's "Switch Port State Table". A switch must support the table. Otherwise will have no effect.	`ibdiagnet --enable_spst`
--discovery_only	Discovers a network and dumps it into	`ibdiagnet --discovery_only`

Parameter	Description	Example
	ibdiganet2.db_csv file. This file may be used later with --load_from_file option.	

Running ibdiagnet without Fabric Discovery

The following ibdiagnet option can be used to avoid fabric discovery and speed up fabric diagnostics.

Parameter	Description	Notes

-f | --load_from_file <path to ibdiagnet2.db_csv file>	Loads ibdiagnet db_csv from file to skip fabric discovery. HCAs and switches configuration will be obtained from the provided file. The input file must be a valid ibdiagnet db_csv file.	• The following command should be run prior to ibdiagnet -f invocation:
		`ibdiagnet --discovery_only -o /tmp`
		• Once valid ibdiagnet.db_csv is created in /tmp directory, one can run:
		`ibdiagnet -f /tmp/ibdiagnet2.db_csv`

Fabric Links Validation

Link Speed Validation

The following options can be used to verify that all fabric links has the same speed and width (if applicable). Ports with degraded speed or width are reported in ibdiagnet.log file.

Parameter	Description	Example	Notes

--ls <0|2.5|5|10|14|25|50|100|FDR10>	Specifies expected link speed SDR/DDR/QDR/FDR/EDR/HDR//NDR/FDR10	`ibdiagnet --ls 50`	The following error will be reported for non-HDR links:
			`-E- Link: Sec0d9a0300246e30/Nec0d9a0300246e30/P19<-->switch-004/U1/P20 - Unexpected actual link speed 25`
			`-E- Link: Sec0d9a0300246e30/Nec0d9a0300246e30/P20<-->switch-004/U1/P16 - Unexpected actual link speed 25`

Link Width Validation

Parameter	Description	Example	Notes
--lw <1x|2x|4x|8x|12x>	Specifies expected link width.	`ibdiagnet --lw 2x`	The following error will be reported for non-2x links: `-E- Link: Sec0d9a0300246e30/Nec0d9a0300246e30/P1<-->Sec0d9a0300246fb0/Nec0d9a0300246fb0/P4 - Unexpected width, actual link width is 4x-E- Link: Sec0d9a0300246e30/Nec0d9a0300246e30/P2<-->Sec0d9a0300246fb0/Nec0d9a0300246fb0/P3 - Unexpected width, actual link width is 4x`

Port Counters

ibdiagnet collects and processes standard InfiniBand port counters and vendor-specific port counters. The following counters are collected by the ibdiagnet:

- PortCounters (collected by default)
- PortCountersExtended (collected by default)
- PortRcvErrorDetails (collected by default)
- PortXmitDiscardDetails (collected by default)
- LLRCounters (collected by default from supporting devices, ConnectX3/SwicthX only)
- PerSL/VL counters (for supporting devices when corresponding option is specified)
- PortExtendedSpeedCounters (for supporting devices when corresponding option is specified)
- Mellanox Diagnostic Counters (for supporting devices when corresponding option is specified)

Port Counter Types

The following options are applicable when port counters are collected and processed by ibdiagnet:

Parameter	Description	Example
--per_slvl_cntrs	Provides a report of all per sl/vl port counters (if supported by devices) These counters are reported in ibdiagnet2.db_csv file.	-

Parameter	Description	Example
--sc	Provides a report of NVIDIA Diagnostic counters in ibdiagnet2.mlnx_cntrs file and ibdiagnet2.db_csv.	-
--scr	Resets all the NVIDIA Diagnostic counters (should be used with -sc option).	`ibdiagnet --scr --sc`
--extended_speeds <dev-type>	Collects and tests port extended speeds counters. Supported dev-type: • sw (switch only) • all (all devices) These counters are reported in ibdiagnet2.db_csv file (PM_INFO section).	-
--pm_per_lane	Lists all counters per lane (if supported by	`ibdiagnet --extended_speeds all --pm_per_lane`

Parameter	Description	Example
	devices). Should be used on combination with --extended_speeds.	
--pm_get_all	Get all PM counters. activate the following flags: --per_slvl_cntrs --sc --extended_speeds all --pm_per_lane	ibdiagnet --pm_get_all
-P | -counter <<PM>=<value>>	If any of the provided counter is greater than its provided value, then print it. If 'all' is used, all counters get the same threshold (0 by default).	`ibdiagnet -P vl15_dropped=1, port_xmit_discard=1` or `ibdiagnet -P vl15_dropped=1 -P port_xmit_discard=1` or `ibdiagnet -P all` Supported PM Counter names are: • symbol_error_counter • port_rcv_remote_physical_errors • port_rcv_errors • port_xmit_discard • port_rcv_switch_relay_errors • vl15_dropped • link_error_recovery_counter • link_down_counter • port_xmit_constraint_errors • port_rcv_constraint_errors • local_link_integrity_errors • excessive_buffer_errors • port_xmit_data

Parameter	Description	Example
		• port_rcv_data • port_xmit_pkts • port_rcv_pkts • port_xmit_wait • port_xmit_data_extended • port_rcv_data_extended • port_xmit_pkts_extended • port_rcv_pkts_extended • port_unicast_xmit_pkts • port_unicast_rcv_pkts • port_multicast_xmit_pkts • port_multicast_rcv_pkts • sync_header_err_cnt • unknown_block_cnt • error_detection_counter_lane0 • error_detection_counter_lane1 • ..... • error_detection_counter_lane11 • fec_correctable_block_counter_lane0 • fec_correctable_block_counter_lane1 • ..... • fec_correctable_block_counter_lane11 • fec_uncorrectable_block_counter_lane0 • fec_uncorrectable_block_counter_lane1 • ..... • fec_uncorrectable_block_counter_lane11 • port_rcv_cells • port_rcv_cell_for_retry • port_rcv_retry • port_xmit_cells • port_xmit_retry_cells • port_xmit_retry • port_symbol_error • port_error_detection_counter_lane0 • ..... • port_error_detection_counter_lane3 • max_retransmission_rate • retransmission_per_sec • fec_corrected_symbol_counter_lane0 • fec_corrected_symbol_counter_lane1 • ..... • fec_corrected_symbol_counter_lane11 • port_fec_correctable_block_counter

Parameter	Description	Example
		• port_fec_uncorrectable_block_counter • port_fec_corrected_symbol_counter • all

Port Counters Reset

Parameter	Description
--pc	Resets all fabric IB spec compliant port counters (PortCounters and PortCountersExtended), RN, AR and HBF counters. **Note:** It is recommended to use this option with `-reset_phy_info`, as both options have cross counters and using only one of them can be confusing on the next iteration of counters or registers collection.
--pm_clear_all	Clear all PM counters. activate the following flags: --scr --p

Port Counters Delta Validation

Parameter	Description	Example
--pm_pause_time <seconds>	Specifies a delay (in seconds) between counters samples. If set to 0, only single sampling is performed.	`ibdiagnet --pm_pause_time 60`

Parameter	Description	Example
	(default - 1 second) The delta between the first and the second counter samples will be written to the PM_DELTA section in db_csv file.	

Alias GUIDs

The below option allows retrieving assigned alias GUIDs from a channel adapter, a router and switch management ports (if supported).

Parameter	Description
--aguid	Run Alias GUID stage and dump its data to ibdiagnet2.db_csv and ibdiagnet2.aguid.

Example:

```
ibdiagnet --aguid
```

Output ibdiagnet2.db_csv:

```
START_AGUID
NodeGUID,PortGUID,PortNum,Index,AGUID
0xe41d2d0300a5f21a,0xe41d2d0300a5f21a,1,0,0xe41d2d0300a5f21a
0xb8599f03004c40dc,0xb8599f03004c40dc,0,0,0xb8599f03004c40dc
```

```
0xb8599f0300e9056e,0xb8599f0300e9056e,0,0,0xb8599f0300e9056e
0x98039b030067a4bd,0x98039b030067a4bd,1,0,0x98039b030067a4bd
0x98039b030067a4bc,0x98039b030067a4bc,1,0,0x98039b030067a4bc
END_AGUID
```

Output ibdiagnet2.aguid:

```
Port Name=pdory/U1/P1, Primary GUID=0xe41d2d0300a5f21a
        alias guid=0xe41d2d0300a5f21a

Port Name=Sb8599f03004c40dc/Nb8599f03004c40dc/P0, Primary
GUID=0xb8599f03004c40dc
        alias guid=0xb8599f03004c40dc

Port Name=Sb8599f0300e9056e/Nb8599f0300e9056e/P0, Primary
GUID=0xb8599f0300e9056e
        alias guid=0xb8599f0300e9056e

Port Name=dory04/U6/P1, Primary GUID=0x98039b030067a4bd
        alias guid=0x98039b030067a4bd
```

Topology Comparison

Topology comparison is used to check that actual fabric matches the designed topology.

Below is a list of parameters controlling topology validation.

Parameter	Description
-w | --write_topo_file <file>	Writes out a topology file for the discovered topology.
-t | --topo_file <file>	Specifies the topology file name. Provided topology file will be compared against the

Parameter	Description
	discovered topology. Any mismatch between the two topologies will be reported in the log file.
--out_ibnl_dir <dir>	Directory for additional custom system definitions (IBNL) files. Ibdiagnet will add IBNL files to this folder in the 'write topo' stage for any node from the discovered topology that does not have a file in ibdm/ibnl. ibdiagnet will use this folder as an input folder in the 'read topo' stage for any node in the given topology file that does not have a file in ibdm/ibnl.

Topology example:

```
HCA_3 dgx1
   U1/P1  -4x-25G-> MSB7700 dingo-200 P1
   U2/P1  -4x-25G-> MSB7700 dingo-200 P6
   U3/P1  -4x-25G-> MSB7700 dingo-200 P9

MSB7700 dingo-200
   P1 -4x-25G->  HCA_3  dgx1  U1/P1
   P6 -4x-25G->  HCA_3  dgx1  U2/P1
   P9 -4x-25G->  HCA_3  dgx1  U3/P1
```

IBNL example:

```
TOPSYSTEM HCA_3
```

```
NODE CA 2 MT4099 U1
   1 -> U1/P1
   2 -> U1/P2

NODE CA 2 MT4099 U2
   1 -> U2/P1
   2 -> U2/P2

NODE CA 2 MT4099 U3
   1 -> U3/P1
   2 -> U3/P2
```

Routing Validation

The following options should be used to enable Static/Adaptive and Multicast routing validation in the InfiniBand fabric, potential credit-loops detection, and Adaptive Routing configuration validation. In some cases, routing validation options should be specified to perform additional routing diagnostics.

Basic Routing Diagnostics

Parameter	Description
-r | -routing	ibdiagnet performs unicast (Static and Adaptive) and Multicast Routing validation, calculates and reports: • The number of CA pairs that are in each number of hops distance • Number of actual paths going through each switch out port considering all the CA-to-CA paths • Number of actual Destination LIDs going through each switch out port considering all the CA-to-CA paths • Scanning multicast routing tables for loops and connectivity • Applies credit-loop detection algorithm

Parameter	Description
	• Applies adaptive routing configuration validation - checking AR LFTs against up-down min-hop tables (if -smdb option is used).
	Switch routing tables are dumped to the following files: Switch routing tables are dumped to the following files: VL2VL configuration: /var/tmp/ibdiagnet2/ibdiagnet2.vl2vl PLFT dump: /var/tmp/ibdiagnet2/ibdiagnet2.plft AR/SHIELD tables dump: /var/tmp/ibdiagnet2/ibdiagnet2.far AR(FLIDs) tables dump: /var/tmp/ibdiagnet2/ibdiagnet2.far_flid (the file is dumped only if FLIDs are enabled) Unicast tables dump: /var/tmp/ibdiagnet2/ibdiagnet2.fdbs Multicast tables dump: /var/tmp/ibdiagnet2/ibdiagnet2.mcfdbs SLVL Table dump: /var/tmp/ibdiagnet2/ibdiagnet2.slvl

Example:

```
ibdiagnet -r
```

Output:

```
##################################
-I- Fabric Qualities Report:
##################################
-I- Verifying all CA to CA paths ...
---------------------- CA to CA : LFT ROUTE HOP HISTOGRAM -------
----------
```

```
The number of CA pairs that are in each number of hops distance.
This data is based on the result of the routing algorithm.

HOPS NUM-CA-CA-PAIRS
  2   24
  3   30
  4   78
  5   22
  6   56
-----------------------------------------------------------------
----------

---------- LFT CA to CA : SWITCH OUT PORT - NUM PATHS HISTOGRAM -
----------
Number of actual paths going through each switch out port
considering
all the CA to CA paths. Ports driving CAs are ignored (as they
must
have = Nca - 1). If the fabric is routed correctly the histogram
should be narrow for all ports on same level of the tree.

NUM-PATHS NUM-SWITCH-PORTS
       0   21
       1   4
       2   8
       3   6
       4   1
       5   6
       6   9
       7   6
       8   12
       9   2
      10   3
      11   6
      12   7
      14   1
```

```
-------------------------------------------------------------------
----------

---------- LFT CA to CA : SWITCH OUT PORT - NUM DLIDS HISTOGRAM -
----------
Number of actual Destination LIDs going through each switch out
port considering
all the CA to CA paths. Ports driving CAs are ignored (as they
must
have = Nca - 1). If the fabric is routed correctly the histogram
should be narrow for all ports on same level of the tree.
A detailed report is provided in
/tmp/ibdmchk.sw_out_port_num_dlids.

NUM-DLIDS NUM-SWITCH-PORTS
       0   21
       1   37
       2   34
-------------------------------------------------------------------
----------

-I- Scanned:210 CA to CA paths
-------------------------------------------------------------------
----------

-I- Scanning all multicast groups for loops and connectivity...
-I- Multicast Group:0xC000 has:7 switches and:9 FullMember ports
-I- Multicast Group:0xC001 has:7 switches and:9 FullMember ports
-I- Multicast Group:0xC002 has:7 switches and:9 FullMember ports
-I- Multicast Group:0xC003 has:7 switches and:8 FullMember ports
-I- Multicast Group:0xC004 has:6 switches and:3 FullMember ports
-------------------------------------------------------------------
----------


##################################
```

```
-I- Credit Loops Report:
##################################
-I- Analyzing Fabric for Credit Loops 1 SLs, 1 VLs used.
-I- Traced 186 unicast paths
-I- no credit loops found
```

Routing Validation Options

The following options can be used when the `"-r"` option is invoked.

Parameter	Description
--r_opt	List of comma-separated options: • skip_vs: Skip collect and check vendor specific routing settings like AR and PLFT. • skip_far: Deprecated! please use '--disable_output far' to skip dumping full ar tables data to file. • sl=<sl_num>: SL number to be used for ar connectivity and credit loop check. • check_sl: Check all SL2VL tables. SL should not be mapped to VL15. • mcast: Multicast credit loop check. It is recommended to use this option with sa_dump. • dump_only: Dump routing configuration files and skip routing checks. • dump_only_skip_routing_tables: Dump routing

Parameter	Description
	data and skip routing tables (LFTs) retrieving. • static_ca2ca: Run also static CA to CA routing check even if AR enabled.
--sa_dump <file>	Use Subnet Manager SMDB file for routing checks. If specified, Adaptive Routing validation is done during routing validation stage (if -r option selected)
--smdb <file>	Load Routing Engine and Ranks from SMDB file. Used for AR validation in routing stage (if -r option selected).
--vlr <file>	This option provides opensm-path-records.dump file that includes source-to-destination to SL mapping. This file is generated by dump_pr Subnet manager plugin. ibdiagnet will use this mapping for credit loop check. This option is mainly applicable in 3D-Torus topologies.

Example:

```
ibdiagnet -r --r_opt=vs,sl=2 --skip
pm,pkey,links,temp_sensing,speed_width_check,nodes_info,sm,dup_guid
```

Routers

This section specifies options for retrieving InfiniBand Routers configuration and verification of Floating LIDs (FLIDs) configuration on a local and adjacent networks. If FLIDs are enabled, two new files are generated:

- ibdiagnet2.arg2flid - represents distribution of "Adaptive Routing" groups per FLIDs on routers

- ibdiagnet.flid – contains details about networks FLIDs ranges, FLIDs enabled on the routers, and distribution of FLIDs per switches in the local network

Examples:

- **ibdiagnet's Output**: FLIDs are disabled/subnet does not have any router

```
Routers
-I- Build Routers Info DB finished successfully
-I- Build Routers Tables finished successfully
-I- Adjacent subnets FLID Table retrieving finished
successfully
-I- Routers FLID Table retrieving finished successfully
-I- Skipping FLID verification
```

- **ibdiagnet's Output**: FLIDS are enabled on some routers

```
Routers
-I- Build Routers Info DB finished successfully
-I- Build Routers Tables finished successfully
-I- Adjacent subnets FLID Table retrieving finished
successfully
-I- Routers FLID Table retrieving finished successfully
-I- AR Group To Router FLID Table retrieving finished
successfully

-I- All routers in the subnet have the same global FLID
range: start=10 end=1100
-I- All routers in the subnet have the same local FLID range:
start=20 end=500
-I- Local FLID range is in the global one.
-I- Local subnet LID and global FLID ranges are OK.
```

```
-I- Local subnet FLID verification finished successfully


-I- Ranges in the subnet: 0x233c are OK
-E- Adjacent subnets FLID verification finished with errors
-E- Different FLID ranges were detected for the subnet:
0x4789
[0x0002c90000000053, 0x0002c90000000053, 0x0002c9000000004a]
(total 3), start=583 end=667  -I- PFRN Settings verification
finished successfully

-I- HCA and Switches FLID verification finished successfully
-E- Remote subnets FLID verification finished with errors
-E- The remote FLID=1794 enabled on the router: name=SW-0-
0/RT GUID=0x0002c90000000041 does not belong to any range
defined on its adjacent subnets
```

- **flid file:** global and local subnets ranges

```
Global FLID range: start=10 end=1100
Local subnet FLID range: start=20 end=500
Local subnet LID and global FLID ranges are OK
```

- **flid file:** router with its subnets ranges and enabled FLIDs

```
0x0002c9000000004a - "SW-0-1/RT" pfrn_rtr_en=0,
ar_group_to_router_lid_table_cap=0, max_ar_group_id=0

#adjacent subnets by subnet prefix
0x0001(Local Subnet) FLID range: start=1934 end=1968,
pfrn_enabled=NA, max_ar_group_id=NA
0x0a54 FLID range: start=1969 end=2062, pfrn_enabled=NA,
max_ar_group_id=NA
```

```
0x37a9 FLID range: start=2063 end=2123, pfrn_enabled=NA,
max_ar_group_id=NA
0x9517 FLID range: start=2124 end=2210, pfrn_enabled=NA,
max_ar_group_id=NA

#enabled FLIDs by subnet prefix
0x0001(Local Subnet): 1934-1935
0x0a54: 1969, 1972, 1979-1982, 1984, 2141-2143, 2199-2205,
2208-2209
unclassified: 1792-1793, 1799, 1801, 1804-1805, 1808, 1811-
1814, 1816-1817
```

- **flid file:** FLIDs per switches

```
FLID per switches FLID per switches
0x0002c90000000068 - "SW-1-0/U1": 1934
0x0002c9000000006c - "SW-1-1/U1": 1934
0x0002c90000000070 - "SW-1-2/U1": 1935
0x0002c90000000074 - "SW-1-3/U1": 1935
```

- **flid file:** compression ratio histogram

```
compression_ratio     #flids
2                       2
```

Congestion Control

The following ibdiagnet options can be used to dump Mellanox/Nvidia Congestion Control configuration from HCAs/switches and Congestion Control Counters.

Parameter	Description
--congestion_control	Dumps Congestion Control configuration to the ibdiagnet2.db_csv file.
--congestion_counters	Dumps Mellanox/Nvidia Congestion Control Counters in ibdiagnet2.db_csv file. This option also activates congestion_control option. If in ibdiagnet configuration file the following are set, congestion counters will be collected: • congestion_counters is set to TRUE • congestion_control is set to FALSE
--clear_congestion_counters	Dumps Congestion Counters to the ibdiagnet2.db_csv file and clears them. This option also activate congestion_control option.
--ppcc <filename|path|pattern>	Enables fetching PPCC (Port Programable Congestion Control) counters. Possible values: 1. File path - ibdiagnet loads from file PPCC Algorithms. 2. Folder path - ibdiagnet loads all files from the directory. 3. Wildcard - ibdiagnet loads files according to the wildcard matching (**Note:** In this case, quotation marks must be used!). For more information on the supported wildcard syntax refer to the manual page by typing 'man 7 glob'

Example:

```
ibdiagnet --congestion_control
ibdiagnet --congestion_counters
ibdiagnet --clear_congestion_counters
ibdiagnet --congestion_counters --ppcc /tmp/file2.algo
ibdiagnet --congestion_control --ppcc '/tmp/*.algo'
```

Output Congestion Control:

```
START_CC_ENHANCED_INFO
NodeGUID,ver0Supported,CC_Capability_Mask
0x0002c9000000001d,1,0x0000000000000002
0x0002c9000000004f,1,0x0000000000000002
0x0002c90000000011,1,0x0000000000000002
END_CC_ENHANCED_INFO


START_CC_SWITCH_GENERAL_SETTINGS
NodeGUID,aqs_time,aqs_weight,en,cap_total_buffer_size
0x0002c9000000004f,0,0,0,0
0x0002c90000000041,0,0,0,0
0x0002c90000000043,0,0,0,0
END_CC_SWITCH_GENERAL_SETTINGS


START_CC_PORT_PROFILE_SETTINGS
NodeGUID,portNum,vl,mode,profile1_min,profile1_max,profile1_percen
0x0002c9000000004f,1,0,0,0,0,0,0,0,0,0,0,0
0x0002c9000000004f,1,1,0,0,0,0,0,0,0,0,0,0
0x0002c9000000004f,1,2,0,0,0,0,0,0,0,0,0,0
END_CC_PORT_PROFILE_SETTINGS
```

```
START_CC_SL_MAPPING_SETTINGS
NodeGUID,portNum,sl_profile_0,sl_profile_1,sl_profile_2,sl_profile_
0x0002c9000000004f,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0x0002c9000000004f,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0x0002c9000000004f,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
END_CC_SL_MAPPING_SETTINGS


START_CC_HCA_GENERAL_SETTINGS
NodeGUID,PortGUID,portNum,en_react,en_notify
0x0002c9000000001d,0x0002c9000000001e,1,0,0
0x0002c90000000011,0x0002c90000000012,1,0,0
0x0002c90000000015,0x0002c90000000016,1,0,0
END_CC_HCA_GENERAL_SETTINGS


START_CC_HCA_RP_PARAMETERS
NodeGUID,PortGUID,portNum,clamp_tgt_rate_after_time_inc,clamp_tgt_r
0x0002c9000000001d,0x0002c9000000001e,1,0,0,0,0,0,0,0,0,0,0,0,0,0
0x0002c90000000011,0x0002c90000000012,1,0,0,0,0,0,0,0,0,0,0,0,0,0
0x0002c90000000015,0x0002c90000000016,1,0,0,0,0,0,0,0,0,0,0,0,0,0
END_CC_HCA_RP_PARAMETERS


START_CC_HCA_NP_PARAMETERS
NodeGUID,PortGUID,portNum,min_time_between_cnps,cnp_sl,cnp_sl_mode
0x0002c9000000001d,0x0002c9000000001e,1,0,0,0
0x0002c90000000011,0x0002c90000000012,1,0,0,0
0x0002c90000000015,0x0002c90000000016,1,0,0,0
END_CC_HCA_NP_PARAMETERS


START_CC_HCA_STATISTICS_QUERY
NodeGUID,PortGUID,portNum,clear,cnp_ignored,cnp_handled,marked_pack
0x0002c9000000001d,0x0002c9000000001e,1,1,8438294795498567466,11946
```

```
0x0002c90000000011,0x0002c90000000012,1,1,13850774289226306924,671
0x0002c90000000015,0x0002c90000000016,1,1,13520874084649801659,313
END_CC_HCA_STATISTICS_QUERY
```

Output Port Programmable Congestion Control:

```
START_CC_HCA_ALGO_CONFIG_SUPPORT
NodeGUID,PortGUID,algo_en,algo_status,trace_en,counter_en,sl_bitmas
0x0002c9000000002d,0x0002c9000000002e,0,0,0,0,0x0186,8,15,32934,238
0x0002c90000000031,0x0002c90000000032,0,1,0,0,0x8bb4,15,1,10469,170
END_CC_HCA_ALGO_CONFIG_SUPPORT


START_CC_HCA_ALGO_CONFIG
NodeGUID,PortGUID,algo_slot,algo_en,algo_status,trace_en,counter_en
0x0002c9000000002d,0x0002c9000000002e,0,1,0,1,0,0xe96f,12,0,"Pi9Mrn
0x0002c9000000002d,0x0002c9000000002e,1,1,1,1,1,0xdd9f,8,13,"hERqon
END_CC_HCA_ALGO_CONFIG


START_CC_HCA_ALGO_CONFIG_PARAMS
NodeGUID,PortGUID,algo_slot,sl_bitmask,encap_len,encap_type,congest
0x0002c9000000002d,0x0002c9000000002e,0,0x78e1,8,0,2670514607,...,N
0x0002c9000000002d,0x0002c9000000002e,1,0x6fdb,15,8,852343172,...,N
END_CC_HCA_ALGO_CONFIG_PARAMS


START_CC_HCA_ALGO_COUNTERS
NodeGUID,PortGUID,algo_slot,clear,sl_bitmask,encap_len,encap_type,c
0x0002c9000000002d,0x0002c9000000002e,1,0,0xf1dd,13,8,939773111,..
0x0002c9000000002d,0x0002c9000000002e,2,0,0xb725,7,3,2936704535,..
END_CC_HCA_ALGO_COUNTERS
```

Fast Recovery

The following ibdiagnet option can be used to dump Fast Recovery configuration from the switches.

Parameter	Description
--fast_recovery	Display Fast Recovery info.

Example:

```
ibdiagnet --fast_recovery
```

Output Fast Recovery:

```
START_PROFILES_CONFIG
NodeGUID,PortNumber,PortProfile
0x08c0eb030096b524,0,0
0x08c0eb030096b524,1,1
0x08c0eb030096b524,2,1
0x08c0eb030096b524,3,1
0x08c0eb030096b524,4,1
0x08c0eb030096b524,5,1
0x08c0eb030096b524,6,1
0x08c0eb030096b524,7,1
0x08c0eb030096b524,8,1
END_PROFILES_CONFIG

START_CREDIT_WATCHDOG_CONFIG
NodeGUID,ProfileNum,en_thr,error_thr_action,en_normal_trap,en_warn
0x08c0eb030096b524,0,0,0,0,0,0,0,0,0,0,0
0x08c0eb030096b524,1,3,0,0,0,0,3,2,1,4,1
END_CREDIT_WATCHDOG_CONFIG
```

```
START_BER_CONFIG
NodeGUID,ProfileNum,BERType,en_thr,error_thr_action,en_normal_trap
0x08c0eb030096b524,0,0,0,0,0,0,0,0,0,0,0,0
0x08c0eb030096b524,0,1,0,0,0,0,0,0,0,0,0,0
0x08c0eb030096b524,0,2,0,0,0,0,0,0,0,0,0,0
0x08c0eb030096b524,1,0,2,0,0,0,0,0,0,0,0,0
0x08c0eb030096b524,1,1,0,0,0,0,0,0,0,0,0,0
0x08c0eb030096b524,1,2,0,0,0,0,0,0,0,0,0,0
END_BER_CONFIG
```

QoS Configuration: Rate Limit per Service Level (SL)

The following ibdiagnet option can be used to dump Rate Limit configuration from HCAs.

> **Note**
>
> HCA Rate Limit configuration is provided by the Subnet Manager once the `"enhanced_qos_policy_file"` Subnet Manager configuration file is populated with the Rate Limit policy rules.

Parameter	Description	Example
--qos	The Rate Limit per SL configuration is dumped to the ibdiagnet2.db_csv file under QOS_CONFI	`ibdiagnet --qos`

Parameter	Description	Example
	G_SL section.	

NVIDIA Scalable Hierarchical Aggregation and Reduction Protocol (SHARP) Support

The following ibdiagnet options can be used to dump SHARP Trees configuration and SHARP traffic counters.

Parameter	Description
--sharp	Collects SHARP Trees configuration and dump to ibdiagnet2.sharp & ibdiagnet2.sharp_an_info. • ibdiagnet.sharp contains SHARP distribution trees and tree QPs structures. • ibdiagnet2.sharp_an_info contains node information.
--sharp_control_version <0|1|2>	Checks and dumps only SHARP nodes with the specified version (default 0): • 0—all versions • 1—version 1 only • 2—version 2 only
--sharp_opt <[csc][dsc][dscp][ad_hoc]>	Comma separated SHARP options once "`--sharp`" option is selected: • csc: Clears SHARP counters.

Parameter	Description
	• dsc: Dumps SHARP performance counters to the ibdiagnet2.db_csv file. • dscp: Dumps SHARP SAT performance counters per port to the ibdiagnet2.db_csv file. • ad_hoc: Indicates that SHARP support ad-hoc trees, avoid warnings for tree_id duplication in the fabric

Examples:

- This example shows 3 level SHARP tree which root (rank 0) is Aggregation Node 0xec0d9a0300246f38.

```
TreeID:0, Max Radix:2
(0), AN:Mellanox Technologies Aggregation Node, lid:48, port
guid:0xec0d9a0300246f38, Child index:0, parent
QPn:0x00000000, remote parent QPn:0x00000000, radix:2
(1), AN:Mellanox Technologies Aggregation Node, lid:52, port
guid:0xec0d9a0300246f58, Child index:0, parent
QPn:0x008de801, remote parent QPn:0x008de801, radix:2
(2), AN:Mellanox Technologies Aggregation Node, lid:80, port
guid:0xec0d9a030027dbb8, Child index:0, parent
QPn:0x00fb6801, remote parent QPn:0x008de802, radix:0
(2), AN:Mellanox Technologies Aggregation Node, lid:32, port
guid:0xec0d9a0300090168, Child index:1, parent
QPn:0x00202801, remote parent QPn:0x008de803, radix:0
(1), AN:Mellanox Technologies Aggregation Node, lid:512, port
guid:0xec0d9a0300246e38, Child index:1, parent
QPn:0x008dc801, remote parent QPn:0x008de802, radix:2
```

```
(2), AN:Mellanox Technologies Aggregation Node, lid:108, port
guid:0xec0d9a030027dbd8, Child index:0, parent
QPn:0x00fb6801, remote parent QPn:0x008dc802, radix:0
(2), AN:Mellanox Technologies Aggregation Node, lid:144, port
guid:0xec0d9a03000b6bf8, Child index:1, parent
QPn:0x006d6801, remote parent QPn:0x008dc803, radix:0
```

- The example shows SHARP port counters:

```
-------------------------------------------------------
AggNodeDesc=Mellanox Technologies Aggregation Node Lid=32
GUID=0xec0d9a0300090168
-------------------------------------------------------
packet_sent=0x0000000000000000
ack_packet_sent=0x0000000000000000
retry_packet_sent=0x0000000000000000
rnr_event=0x0000000000000000
timeout_event=0x0000000000000000
oos_nack_rcv=0x0000000000000000
rnr_nack_rcv=0x0000000000000000
packet_discard_transport=0x0000000000000000
packet_discard_sharp=0x0000000000000000
```

Bit Error Rate (BER)

The Bit Error Rate (BER) is the number of bit errors per unit time divided by the total number of transferred bits during a studied time interval. BER is a unitless performance measure, often expressed as a percentage.

Parameter	Description	Notes
--get_phy_info	Collects BER information for fabric	Applicable to all EDR/HDR and future InfiniBand devices.

Parameter	Description	Notes
	ports and checks BER validating with specific thresholds. Errors will be reported to the ibdiagnet2.log and ibdiagnet2.db_csv files.	
--ber_test	**Deprecated.** Provides a BER test for each port. Calculate BER for each port and check no BER value has exceeded the BER threshold. (default threshold="10^-12").	This option is available only when using SwitchX/ConnectX-4 and ConnectX-3 devices.
--ber_thresh <value>	**Deprecated.** Specifies the threshold value for the BER test. The reciprocal number of the BER should be provided.	This option is available only when using SwitchX/ConnectX-4 and ConnectX-3 devices.

Parameter	Description	Notes
	For example, the value of 10^-12 should be 1000000000000 or 0xe8d4a51000 (10^12). If the given threshold is 0, then all BER values for all ports will be reported.	
--llr_active_cell <64|128>	**Deprecated.** Specifies the Link Level Retransmission (LLR) active cell size for BER test, when LLR is active in the fabric.	This option is available only when using SwitchX/ConnectX-4 and ConnectX-3 devices

Example:

```
ibdiagnet --get_phy_info
```

Fabric Health Validation Example

For NDR/HDR/EDR links, symbol errors (NDR/HDR) or effective errors (EDR) are the actual errors seen by the application level after error correction.

The below methodology is recommended as a first step if fabric performance is degraded.

1. Make sure the significant traffic is running in the fabric

2. ibdiagnet --pc --reset_phy_info -i <mlx_dev>

3. Wait for some time (5-10 minutes)

4. ibdiagnet --get_phy_info -i <mlx_dev>

5. Review ibdiagnet2.log

6. Contact Support if Symbol/Effective BER Check finished with errors.

For detailed description of cmd line parameters, see previous chapter “Bit Error Rate”

BER check log file fragment:

```
-E- Symbol BER Check finished with errors
-E- H-10/U1/P1 - BER exceeds threshold - BER type: Symbol BER,
FEC mode: STD-RS, BER value = 1.500000e+01 / threshold =
5.000000e-12
-E- H-14/U1/P1 - BER exceeds threshold - BER type: Symbol BER,
FEC mode: STD-LL-RS, BER value = 1.500000e+01 / threshold =
5.000000e-12
-E- H-3/U1/P1 - BER exceeds threshold - BER type: Symbol BER, FEC
mode: MLNX_RS_544_514_PLR, BER value = 1.500000e+01 / threshold =
5.000000e-12
-E- H-7/U1/P1 - BER exceeds threshold - BER type: Symbol BER, FEC
mode: MLNX_RS_271_257_PLR, BER value = 1.500000e+01 / threshold =
5.000000e-12
-E- SW-1-0/U1/P4 - BER exceeds threshold - BER type: Symbol BER,
FEC mode: RS_FEC_544_514, BER value = 1.500000e+01 / threshold =
5.000000e-12
-E- SW-1-0/U1/P5 - BER exceeds threshold - BER type: Symbol BER,
FEC mode: STD-LL-RS, BER value = 1.500000e+01 / threshold =
5.000000e-12

-----------------------------------------------
Fabric Summary
```

```
Total Nodes             : 24
IB Switches             : 8
IB Channel Adapters     : 16
IB Aggregation Nodes    : 0
IB Routers              : 0

Total number of links   : 32
Links at 4x10           : 32

High BER reported by 6 ports
```

BER check error section in db_csv file:

```
START_ERRORS_SYMBOL_BER_CHECK
Scope,NodeGUID,PortGUID,PortNumber,EventName,Summary
PORT,0x0002c90000000005,0x0002c90000000006,1,BER_EXCEEDS_THRESHOLD
exceeds threshold - BER type: Symbol BER, FEC mode: STD-RS, BER
value = 1.500000e+01 / threshold = 5.000000e-12 "
PORT,0x0002c90000000015,0x0002c90000000016,1,BER_EXCEEDS_THRESHOLD
exceeds threshold - BER type: Symbol BER, FEC mode: STD-LL-RS,
BER value = 1.500000e+01 / threshold = 5.000000e-12 "
PORT,0x0002c90000000025,0x0002c90000000026,1,BER_EXCEEDS_THRESHOLD
exceeds threshold - BER type: Symbol BER, FEC mode:
MLNX_RS_544_514_PLR, BER value = 1.500000e+01 / threshold =
5.000000e-12 "
PORT,0x0002c90000000035,0x0002c90000000036,1,BER_EXCEEDS_THRESHOLD
exceeds threshold - BER type: Symbol BER, FEC mode:
MLNX_RS_271_257_PLR, BER value = 1.500000e+01 / threshold =
5.000000e-12 "
PORT,0x0002c90000000049,0x0002c90000000049,4,BER_EXCEEDS_THRESHOLD
exceeds threshold - BER type: Symbol BER, FEC mode:
RS_FEC_544_514, BER value = 1.500000e+01 / threshold = 5.000000e-
12 "
```

```
PORT,0x0002c90000000049,0x0002c90000000049,5,BER_EXCEEDS_THRESHOLD
exceeds threshold - BER type: Symbol BER, FEC mode: STD-LL-RS,
BER value = 1.500000e+01 / threshold = 5.000000e-12 "
END_ERRORS_SYMBOL_BER_CHECK
```

Rail Optimized Topology Validation

This section specifies options for fast HCA to Top-of-Rack cabling validation in the rail optimized topologies like "DGX SuperPOD". The feature checks that all HCAs nodes connected to the same Top-of-Rack switch have the same PCIe address (BDF) in the corresponding server. The "Rail Optimized Topology" validation is applicable to the compute nodes. An output ibdiagnet2.rails file includes PCIE BDF details of the HCAs with regard to the corresponding Top of the Rack switch.

Parameter	Description
--rail_validation	Checks that a topology is rail optimized. Data will be dumped to the ibdiagnet2.rails file. Warnings and errors will be dumped to the ibdiagnet2.log file.
--rail_validation_opt	Comma separated Rail Optimized Validation options. • regex='regular expression' - only nodes matching the regular expression will be included in the report

Example:

```
ibdiagnet --rail_validation
```

- **ibdiagnet's Output:**

```
Rail Optimized Topology Validation
-W-     Node rail connectivity mismatch on the switch:
"SwitchIB Mellanox Technologies" GUID=0xe41d2d030003e470
-W-              rail A (PCIe 0000:04:00.0): 1 ports ==> r-
ufm118/U1/P1 <--> Se41d2d030003e470/Ne41d2d030003e470/P35
-W-              rail B (PCIe 0000:09:00.0): 1 ports ==> r-
ufm112/U2/P1 <--> Se41d2d030003e470/Ne41d2d030003e470/P30
-W-              rail C (PCIe 0000:10:00.0): 2 ports ==> r-
ufm218/U1/P1 <--> Se41d2d030003e470/Ne41d2d030003e470/P33,
...
-W- Rail Optimized Topology Validation ended with 1 warnings
-I- Rail Optimized Topology validation is usually applicable
to the compute nodes.
-I- If detected mis-cabled nodes are not compute ones, please
apply Rail Optimized Topology check for specific set of nodes
by invoking: --rail_validation_opt regex='reg expression'
```

- **ibdiagnet.rails file content:**

```
Node rail connectivity mismatch on the switch: SwitchIB
Mellanox Technologies GUID=0xe41d2d030003e470
    rail A(0000:04.00.00): 1 ports:
                           r-ufm118/U1/P1 <-->
Se41d2d030003e470/Ne41d2d030003e470/P35
    rail B(0000:09.00.00): 1 ports:
                           r-ufm112/U2/P1 <-->
Se41d2d030003e470/Ne41d2d030003e470/P30
    rail C(0000:10.00.00): 2 ports:
                           r-ufm218/U1/P1 <-->
Se41d2d030003e470/Ne41d2d030003e470/P33
                           r-ufm216/U2/P1 <-->
Se41d2d030003e470/Ne41d2d030003e470/P34
```

Example:

```
ibdiagnet --rail_validation --rail_validation_opt regex='[a-zA-Z]-ufm11[0-9]*'
```

Output:

- HCAs installed in r-ufm118 and r-ufm112 servers will be included in the report, as their node descriptions match provided the regular expression
- HCAs installed in r-ufm216 and r-ufm218 servers will be excluded from the report, as their node descriptions do not match provided regular expression

```
Rail Optimized Topology Validation
-W-     Node rail connectivity mismatch on the switch: "SwitchIB Mellanox Technologies" GUID=0xe41d2d030003e470
-W-              rail A (PCIe 0000:04:00.0): 1 ports ==> r-ufm118/U1/P1 <--> Se41d2d030003e470/Ne41d2d030003e470/P35
-W-              rail B (PCIe 0000:09:00.0): 1 ports ==> r-ufm112/U2/P1 <--> Se41d2d030003e470/Ne41d2d030003e470/P30
-W- Rail Optimized Topology Validation ended with 1 warnings
-I- Rail Optimized Topology validation is usually applicable to the compute nodes.
-I- If detected mis-cabled nodes are not compute ones, please apply Rail Optimized Topology check for specific set of nodes by invoking: --rail_validation_opt regex='reg expression'
```

Dragonfly and Topology Validation

This section specifies the options for DragonFly+ topology validation. Topology validation checks that all DFP+ islands are built as two level “fat-trees” and they are properly interconnected. It also reports on a network theoretical “bisectional” bandwidth of the topology.

A newly generated output file – ibdiagnet2.dfp will contain details about DFP islands, their root switches, connectivity tables, global links and theoretical bandwidth.

Parameter	Description
--dfp	Provides a report of the fabric Dragonfly+ analysis. Data will be dumped to the ibdiagnet2.dfp file. Warnings and errors will be dumped to the ibdiagnet2.log file.
--dfp_opt	Comma separated Dragonfly+ validation options. Possible values are: • max_cas=<num> - maximum number of CA on Dragonfly+ islands root switches (default=1). This parameter is mutual exclusive with –smdb parameter
--smdb <file>	Loads DFP+ islands roots from the subnet manager SMDB file (if routing engine reported in opensm-smdb.dump file is “dfp”).

Example:

```
ibdiagnet ––dfp
```

- **ibdiagnet's Output**:

```
DFP Topology Validation
-I- 5 DFP islands were detected
```

```
-I- DFP Symmetrical switch connectivity discovered, global
links per island: 16
-I- Partially resilient DFP discovered
-I- All DFP islands have the same bandwidth: 160 Gbps
-I- Theoretical DFP network bisection bandwidth: 107 Gbps
-W- DFP large topology was detected; may be implemented as
medium
-W- DFP validation finished with warnings
```

- **dfp file islands details**:

```
island: 0
 rank: 0 (size: 4)
        0x0002c90000000194 R0_G0_S1/U1
        0x0002c90000000198 R0_G0_S3/U1
        0x0002c900000001cc R1_G0_S0/U1
        0x0002c900000001d0 R1_G0_S2/U1

 rank: 1 (size: 4)
        0x0002c90000000004 G0_L0/U1
        0x0002c90000000018 G0_L1/U1
        0x0002c9000000002c G0_L2/U1
        0x0002c90000000040 G0_L3/U1
```

- **dfp file islands connectivity table**:

```
island 0, bandwidth 160 [Gb/s], global links 16, resilient
connection to all: No
        switch 0x0002c90000000194, global links: 4, resilient
connection to all: No, connected islands: 2, free ports: 73
                island: 2, global links: 3, resilient: Yes
                island: 3, global links: 1, resilient: No
```

```
        switch 0x0002c90000000198, global links: 4, resilient
connection to all: No, connected islands: 4, free ports: 73
                island: 2, global links: 1, resilient: No
                island: 1, global links: 1, resilient: No
                island: 3, global links: 1, resilient: No
                island: 4, global links: 1, resilient: No
        switch 0x0002c900000001d0, global links: 4, resilient
connection to all: No, connected islands: 3, free ports: 73
                island: 1, global links: 1, resilient: No
                island: 3, global links: 2, resilient: No
                island: 4, global links: 1, resilient: No
        switch 0x0002c900000001cc, global links: 4, resilient
connection to all: No, connected islands: 2, free ports: 73
                island: 1, global links: 2, resilient: No
                island: 4, global links: 2, resilient: No
```

Fat-Tree Topology Validation

This section specifies the options for Fat-Tree topology validation. Topology validation checks that the provided topology is a properly connected Fat-Tree topology. It detects tree structure, its “connectivity groups” and neighborhoods and their link issues. It also reports on a network theoretical “bisectional” bandwidth.

A newly generated output file – ibdiagnet2.fat_tree contains details about switches uplinks/downlinks issues and tree structure by levels. In a case FLIDs were applied on the network the output could contain warning if the same FLID appears in some neighborhoods.

Parameter	Description
--ft	Provides a report of the fabric Fat-Tree analysis. Data will be dumped to the ibdiagnet2.fat_tree file. Warnings and errors will be dumped to the ibdiagnet2.log and ibdiagnet2.fat_tree files.

Parameter	Description
--ft_roots_regex_opt	The regular expression to select Fat-Tree root nodes. Only nodes matching the regular expression will be taken as roots.
--smdb <file>	Loads Fat-Tree roots from the subnet manager SMDB file (the routing engine reported in opensm-smdb.dump file should be one form the following list: “Fat-Tree”, Adaptive Routing Fat-Tree”, “UPDN”, “Adaptive Routing UPDN”).

Example:

```
ibdiagnet —-ft
```

- **ibdiagnet's Output**:

```
 Fat-Tree Topology Validation
-I- Fat-Tree topology detection finished successfully

-I- 2 level Fat-Tree was discovered:
         rank: 0(Roots) #switches: 4
         rank: 1 #switches: 4

-I- Fat-Tree topology validation finished successfully
-E- For more errors see the dump file: ibdiagnet2.fat_tree
-I- Calculated Fat-Tree bisectional bandwidth: 800.00 [Gb/s]
```

- **fat_tree file “connectivity groups”/neighborhoods and their switches uplinks/downlinks issues details**:

```
-I- Neighborhood 0: all spines have the same number of
downlinks: 4
-I- Neighborhood 0: all lines have the same number of
uplinks: 4
-W- Neighborhood 0: lines with different number of downlinks
(expected  4 downlinks)
         2 downlinks: BM1-2 BM1-1
```

- **fat_tree file tree structure - switches by rank**:

```
rank: 0 (Roots)size: 4
        BM0-2
        BM0-0
        BM0-1
        BM0-3

rank: 1 (Leaves)size: 4
        BM1-0
        BM1-2
        BM1-1
        BM1-3
```

- **fat_tree file tree structure – “connectivity groups”/neighborhoods by rank**

```
on ranks (0, 1) -- neighborhoods: 1
        neighborhood: 0
                spines: 4 switches
                        BM0-2
                        BM0-0
                        BM0-1
                        BM0-3
```

```
                lines: 4 switches
                        BM1-0
                        BM1-2
                        BM1-1
                        BM1-3
```

- **fat_tree file - the same FLID is found in some neighborhoods**

```
-W- FLID=155 appears in the following neighborhoods: 1, 3
```

IBDIAGNET Output Control

This section specifies ibdiagnet options to control location of ibdiagnet output files.

Parameter	Description
-o | -output_path <directory>	Specifies the directory where the output all files will be located. (default="/var/tmp/ibdiagnet2/").

Write and Read Capability Files (Deprecated)

This section is relevant only for devices not supporting GeneralInfoSMP and GeneralInfoGMP MADs.

Parameter	Description
--write_capability <file>	Writes out an example file for capability masks configuration, also writes the

Parameter	Description
	default capability masks for some devices.
--read_capability <file>	Specifies the capability of the mask configuration file for the fabric. ibdiagnet uses this mapping for Vendor Specific MADs sending.

Example:

```
# This capability masks configuration file was automatically
generated by ibdiagnet

# Starting of SMP Vendor Specific Attributes section
SMP

# Capability Mask per vendor id and device id
#-----------------------------------------
# Here come devices that don't support GeneralInfo MAD.
# The Capability Mask is presented in IPv6 format
# NOTE: It is illegal to define a rule for a DevID with no FW and
then define one with FW.
# Example: VenID: 0x2c9 DevID: 0xc738 mask: ::3
#          meaning devices of vendor 0x2c9 and device 0xc738
#          the supported SMP vendor specific attributes are
#          those of capability with mask bit 1 and 2
VenID: 0x2c9 DevID: 0x66 mask: ::
VenID: 0x2c9 DevID: 0x191 mask: ::c0
VenID: 0x2c9 DevID: 0x1b4 mask: ::c0
VenID: 0x2c9 DevID: 0x3f8 mask: ::
…
```

Useful Options

Limit Error and Warning Printing to the Screen

Parameter	Description
--screen_num_errs <num>	Specifies the maximal number error/warning messages logged to the screen (default=5). If the number of errors/warnings is higher than a <num> value. Additional error/warning messages will be logged to the ibdiagne2.log file.

Example:

```
ibdiagnet
```

Output (default):

```
Nodes Information
-I- Devid: 4099(0x1003), PSID: MT_1090120019, Latest FW
Version:2.42.5000
-I- Devid: 4103(0x1007), PSID: MT_1090111019, Latest FW
Version:2.42.5000
-I- Devid: 4115(0x1013), PSID: MT_2190110032, Latest FW
Version:12.100.5600
-I- Devid: 4119(0x1017), PSID: MT_0000000008, Latest FW
Version:16.18.160
-I- Devid: 51000(0xc738), PSID: MT_1270110020, Latest FW
Version:9.3.1700
-I- Devid: 52000(0xcb20), PSID: MT_1880110032, Latest FW
Version:11.1100.26
-E- FW Check finished with errors
-W- r-ufm118/U1 - Node with Devid:4115(0x1013),PSID:MT_2190110032
has FW version 12.27.6008 while the latest FW version for the
```

```
same Devid/PSID on this fabric is 12.100.5600
-W- r-ufm112/U2 - Node with Devid:4115(0x1013),PSID:MT_2190110032
has FW version 12.26.4012 while the latest FW version for the
same Devid/PSID on this fabric is 12.100.5600
-W- r-ufm218/U1 - Node with Devid:4115(0x1013),PSID:MT_2190110032
has FW version 12.26.4000 while the latest FW version for the
same Devid/PSID on this fabric is 12.100.5600
-W- r-ufm216/U2 - Node with Devid:4115(0x1013),PSID:MT_2190110032
has FW version 12.26.4000 while the latest FW version for the
same Devid/PSID on this fabric is 12.100.5600
-E- r-ufm101/U2 - The firmware of this device returned invalid
general info data
```

Example:

```
ibdiagnet --screen_num_errs 3
```

Output ((--screen_num_errs 3):

```
Nodes Information
-I- Devid: 4099(0x1003), PSID: MT_1090120019, Latest FW
Version:2.42.5000
-I- Devid: 4103(0x1007), PSID: MT_1090111019, Latest FW
Version:2.42.5000
-I- Devid: 4115(0x1013), PSID: MT_2190110032, Latest FW
Version:12.100.5600
-I- Devid: 4119(0x1017), PSID: MT_0000000008, Latest FW
Version:16.18.160
-I- Devid: 51000(0xc738), PSID: MT_1270110020, Latest FW
Version:9.3.1700
-I- Devid: 52000(0xcb20), PSID: MT_1880110032, Latest FW
Version:11.1100.26
-E- FW Check finished with errors
```

```
-I- Errors/Warnings list will be reported in log file
```

Node Name Map

The following ibdiagnet option can be used to provide meaningful names for unmanaged switches in ibdiagnet log and dump files. Same file can be used in opensm and infiniband-diags utilities such as ibnetdiscover.

Parameter	Description
-m|--map <map-file>	Specifies the mapping file that maps unmanaged switch node GUID to the name. The format of the content of file should be as follows: `0x[0-9a-fA-F]+ "name" e.g 0x123456 "Switch 1"` The file path can be specified via environment variable `"IBUTILS_NODE_NAME_MAP_FILE_PATH"` .

Scope

The following ibdiagnet options allow counters and diagnostics fetching only from subset of nodes/switches in the fabric.

Parameter	Description
--scope <file>	The file with a list of Node-GUIDs and ports belonging to the scope.
--exclude_scope <file>	The file with a list of Node-GUIDs and ports which the counters fetching, and diagnostics should not be applied to.

> **Note**
>
> The ibdiagnet2.ibnetdiscover file will not be generated if any of the options is provided.

File format:

Scope file format includes the version and the list of nodes to include in the scope, according to the following syntax:

- version:<format version number> - Scope file format version, must be first line of the file. Supported version 1.0
- Comment lines start with #.
- Nodes line of the following formats:
  - <Node GUID> - Includes node with specified node GUID with all its ports.
  - <Node GUID>@port1/port2/... - Includes only the specified ports of specified node.

    Note: When using exclude scope option, only the specified ports of the node will be excluded.
  - ALL_SWITCHES - Includes all switches with all ports in the scope.
  - ALL_CAS - Includes all HCAs in the scope.
  - ALL_ROUTERS - Includes all routers in the scope.

Examples:

- Defining a scope for nodes with Node GUIDs 0x10001, 0x10002, 0x10003 with all their ports:

```
version: 1.0
0x10001
0x10002
```

```
0x10003
```

- Defining a scope for ports 1,2,17 of node with Node GUID 0x10002:

```
version: 1.0
0x10002@1/2/17
```

- Defining a scope for all switches (with all their ports):

```
version: 1.0
ALL_SWITCHES
```

- Defining a scope for all CAs:

```
version: 1.0
ALL_CAS
```

- Define scope with all the following nodes:
  - node with Node GUIDs 0x10001
  - port 1,2,17 of node with Node GUID 0x10002
  - All CAs

```
version: 1.0
0x10001
0x10002@1/2/17
ALL_CAS
```

Note

Scope feature is not applicable for routing validation stages!

Skipping Validation Stages

Some data collection/diagnostic can be skipped in order to speed up ibdiagnet reporting. For instance, when only routing validation is required, no need to perform port counters fetching and checks.

Parameter	Description
--skip <stage>	Skips the executions of particular diagnostic stages.

The following stages can be skipped:

Parameter	Description
dup_guids	Duplicated GUIDs check
dup_node_desc	Duplicated node description check
lids	Valid LID assignment check
sm	Subnet Manager checks
nodes_info	Fetching vendor specific data from nodes
pkey	Partitions fetch and validation
links	Fetching links data
pm	Fetching and checking port counters
speed_width_check	Link speed and link width checks
temp_sensing	Fetching temperature sense
virt	Virtualization stage
hca_cfg_check	Post Reports SM Validations
all	Skip all above stages

If the Virtualization stage is skipped, the ibdiagnet2.ibnetdiscover file will not contain virtual ports information.

Example:

```
ibdiagnet -r --r_opt=vs,sl=2 --skip pm, pkey, links,
temp_sensing,speed_width_check,nodes_info,sm,dup_guids,dup_node_des
```

Running ibdiagnet in 3D-Torus Topology

Parameter	Description
--vlr <file>	This option provides opensm-path-records.dump file that includes source-to-destination to SL mapping. This file is generated by dump_pr Subnet manager plugin. ibdiagnet will use this mapping for MADs sending on correct SL.

Dumping the Old Version of PORTS Section in the CSV File

Parameter	Description
--back_compat_db <ver>	Indicates the old version of PORTS section in CSV file for backward compatibility. If the given version is less than 2.0 (also not 0), the following fields will not be dumped in the CSV: • `CapMsk2,FECActv,RetransActv`

Debug Mode

Debug diagnostics can be added into the debug log file (ibdiagnet2.debug) generated by the ibdiagnet.

Parameter	Description
--dbg_modules	Comma separated module's names to be added to the debug log file. Possible values are: • IBIS • IBDIAG, • IBDM, • IBDIAGNET, • ALL
--dbg_level	Verbosity levels to be applied on the debug log file. Possible values are: • 0x01 - Error • 0x02 - Info • 0x04 - MAD • 0x08 - Discover • 0x10 - Debug • 0x20 - Funcs • 0x80 - Sys • 0xff - ALL

Examples:

```
ibdiagnet --dbg_modules IBIS,IBDIAG
```

The following command creates debug logs for IBIS and IBDIAG modules with Discover and Debug verbosity levels:

```
ibdiagnet --dbg_modules IBIS,IBDIAG --dbg_level 0x08,0x10
```

Output (ibdiagnet2.debug):

```
[Nov 18 10:47:11 201508][6536][0x02 0x10] ->
(ibdiag_discover.cpp,1780,BuildPortInfoExtendedDB): The node r-
ufm98/U2 doesn't support PortInfoExtended MAD
[Nov 18 10:47:11 201515][6536][0x02 0x10] ->
(ibdiag_discover.cpp,1745,BuildPortInfoExtendedDB): node: sw-
osm81/U1 port_info_cap_mask: 0x4250c848 port_info_cap_mask2: 0x0
[Nov 18 10:47:11 201520][6536][0x02 0x10] ->
(ibdiag_discover.cpp,1780,BuildPortInfoExtendedDB): The node sw-
osm81/U1 doesn't support PortInfoExtended MAD
[Nov 18 10:47:11 201575][6536][0x01 0x10] -> (ibis.cpp, 616,
DoAsyncSend): Send MAD with TID=2952790016
[Nov 18 10:47:11 201591][6536][0x01 0x10] -> (ibis_mads.cpp, 325,
AsyncSendAndRec): Send MAD with data_ptr:0x1835c68
[Nov 18 10:47:11 201643][6536][0x01 0x10] -> (ibis.cpp, 616,
DoAsyncSend): Send MAD with TID=2969567232
[Nov 18 10:47:11 201656][6536][0x01 0x10] -> (ibis_mads.cpp, 325,
AsyncSendAndRec): Send MAD with data_ptr:0x182e2c8
[Nov 18 10:47:11 201667][6536][0x01 0x10] -> (ibis.cpp, 632,
DoAsyncRec): Receive MAD with TID=2952790016
[Nov 18 10:47:11 201672][6536][0x01 0x10] -> (ibis.cpp, 635,
DoAsyncRec): UMAD Status=0x0000
```

Validation of SM configuration for HCAs

Ibdiagnet is checking that all HCAs have the same SM configurations for some features.

Features are:

- OOOSLMask (ar_sl_mask) (by default)

- adaptive_timeout_sl_mask (by default)
- virt_enabled (by default, avoid --skip virt)
- sl2vl (-r)
- VL Arbitration (-r)
- CC: (--congestion_control)
    - CongestionHCAGeneralSettings
    - CongestionHCARPParameters
    - CongestionHCANPParameters
    - CongestionHCAAlgoConfig
    - CongestionHCAConfigParams

> **Info**
>
> This validation will be start automatically by data that received by ibdiagnet
>
> This validation can be skipped with '--skip hca_cfg_check'

```
Command line to get all parameters

   ibdiagnet -r --congestion_control
```

Every field has a separated warning and there is also a new line in the 'Fabric Summary'

Examples:

ibdiagnet2.log:

```
-W- Post Reports SM Configuration Validations finished with
warnings
-W- Field 'OOOSLMask' has 2 different values across the fabric
[0,1]
-W- Field 'AdaptiveTimeoutSLMask' has 2 different values across
the fabric [2,3]
-W- Field 'SL2VL_0' has 3 different values across the fabric
[2,3,4]
-W- Field 'SL2VL_1' has 2 different values across the fabric
[2,3]
-W- Field 'SL2VL_2' has 4 different values across the fabric
[2,3,4,5]
-I- All other warnings can be found in ibdiagnet2.db_csv
```

ibdiagnet2.db_csv:

```
Scope,NodeGUID,PortGUID,PortNumber,EventName,Summary
CLUSTER,0x00,0x00,0x00,DIFFERENT_VALUE_BY_SM_CONFIGURATION,"Field
'OOOSLMask' has 2 different values across the fabric [0,1]"
CLUSTER,0x00,0x00,0x00,DIFFERENT_VALUE_BY_SM_CONFIGURATION,"Field
'AdaptiveTimeoutSLMask' has 2 different values across the fabric
[2,3]"
CLUSTER,0x00,0x00,0x00,DIFFERENT_VALUE_BY_SM_CONFIGURATION,"Field
'SL2VL_0' has 3 different values across the fabric [2,3,4]"
CLUSTER,0x00,0x00,0x00,DIFFERENT_VALUE_BY_SM_CONFIGURATION,"Field
'SL2VL_1' has 2 different values across the fabric [2,3]"
CLUSTER,0x00,0x00,0x00,DIFFERENT_VALUE_BY_SM_CONFIGURATION,"Field
'SL2VL_2' has 4 different values across the fabric [2,3,4,5]"
...
```

Fabric Summary:

```
Post Reports SM Configuration Validations: 82 fields have
different value across the fabric.
```

Post Reports SM Configuration Validations: 82 fields have different value across the fabric.

Security Keys

Security Keys

When any security keys should be used with ibdiagnet, it can use option security keys to provide path to all security files that should be used:

Parameter	Description
--security_keys <path to key file>	Specifies the path to the directory with the key files. (guid2lid, guid2mkey, neighbors, guid2vskey, guid2cckey, guid2_m2n_key, guid2pmkey).

File names

Type	File names
Management Key (MKEY)	guid2lid guid2mkey neighbors guid2vskey
VendorSpecific Key (VS Key)	guid2vskey
CongestionControl Key (CC Key)	guid2cckey

Type	File names
Manager2Node Key (M2N Key)	guid2_m2n_key
Performance Management Key (VS Key)	guid2pmkey

Management Key (MKEY)

When MKEY protection is enabled by the Subnet Manager on the fabric devices, the following command options should be used in ibdiagnet:

Parameter	Description
--m_key <m_key>	Specifies constant MKey for the fabric. The MKey value should be specified when a single MKEY is shared by all InfiniBand devices. The mkey value can be obtained from the opensm.conf file (m_key parameter).

Example of guid2mkey/guid2cckey/guid2vskey file:

```
0x0002c9000000001e 0x0000000000000111
0x0002c9000000002a 0x0000000000000222
0x0002c90000000026 0x0000000000000333
0x0002c90000000006 0x0000000000000444
```

Example of guid2lid file:

```
0x0002c9000000004b 0x0027 0x0027
0x0002c9000000002a 0x001a 0x001b
0x0002c90000000006 0x0004 0x0005
0x0002c90000000047 0x000e 0x000e
```

Example of neighbours file:

```
0x0002c9000000004d:4 0x0002c9000000000e:1
0x0002c9000000004b:1 0x0002c9000000002e:1
0x0002c90000000049:2 0x0002c90000000022:1
0x0002c90000000006:1 0x0002c9000000004d:2
```

Example:

```
ibdiagnet --mkey 0x00ff
ibdiagnet --security_keys /var/cache/opensm/
ibdiagnet --security_keys /tmp/opensm_mkey_files/
```

VendorSpecific Key (VS Key)

When VendorSpecific keys are generated and provisioned by the Subnet Manager, VendorSpecific keys should be sent in VendorSpecific MADs per endport.

Parameter	Description
--vs_key <vs_key>	Specifies constant vs_key. VendorSpecific Key parameter value format: 0x<hex_value>

Example of guid2vskey FILE:

```
0x0000000000000111 0x0000000000000001
0x0000000000000222 0x2
0x0000000000000333 0x0000000003
0x0000000000000444 0x0000000000000004
```

Example:

```
ibdiagnet --vs_key 0x0000000000000123
ibdiagnet --vs_key 0x123
```

CongestionControl Key (CC Key)

When CongestionControl keys are generated and provisioned by the Subnet Manager, CongestionControl keys should be sent in CongestionControl (CC) MADs perend port.

Parameter	Description
--cc_key <cc_key>	Specifies constant cc_key. CC Key parameter value format: 0x<hex_value>

Example of guid2cckey FILE:

```
0x0000000000000111 0x0000000000000001
0x0000000000000222 0x2
0x0000000000000333 0x0000000003
0x0000000000000444 0x0000000000000004
```

Example:

```
ibdiagnet --cc_key 0x0000000000000123
ibdiagnet --cc_key 0x123
```

Manager2Node Key (M2N Key)

When Manager2Node keys are generated and provisioned by the Subnet Manager, Manager2Node keys should be sent in M2N (pFRN) MADs per node.

Parameter	Description
--m2n_key <m2n_key>	Specifies constant m2n_key for the fabric. M2N Key parameter value format: 0x<hex_value>

Example of guid2_m2n_key FILE:

```
0x0000000000000111 0x0000000000000001
0x0000000000000222 0x2
0x0000000000000333 0x0000000003
0x0000000000000444 0x0000000000000004
```

Examples:

```
ibdiagnet --m2n_key 0x0000000000000123
ibdiagnet --m2n_key 0x123
```

Performance Management Key (VS Key)

When PerformanceManagement keys are generated and provisioned by the Subnet Manager, PerformanceManagement keys should be sent in PerformanceManagement MADs per endport.

Parameter	Description
--pm_key <vs_key>	Specifies constant pm_key. PerformanceManagement Key parameter value format: 0x<hex_value>
--pm_key_file <path_to_pm_key_file>	Specifies the path to pm_key_file: guid2pm_key. The guid2pm_key file includes pair of values GUID and key in the following format: 0x<guid_hex_value> 0x<pm_key_hex_value> (Deprecated, see --security_keys).

Example of guid2pmkey FILE:

```
0x0000000000000111 0x0000000000000001
0x0000000000000222 0x2
0x0000000000000333 0x0000000003
0x0000000000000444 0x0000000000000004
```

Example:

```
ibdiagnet --vs_key 0x0000000000000123
ibdiagnet --vs_key 0x123
```

Aggregation Management Key (AMKey)

Aggregation Management Key (AM key) is sent in SHARP Management MADs to the Aggregation nodes. (default 0). Aggregation management key is configured in SHARP Aggregation Manager configuration file.

Parameter	Description
--am_key <am_key>	Specifies constant am_key. AM Key parameter value format: `0x<hex_value>`
--am_key_file <path_to_am_key_file>	Specifies the path to am_key_file: guid2am_key.

Parameter	Description

	The guid2am_key file includes pair of values GUID and key in the following format:
	`0x<guid_hex_value>`
	`0x<am_key_hex_value>`

Example of guid2amkey file:

```
0x0000000000000111 0x0000000000000001
0x0000000000000222 0x2
0x0000000000000333 0x0000000003
0x0000000000000444 0x0000000000000004
```

Example:

```
ibdiagnet --am_key 0x0000000000000123
ibdiagnet --am_key 0x123
ibdiagnet --am_key_file /tmp/guid2am_key
ibdiagnet --am_key_file /tmp/am_keys
```

Plugins

The chapter contains the following sections:

Cable Diagnostic

> **Note**
>
> Please be aware that cable is deprecated and will be entirely removed in the next version.

> **Note**
>
> Cable plugin and its parameters are now obsolete and not supported in the CMIS cables, please use `--get_phy_info` instead.

Cable plugin collects various information from the cables attached to the fabric ports.

Parameter	Description
--get_cable_info	Gets cable info from the fabric ports and dump cable info to ibdiagnet2.db_csv and ibdiagnet2.cables. Not supported in CMIS cables, --get_phy_info should be used instead.
--cable_info_disconnected	Gets cable info on disconnected ports (the cable is attached only to the switch port). This option is applicable with the "get-cable-info" flag. Data will be dumped to ibdiagnet2.db_csv and ibdiagnet2.cables. Not supported in CMIS cables, --get_phy_info and --phy_cable_disconnected should be used instead.

Example:

```
ibdiagnet --get-cable-info --cable_info_disconnected
```

The data is dumped to the ibdiagnet2.cables file in the following format:

```
-------------------------------------------------------
Port=1 Lid=0x00a4 GUID=0xf45214030046a0a1 Port Name=coral-ufm-
001/U1/P1
-------------------------------------------------------
Vendor: Mellanox
OUI: 0x2c9
PN: MCP1600-E002
SN: MT1739VS02126
Rev: A3
```

```
Length: 2 m
Type: Copper cable- unequalized
SupportedSpeed: SDR/DDR/QDR/FDR/EDR
Temperature: N/A
PowerClass: 1
NominalBitrate: 0 Gb/s
CDREnableTxRx: N/A N/A
InputEq: N/A
OutputAmp: N/A
OutputEmp: N/A
FW Version: N/A
Attenuation(5,7,12): 7 8 13
RX power type: OMA
RX1 Power: 0.000 mW, -999.999 dBm
RX2 Power: 0.000 mW, -999.999 dBm
RX3 Power: 0.000 mW, -999.999 dBm
RX4 Power: 0.000 mW, -999.999 dBm
TX1 Bias: 0.000 mA
TX2 Bias: 0.000 mA
TX3 Bias: 0.000 mA
TX4 Bias: 0.000 mA
TX1 Power: 0.000 mW, -999.999 dBm
TX2 Power: 0.000 mW, -999.999 dBm
TX3 Power: 0.000 mW, -999.999 dBm
TX4 Power: 0.000 mW, -999.999 dBm
```

PHY Diagnostics

PHY diagnostics plugin collects NVIDIA® Mellanox® Vendor-Specific Data from NVIDIA devices including BER data, low level PHY counters, sensors, fans, PSUs, etc.

Collected information is dumped into ibdiagnet2.db_.csv file under following sections:

- PHY_DBs

- FANS_SPEED
- FANS_THRESHOLDS
- FANS_ALERT
- FAN_SERIAL_NUMBERS
- TEMPERATURE_SENSORS
- TEMPERATURE_ALERTS
- POWER_SENSORS
- PSU information
    - NodeGuid, PSUIndex,IsPresent, IsFRU, ACInput, DCState, AlertState, FanState, TemperatureState, SerialNumber
- SYSTEM_GENERAL_INFORMATION
    - NodeGuid, SerialNumber, PartNumber, Revision, ProductName
- PHY_PORT_GRADE

Parameter	Description
--get_phy_info	Queries all ports for PHY information including BER counters. Data will be dumped to the ibdiagnet2.db_csv and ibdiagnet2.net_dump_ext files.
--reset_phy_info	Queries and clears all ports PHY information. **Note:** It is recommended to use this option with `-pc`, as both options have cross counters and using only one of them can be confusing on the next iteration of counters or registers collection.

Parameter	Description
--get_ppamp	Queries all ports for PPAMP (Port Phy opAMP data), works with `--get_phy_info`. Data will be dumped to the ibdiagnet2.db_csv file.
--show_cap_reg	Indicates to dump also capability registers, works with `--get_phy_info`. Data will be dumped to the ibdiagnet2.db_csv file.

Example:

```
ibdiagnet --get_phy_info
ibdiagnet --reset_phy_info
ibdiagnet --get_phy_info -get_ppamp
ibdiagnet --get_phy_info --show_cap_reg
```

PCI Diagnostics

Retrieving PCIE diagnostic plugin retrieves the data about servers' PCIE topologies.

Parameter	Description
--get_p_info	Queries HCA ports for PCI information. Data will be dumped to the ibdiagnet2.db_csv file.
--reset_p_info	Queries and clears HCA ports PCI information.

Example:

```
ibdiagnet --get_p_info
```

```
ibdiagnet --reset_p_info
```

Skip Loading Plugin (Debugging Purposes Only)

Parameter	Description
--skip_plugin	Skips the loads of the given library name. Applicable skip plugins: • libibdiagnet_cable_diag_plugin-2.1.1 • libibdiagnet_phy_diag_plugin-2.1.1 • or any other plugin installed in the system

Loading the Plugin from a Specific Location

Plugin loading path can be specified using the "IBDIAGNET_PLUGINS_PATH" environmental variable.

Example:

```
IBDIAGNET_PLUGINS_PATH=/tmp/plugins ibdiagnet --get_phy_info
```

Output:

```
Detected different plugin libraries with the same name, using the
first one:
```

```
        1. /tmp/plugins/libibdiagnet_phy_diag_plugin-2.1.1.so
        2.
/usr/share/ibdiagnet2.1.1/plugins//libibdiagnet_phy_diag_plugin-
2.1.1.so
Detected different plugin libraries with the same name, using the
first one:
        1. /tmp/plugins/libibdiagnet_cable_diag_plugin-2.1.1.so
        2.
/usr/share/ibdiagnet2.1.1/plugins//libibdiagnet_cable_diag_plugin-
2.1.1.so

Running: ibdiagnet --get_phy_info
----------
Load Plugins from:
/tmp/plugins
/usr/share/ibdiagnet2.1.1/plugins/
(You can specify more paths to be looked in with
"IBDIAGNET_PLUGINS_PATH" env variable)

Plugin Name                                   Result     Comment
libibdiagnet_cable_diag_plugin-2.1.1          Succeeded  Plugin
loaded
libibdiagnet_phy_diag_plugin-2.1.1            Succeeded  Plugin
loaded

-----------------------------------------------
```

ibdiagnet Dump Files

The following dump files are generated by ibdiagnet (depending on ibdiagnet command line parameters or configuration file settings).

Filename	Description
ibdiagnet2.log	Log file
ibdiagnet2.aports	APorts dump file
ibdiagnet2.lst	Fabric links in LST format
ibdiagnet2.net_dump	Fabric link dump including split cable mapping and FEC info
ibdiagnet2.net_dump_agg	Fabric link dump including split cable mapping and FEC info (aggregated by APort)
ibdiagnet2.net_dump_ext	Extended fabric link dump with FEC, BER and additional phy data
ibdiagnet2.sm	Subnet Managers
ibdiagnet2.pm	IB spec compliant Ports Counters
ibdiagnet2.pm_agg	IB spec compliant Ports Counters (aggregated by APort)
ibdiagnet2.mlnx_cntrs	Mellanox Diagnostic counters
ibdiagnet2.fdbs	Unicast FDBs **Note:** Dump disabled by default (*).
ibdiagnet2.mcfdbs	Multicast FDBs
ibdiagnet2.ar	Adaptive routing tables **Note:** Dump disabled by default (*).

Filename	Description
ibdiagnet2.far	Adaptive routing tables including SHIELD settings
ibdiagnet2.far_flid	Adaptive routing tables including only non-local FLIDs
ibdiagnet2.rn	SHIELD configuration tables
ibdiagnet2.rnc	SHIELD counters (Old file, disable by default)
ibdiagnet2.rnc2	SHIELD, SHIELDv2, HBF counters
ibdiagnet2.nodes_info	Nodes Information (FW version, etc)
ibdiagnet2.db_csv	ibdiagnet internal database
ibdiagnet2.pkey	Pkey tables
ibdiagnet2.ppcc	Port Programmable Congestion Control file
ibdiagnet2.vports	Virtualization:
ibdiagnet2.vport_pkeys	virtualization pkey tables
ibdiagnet2.aguid	alias GUIDs (ConnectX-3 only)
ibdiagnet2.slvl	SLVL tables of the fabric switches
ibdiagnet2.cables	Cable info
Ibdiagnet2.flid	FLIDs configuration details
ibdiagnet2.rails	"rails optimized" validation tests details
ibdiagnet2.sharp ibdiagnet.sharp_pm	SHARP data
ibdiagnet2.ibnetdiscover	Discovered network in "ibnetdiscover" format
ibdiagnet2.iblinkinfo	Discovered network in "iblinkinfo" format
ibdiagnet2.guid	A "scope file" created by Scope Builder

*If dump is disabled by default, use `'–enable_output'` to enable dump.

Warnings and Errors Printing Policy

Limited number of warnings/errors are printed to screen and log file. All warnings/errors are printed in the CSV file.

CRT	LOG	DB_CSV
Up to 5 errors/warnings	• Up to 5 warnings • All errors. **Note:** Exclusion for few kinds of errors that have limitation up to 5.	All error & warnings

Flexible Output Control

The chapter contains the following sections:

Special Keywords

The following are special predefined keywords used in ibdiagnet:

Parameter	Description
default	Used for setting the default value of a specific property for all elements (such as disabled/enabled output or specific path)
all	When used, all previously defined or subsequently values will be ignored and use the value that was set with this keyword.

CSV Section Prefix

To use the output control of CSV sections from the ibdiagnet2.db_csv file, a 'csv:' prefix is required.

Example:

```
csv:nodes - section 'NODES' in ibdiagnet2.db_csv
```

Flexible Output Options

This section specifies options for output files control. Output sections or files can be removed from output disabled/enabled or redirected to the separated files.

Parameter	Description
--disable_output <list of file types>	Disables the output to the specific dump files or removes specific CSV section in ibdiagnet2.db_csv file.
--enable_output <list of file types>	Enables the output to the specific dump files or to the specific CSV sections in ibdiagnet2.db_csv file.
--path <list pairs type=path>	Sets the custom path for specific file per type. Can be used with `-o | --output_path`.

1. **Example:**

   ```
   ibdiagnet --disable_output vports,vports_pkey,pkey
   ```

   Result:

   VPorts, VPorts Pkey and Partition keys files will not be generated.

2. **Example:**

   ```
   ibdiagnet  --disable_output csv:nodes
   ```

Result:

All the expected files will be dumped, section 'NODES' will not be created in ibdiagnet2.db_csv.

3. **Example:**

```
ibdiagnet  --disable_output default --enable_output
db_csv,lst,net_dump,pm
```

Output:

```
-I- Database                        :
/var/ibdiagnet2/ibdiagnet2.db_csv
-I- LST                             :
/var/ibdiagnet2/ibdiagnet2.lst
-I- Network dump                    :
/var/ibdiagnet2/ibdiagnet2.net_dump
-I- Ports Counters                  :
/var/ibdiagnet2/ibdiagnet2.pm
```

4. **Example:**

```
ibdiagnet --disable_output default --enable_output csv:NODES
```

Result:

Only 'NODES' & 'PM_INFO' section will be dumped to the ibdiagnet2.db_csv file.

5. **Example:**

```
--path default=/tmp/,db_csv=/tmp/db/,vports_pkey=/tmp/vkeys/
```

Output:

```
-I- You can find detailed errors/warnings in:
/tmp/ibdiagnet2.log


-I- Database                        : /tmp/db/ibdiagnet2.db_csv
-I- VPorts Pkey                     :
/tmp/vkeys/ibdiagnet2.vports_pkey
```

Exclusions/Exceptions

ibdiagnet can generate more of one `ibnl` file. In this case, the provided value in the `–path` parameter will be translated to the full path only, without filename or file type. The name of file cannot be controlled and all output files will be named by the default rule.

Example:

```
ibdiagnet -w /tmp/my.topo --path ibnl=/tmp/my_ibnl
```

Result:

```
/tmp/my_ibnl/SYSe41d2d030003e470.ibnl
/tmp/my_ibnl/SYSe91d2fe3cf03e321.ibnl
```

Document Revision History

Date	Revision	Section	Description
January 31, 2025	2.21.0	Basic Commands	Updated section
		Security Keys	Updated section
		Cable Diagnostic (Plugin)	Added a note that plugin is deprecated and will be entirely removed in the next version.
		Cable Diagnostic	Added a note that plugin is deprecated and will be entirely removed in the next version.
November 11, 2024	2.19.0	Basic Commands	Added the " --security_keys <path to key file>" command
			Updated section
		ibdiagnet Dump Files	Updated section
		Useful Options	Added the "hca_cfg_check" parameter
		Validation of SM configuration for HCAs	New section
		Security Keys	New section
May 11, 2023	2.13.0	Basic Commands	Added the "--fast_recovery" command
			Updated the "--sharp_opt <[csc][dsc][dscp][ad_hoc]>" command
		Fast Recovery	New section
		NVIDIA Scalable Hierarchical Aggregation and Reduction Protocol (SHARP) Support	Updated the "--sharp_opt <[csc][dsc][dscp][ad_hoc]>" command
		ibdiagnet Dump Files	Added "ibdiagnet2.iblinkinfo" filename
January 16, 2023	2.11.0	Basic Commands	Added the "--ppcc <filename|path|pattern>" command

Date	Revision	Section	Description
			Updated the "-P | --counter <cntr=threshold,... | all=threshold>" command
		Port Counters	Updated the "-P | -counter <<PM>=<value>>" command
		Routing Validation	Updated the "-r | -routing" command
		Routers	Updated the section.
		Congestion Control	Added the "--ppcc <filename|path|pattern>" command and "Output Port Programmable Congestion Control" sample.
		ibdiagnet Dump Files	Updated section
May 03, 2022	2.9.0	Basic Commands	Added new commands
		Port Counters	Updated the Port Counters Types and Port Counters Reset parameters
		Routing Validation	Updated Routing Validation Options
		Infiniband Security	Added the following sections: • VendorSpecific Key (VS Key) • CongestionControl Key (CC Key) • Manager2Node Key (M2N Key)
		Fat-Tree Topology Validation	Added new parameters

applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

Trademarks

NVIDIA and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.
