



Appendix - Diagnostic Utilities

Table of contents

InfiniBand Diagnostics Commands

Diagnostic Tools

Dependencies

Utilities Descriptions

Note

For UFM-SDN Appliance, all the below diagnostics commands have ib prefix.

For example, for UFM-SDN Appliance, the command `ibstat` is `ib ibstat`.

InfiniBand Diagnostics Commands

Command	Description
<code>ibstat</code>	Shows the host adapters status.
<code>ibstatus</code>	Similar to <code>ibstat</code> but implemented as a script.
<code>ibnetdiscover</code>	Scans the topology.
<code>ibaddr</code>	Shows the LID range and default GID of the target (default is the local port).
<code>ibroute</code>	Displays unicast and multicast forwarding tables of the switches.
<code>ibtracert</code>	Displays unicast or multicast route from source to destination.
<code>ibping</code>	Uses vendor MADs to validate connectivity between InfiniBand nodes. On exit, (IP) ping-like output is shown.
<code>ibsysstat</code>	Obtains basic information for the specific node which may be remote. This information includes: hostname, CPUs, memory utilization.
<code>sminfo</code>	Queries the SMInfo attribute on a node.
<code>smpdump</code>	A general purpose SMP utility which gets SM attributes from a specified SMA. The result is dumped in hex by default.
<code>smpquery</code>	Enables a basic subset of standard SMP queries including the following: node info, node description, switch info, port info. Fields are displayed in human readable format.

Command	Description
perfquery	Dumps (and optionally clears) the performance counters of the destination port (including error counters).
ibswitches	Scans the net or uses existing net topology file and lists all switches.
ibhosts	Scans the net or uses existing net topology file and lists all hosts.
ibnodes	Scans the net or uses existing net topology file and lists all nodes.
ibportstate	Gets the logical and physical port states of an InfiniBand port or disables or enables the port (only on a switch). Note: This tool can change port settings. Should be used with caution.
saquery	Issues SA queries.
ibdiagnet	ibdiagnet scans the fabric using directed route packets and extracts all the available information regarding its connectivity and devices.
ibnetsplit	Automatically groups hosts and creates scripts that can be run to split the network into sub-networks each containing one group of hosts.
lbqueryerrors	Queries IB spec-defined errors from all fabric ports. Note: This tool can change reset port counters Should be used with caution.
smparquery	Queries adaptive-routing related settings from a particular switch. Note: This tool can change reset port counters Should be used with caution.

Diagnostic Tools

Model of operation: All utilities use direct MAD access to operate. Operations that require QP 0 mads only, may use direct routed mads, and therefore may work even in subnets that are not configured. Almost all utilities can operate without accessing the SM, unless GUID to lid translation is required.

Dependencies

Multiple port/Multiple CA support:

When no InfiniBand device or port is specified (as shown in the following example for "Local umad parameters"), the tools select the interface port to use by the following criteria:

1. The first InfiniBand ACTIVE port.
2. If not found, the first InfiniBand port that is UP (physical link up).

If a port and/or CA name is specified, the **tool** attempts to fulfill the user's request and will fail if it is not possible.

For example:

```
ibaddr # use the 'best port'  
ibaddr -C mthca1 # pick the best port from mthca1 only.  
ibaddr -P 2 # use the second (active/up) port from the first available IB device.  
ibaddr -C mthca0 -P 2 # use the specified port only.
```

Common Options & Flags

Most diagnostics take the following flags. The exact list of supported flags per utility can be found in the usage message and can be shown using `util_name -h` syntax.

```
# Debugging flags  
-d raise the IB debugging level. May be used several times (-ddd or -d -d -d).  
-e show umad send receive errors (timeouts and others)  
-h show the usage message  
-v increase the application verbosity level.  
May be used several times (-vv or -v -v -v)  
-V show the internal version info.
```

```
# Addressing flags  
-D use directed path address arguments.  
The path is a comma separated list of out ports.  
Examples:  
"0" # self port  
"0,1,2,1,4" # out via port 1, then 2, ...  
-G use GUID address arguments.  
In most cases, it is the Port GUID.  
Examples:  
"0x08f1040023"
```

```
-s <smlid> use 'smlid' as the target lid for SA queries.
```

```
# Local umad parameters:
```

```
-C <ca_name> use the specified ca_name.
```

```
-P <ca_port> use the specified ca_port.
```

```
-t <timeout_ms> override the default timeout for the  
solicited mads.
```

CLI notation: all utilities use the POSIX style notation, meaning that all options (flags) must precede all arguments (parameters).

Utilities Descriptions

ibstatus

A script that displays basic information obtained from the local InfiniBand driver. Output includes LID, SMLID, port state, link width active, and port physical state.

Syntax

```
ibstatus [-h] [devname[:port]]
```

Examples:

```
ibstatus # display status of all IB ports  
ibstatus mthca1 # status of mthca1 ports  
ibstatus mthca1:1 mthca0:2 # show status of specified ports
```

See also: `ibstat`

ibstat

Similar to the `ibstatus` utility but implemented as a binary and not as a script. Includes options to list CAs and/or ports.

Syntax

```
ibstat [-d(ebug) -l(ist_of_cas) -p(ort_list) -s(hort)] <ca_name> [portnum]
```

Examples:

```
ibstat # display status of all IB ports
ibstat mthca1 # status of mthca1 ports
ibstat mthca1 2 # show status of specified ports
ibstat -p mthca0 # list the port guids of mthca0
ibstat -l # list all CA names
```

See also: `ibstatus`

ibroute

Uses SMPs to display the forwarding tables (unicast (LinearForwardingTable or LFT) or multicast (MulticastForwardingTable or MFT)) for the specified switch LID and the optional lid (mlid) range. The default range is all valid entries in the range 1...FDBTop.

Syntax

```
ibroute [options] <switch_addr> [<startlid> [<endlid>]]
```

Nonstandard flags:

```
-a show all lids in range, even invalid entries.
-n do not try to resolve destinations.
-M show multicast forwarding tables. In this case the range
parameters are specifying mlid range.
node-name-map node name map file
```

Examples:

```
ibroute 2 # dump all valid entries of switch lid 2
ibroute 2 15 # dump entries in the range 15...FDBTop.
ibroute -a 2 10 20 # dump all entries in the range 10..20
```

```
ibroute -n 2 # simple format
ibroute -M 2 # show multicast tables
```

See also: `ibtracert`

ibtracert

Uses SMPs to trace the path from a source GID/LID to a destination GID/LID. Each hop along the path is displayed until the destination is reached or a hop does not respond. By using the `-m` option, multicast path tracing can be performed between source and destination nodes.

Syntax

```
ibtracert [options] <src-addr> <dest-addr>
```

Nonstandard flags:

```
-n simple format; don't show additional information.
-m <mlid> show the multicast trace of the specified mlid.
-f <force> force
node-name-map node name map file
```

Examples:

```
ibtracert 2 23 # show trace between lid 2 and 23
ibtracert -m 0xc000 3 5 # show multicast trace between lid 3
and 5 for mcast lid 0xc000.
```

smpquery

Enables a basic subset of standard SMP queries including the following node info, node description, switch info, port info. Fields are displayed in human readable format.

Syntax


```
smpquery [options] <op> <dest_addr> [op_params]
```

Currently supported operations and their parameters:

```
nodeinfo <addr>
nodedesc <addr>
portinfo <addr> [<portnum>] # default port is zero
switchinfo <addr>
pkeys <addr> [<portnum>]
sl2vl <addr> [<portnum>]
vlarb <addr> [<portnum>]
GUIDInfo (GI) <addr>
MInxExtPortInfo (MEPI) <addr> [<portnum>]
Combined (-c) : use Combined route address argument
node-name-map : node name map file
extended (-x) : use extended speeds
```

Examples:

```
smpquery nodeinfo 2 # show nodeinfo for lid 2
smpquery portinfo 2 5 # show portinfo for lid 2 port 5
```

smpdump

A general purpose SMP utility that gets SM attributes from a specified SMA. The result is dumped in hex by default.

Syntax

```
smpdump [options] <dest_addr> <attr> [mod]
```

Nonstandard flags:

```
-s show output as string
```

Examples:

```
smpdump -D 0,1,2 0x15 2 # port info, port 2  
smpdump 3 0x15 2 # port info, lid 3 port 2
```

ibaddr

Can be used to show the LID and GID addresses of the specified port or the local port by default. This utility can be used as simple address resolver.

Syntax

```
ibaddr [options] [<dest_addr>]
```

Nonstandard flags:


```
gid_show (-g) : show gid address only  
lid_show (-l) : show lid range only  
Lid_show (-L) : show lid range (in decimal) only
```

Examples:

```
ibaddr # show local address  
ibaddr 2 # show address of the specified port lid  
ibaddr -G 0x8f1040023 # show address of the specified port guid
```

sminfo

Issues and dumps the output of an sminfo query in human readable format. The target SM is the one listed in the local port info or the SM specified by the optional SM LID or by the SM direct routed path.

 **Warning**

CAUTION: Using sminfo for any purpose other than a simple query might result in a malfunction of the target SM.

Syntax

```
sminfo [options] <sm_lid | sm_dr_path> [sminfo_modifier]
```

Nonstandard flags:

```
-s <state> # use the specified state in sminfo mad  
-p <priority> # use the specified priority in sminfo mad  
-a <activity> # use the specified activity in sminfo mad
```

Examples:

```
sminfo # show sminfo of SM listed in local portinfo  
sminfo 2 # query SM on port lid 2
```

perfquery

Uses PerfMgt GMPs to obtain the PortCounters (basic performance and error counters) from the Performance Management Agent (PMA) at the node specified. Optionally show aggregated counters for all ports of node. Also, optionally, reset after read, or only reset counters.

```
perfquery [options] [<lid | guid> [[port] [reset_mask]]]
```

Nonstandard flags:

```
-a Shows aggregated counters for all ports of the destination lid.  
-r Resets counters after read.
```

- R Resets only counters.
- Extended (-x) Shows extended port counters
- Xmtsl (-X) Shows Xmt SL port counters
- Rcvsl ,(-S) Shows Rcv SL port counters
- Xmtdisc (-D) Shows Xmt Discard Details
- rcvrr, (-E) Shows Rcv Error Details
- extended_speeds (-T) Shows port extended speeds counters
- oprcvcounters Shows Rcv Counters per Op code
- flowctlcounters Shows flow control counters
- vloppackets Shows packets received per Op code per VL
- vlopdata Shows data received per Op code per VL
- vlxmitflowctlerrors Shows flow control update errors per VL
- vlxmitcounters Shows ticks waiting to transmit counters per VL
- swportvlcong Shows sw port VL congestion
- rcvcc Shows Rcv congestion control counters
- slrcvfecn Shows SL Rcv FECN counters
- slrcvbecn Shows SL Rcv BECN counters
- xmitcc Shows Xmit congestion control counters
- vlxmittlecc Shows VL Xmit Time congestion control counters
- smplctl (-c) Shows samples control
- loop_ports (-l) Iterates through each port

Examples:

```

perfquery # read local port's performance counters
perfquery 32 1 # read performance counters from lid 32, port 1
perfquery -a 32 # read from lid 32 aggregated performance counters
perfquery -r 32 1 # read performance counters from lid 32 port 1 and reset
perfquery -R 32 1 # reset performance counters of lid 32 port 1 only
perfquery -R -a 32 # reset performance counters of all lid 32 ports
perfquery -R 32 2 0xf000 # reset only non-error counters of lid 32 port 2

```

ibping

Uses vendor mads to validate connectivity between InfiniBand nodes. On exit, (IP) ping like output is show. ibping is run as client/server. The default is to run as client. Note also

that a default ping server is implemented within the kernel.

Syntax

```
ibping [options] <dest lid | guid>
```

Nonstandard flags:

```
-c <count> stop after count packets  
-f flood destination: send packets back to back w/o delay  
-o <oui> use specified OUI number to multiplex vendor MADs  
-S start in server mode (do not return)
```

ibnetdiscover

Performs InfiniBand subnet discovery and outputs a human readable topology file. GUIDs, node types, and port numbers are displayed as well as port LIDs and node descriptions. All nodes (and links) are displayed (full topology). This utility can also be used to list the current connected nodes. The output is printed to the standard output unless a topology file is specified.

Syntax

```
ibnetdiscover [options] [<topology-filename>]
```

Nonstandard flags:

```
l Lists connected nodes  
H Lists connected HCAs  
S Lists connected switches  
g Groups  
full (-f) Shows full information (ports' speed and width, vlcaps)  
show (-s) Shows more information  
Router_list (-R) Lists connected routers  
node-name-map Nodes name map file  
cache filename to cache ibnetdiscover data to
```

load-cache filename of ibnetdiscover cache to load
diff filename of ibnetdiscover cache to diff
diffcheck Specifies checks to execute for --diff
ports : (-p) Obtains a ports report
max_hops (-m) Reports max hops discovered by the library
outstanding_smps (-o) Specifies the number of outstanding SMP's which should be issued during the scan

ibhosts

Traces the InfiniBand subnet topology or uses an already saved topology file to extract the CA nodes.

Syntax

```
ibhosts [-h] [<topology-file>]
```

Dependencies: ibnetdiscover, ibnetdiscover format

ibswitches

Traces the InfiniBand subnet topology or uses an already saved topology file to extract the InfiniBand switches.

Syntax

```
ibswitches [-h] [<topology-file>]
```

Dependencies: ibnetdiscover, ibnetdiscover format

ibportstate

Enables the port state and port physical state of an InfiniBand port to be queried or a switch port to be disabled or enabled.

Syntax

```
ibportstate [-d(ebug) -e(rr_show) -v(erbose) -D(irect) -G(uid) -s smlid -V(ersion) -C
ca_name -P ca_port -t timeout_ms] <dest dr_path | lid | guid> <portnum> [<op>]
```

Supported ops: enable, disable, query, on, off, reset, speed, espeed, fdr10, width, down, arm, active, vls, mtu, lid, smlid, lmc, mkey, mkeylease, mkeyprot

Examples:

```
ibportstate 3 1 disable # by lid
ibportstate -G 0x2C9000100D051 1 enable # by guid
ibportstate -D 0 1 # by direct route
```

ibnodes

Uses the current InfiniBand subnet topology or an already saved topology file and extracts the InfiniBand nodes (CAs and switches).

Syntax

```
ibnodes [<topology-file>]
```

Dependencies: ibnetdiscover, ibnetdiscover format

ibqueryerrors

Queries or clears the PMA error counters in PortCounters by walking the InfiniBand subnet topology.

```
ibqueryerrors [options]
```

Syntax

Options:

--suppress, -s <err1,err2,...> suppress errors listed

--suppress-common, -c suppress some of the common counters

--node-name-map <file> node name map file

--port-guid, -G <port_guid> report the node containing the port specified by <port_guid>

--, -S <port_guid> Same as "-G" for backward compatibility

--Direct, -D <dr_path> report the node containing the port specified by <dr_path>

--skip-sl don't obtain SL to all destinations

--report-port, -r report port link information

--threshold-file <val> specify an alternate threshold file, [default](#): /etc/infiniband-diags/error_thresholds

--GNDN, -R (This option is obsolete and does nothing)

--data include data counters for ports with errors

--switch print data for switches only

--ca print data for CA's only

--router print data for routers only

--details include transmit discard details

--counters print data counters only

--clear-errors, -k Clear error counters after read

--clear-counts, -K Clear data counters after read

--load-cache <file> filename of ibnetdiscover cache to load

--outstanding_smps, -o <val> specify the number of outstanding SMP's which should be issued during the scan

--config, -z <config> use config file, [default](#): /etc/infiniband-diags/ibdiag.conf

--Ca, -C <ca> Ca name to use

--Port, -P <port> Ca port number to use

--timeout, -t <ms> timeout in ms

--m_key, -y <key> M_Key to use in request

--errors, -e show send and receive errors

--verbose, -v increase verbosity level

--debug, -d raise debug level

--help, -h help message

--version, -V show version

smparquery

Issues Adaptive routing-related queries to the fabric switch.

Syntax

Supported ops (and aliases, *case* insensitive):

ARInfo (ARI) <addr>

ARGroupTable (ARGT) <addr> [<plft>] [<group_table>] [<blocknum>]

ARLFTTable (ARLT) <addr> [<plft>] [<blocknum>]

PLFTInfo (PLFTI) <addr>

PLFTDef (PLFTD) <addr> [<blocknum>]

PLFTMap (PLFTM) <addr> [<plft>] [<control_map>]

PortSLToPLFTMap (PLFTP) <addr> [<blocknum>]

RNSubGroupDirectionTable (DIRT) <addr> [<blocknum>]

RNGenStringTable (GSTR) <addr> [<plft>] [<blocknum>]

RNGenBySubGroupPriority (GSGP) <addr>

RNRcvString (RSTR) <addr> [<blocknum>]

RNXmitPortMask (RNXM) <addr> [<blocknum>]

PortRNCounters (RNPC) <addr>

Options:

Main

-C | --Ca <ca> : Ca name to use

-P | --Port <port> : Ca port number to use

-D | --Direct : use Direct address argument

-L | --Lid : use LID address argument

-h | --help : help message

-V | --version : show version

-d | --debug : Print debug logs

saquery

Issues SA queries.

Syntax

```
saquery [-h -d -P -N -L -G -s -g][<name>]
```

Queries node records by default.

d	Enables debugging
P	Gets PathRecord info
N	Gets NodeRecord info
L (-L)	Returns just the Lid of the name specified
G (-G)	Returns just the Guid of the name specified
S (-S)	Returns the PortInfoRecords with isSM capability mask bit on
G (-g)	Gets multicast group info
L (-l)	Returns the unique Lid of the name specified
O (-O)	Returns name for the Lid specified
m(-m)	Gets multicast member info (if multicast group specified, list member GIDs only for group specified for example 'saquery -m 0xC000')
x (-x)	Gets LinkRecord info"
c (-c)	Gets the SA's class port info
S (-S)	Gets ServiceRecord info
l (-l)	Gets InformInfoRecord (subscription) info
list (-D)	the node desc of the CA's
src-to-dst (<src:dst>)	Gets a PathRecord for <src:dst> where src and dst are either node names or LIDs
sgid-to-dgid (<sgid-dgid>)	Gets a PathRecord for <sgid-dgid> where sgid and dgid are addresses in IPv6 format
node-name-map	Specifies a node name map file
smkey <val>	SA SM_Key value for the query. If non-numeric value (like 'x') is specified then saquery will prompt for a value. Default (when not specified here or in ibdiag.conf) is to use SM_Key == 0 (or \"untrusted\")
slid <lid>	Source LID (PathRecord)
dlid <lid>	Destination LID (PathRecord)
mild <lid>	Multicast LID (MCMemberRecord)
sgid <gid>	Source GID (IPv6 format) (PathRecord)
dgid <gid>	Destination GID (IPv6 format) (PathRecord)
gid <gid>	Port GID (MCMemberRecord)
mgid <gid>	Multicast GID (MCMemberRecord)
Reversible", 'r', 1, NULL"	Reversible path (PathRecord)
numb_path ", 'n', 1, NULL"	Number of paths (PathRecord)
pkey: P_Key (PathRecord, MCMemberRecord).	QoS Class (PathRecord)
qos_class (-Q)	Service level (PathRecord, MCMemberRecord)
sl	MTU and selector (PathRecord, MCMemberRecord)
mtu : (-M)	Rate and selector (PathRecord, MCMemberRecord)
rate (-R)	Packet lifetime and selector (PathRecord, MCMemberRecord)
pkt_lifetime	If non-numeric value (like 'x') is specified then saquery will prompt for a value. Traffic Class (PathRecord, MCMemberRecord)

qkey (-q)	Flow Label (PathRecord, MCMemberRecord)
(PathRecord, MCMemberRecord).	Hop limit (PathRecord, MCMemberRecord)
tclass (-T)	Scope (MCMemberRecord)
flow_label : (-F)	Join state (MCMemberRecord)
hop_limit : (-H)	Proxy join (MCMemberRecord)
scope	ServiceID (PathRecord)
join_state (-J)	
proxy_join (-X)	
service_id	

Dependencies: OpenSM libvendor, OpenSM libopensm, libibumad

ibsysstat

```
ibsysstat [options] <dest lid | guid> [<op>]
```

Nonstandard flags:

Current supported operations:

ping - verify connectivity to server (default)

host - obtain host information from server

cpu - obtain cpu information from server

-o <oui> use specified OUI number to multiplex vendor mads

-S start in server mode (do not return)

ibnetsplit

Automatically groups hosts and creates scripts that can be run in order to split the network into sub-networks containing one group of hosts.

Syntax

- Group:

```
ibnetsplit [-v][-h][-g grp-file] -s <.lst | .net | .topo> <-r head-ports | -d max-dist>
```

- Split:

```
ibnetsplit [-v][-h][-g grp-file] -s <.lst|.net|.topo>  
-o out-dir
```

- Combined:

```
ibnetsplit [-v][-h][-g grp-file] -s <.lst|.net|.topo> <-r head-ports|-d max-dist> -o  
out-dir
```

Usage

- Grouping:
The grouping is performed if the -r or -d options are provided.
- If the -r is provided with a file containing group head ports, the algorithm examines the hosts distance from the set of node ports provided in the head-ports file (these are expected to be the ports running standby SM's).
- If the -d is provided with a maximum distance of the hosts in each group, the algorithm partition the hosts by that distance.

Note

This method of analyzation may not be suitable for some topologies.

The results of the identified groups are printed into the file defined by the -g option (default ibnetsplit.groups) and can be manually edited. For groups where the head port is a switch, the group file uses the FIRST host port as the port to run the isolation script from.

- Splitting:
 - If the -o flag is included, this algorithm analyzes the MinHop table of the topology and identifies the set of links and switches that may potentially be used for routing each group ports. The cross-switch links between switches of

the group to other switches are declared as split-links and the commands to turn them off using Directed Routes from the original Group Head ports are written into the out-dir provided by the -o flag.

Both stages require a subnet definition file to be provided by the -s flag. The supported formats for subnet definition are:

- *.net - for ibnetdiscover
- *.lst - for opensm-subnet.lst or ibiagnet.lst
- *.topo - for a topology file

HEAD PORTS FILE

This file is provided by the user and defines the ports by which grouping of the other host ports is defined.

Format:

Each line should contain either the name or the GUID of a single port. For switches the port number shall be 0.

```
<node-name>/P<port-num> | <PGUID>
```

GROUPS FILE

This file is generated by the program if the head-ports file is provided to it. Alternatively it can be provided (or edited) by the user if different grouping is desired. The generated script for isolating or connecting the group should be run from the first node in each group.

Format:

Each line may be either:

```
GROUP: <group name>  
<node-name>/P<port-num> | <PGUID>
```

ibdiagnet

ibdiagnet scans the fabric using directed route packets and extracts all the available information regarding its connectivity and devices.

It then produces the following files in the output directory (see below):

- "ibdiagnet2.log" - A log file with detailed information.
- "ibdiagnet2.db_csv" - A dump of the internal tool database.
- "ibdiagnet2.lst" - A list of all the nodes, ports and links in the fabric.
- "[ibdiagnet2.pm](#)" - A dump of all the nodes PM counters.
- "ibdiagnet2.mlnx_cntrs" - A dump of all the nodes Mellanox diagnostic counters.
- "ibdiagnet2.net_dump" - A dump of all the links and their features.
- "ibdiagnet2.pkey" - A list of all pkeys found in the fabric.
- "ibdiagnet2.aguid" - A list of all alias GUIDs found in the fabric.
- "[ibdiagnet2.sm](#)" - A dump of all the SM (state and priority) in the fabric.
- "ibdiagnet2.fdfs" - A dump of unicast forwarding tables of the fabric switches.
- "ibdiagnet2.mcfdfs" - A dump of multicast forwarding tables of the fabric switches.
- "ibdiagnet2.slvl" - A dump of SLVL tables of the fabric switches.
- "ibdiagnet2.nodes_info" - A dump of all the nodes vendor specific general information for nodes who supports it.
- "ibdiagnet2.plft" - A dump of Private LFT Mapping of the fabric switches.
- "[ibdiagnet2.ar](#)" - A dump of Adaptive Routing configuration of the fabric switches.
- "ibdiagnet2.vl2vl" - A dump of VL to VL configuration of the fabric switches.

Load plugins from:

`/tmp/ibutils2/share/ibdiagnet2.1.1/plugins/`

You can specify additional 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
```

Syntax

```
[-i|--device <dev-name>] [-p|--port <port-num>]
[-g|--guid <GUID in hex>] [--skip <stage>]
[--skip_plugin <library name>] [--sc]
[--scr] [--pc] [-P|--counter <<PM>=<value>>]
[--pm_pause_time <seconds>] [--ber_test]
[--ber_thresh <value>] [--llr_active_cell <64|128>]
[--extended_speeds <dev-type>] [--pm_per_lane]
[--ls <2.5|5|10|14|25|FDR10|EDR20>]
[--lw <1x|4x|8x|12x>] [--screen_num_errs <num>]
[--smp_window <num>] [--gmp_window <num>]
[--max_hops <max-hops>] [--read_capability <file name>]
[--write_capability <file name>]
[--back_compat_db <version.sub_version>]
[-V|--version] [-h|--help] [-H|--deep_help]
[--virtual] [--mads_timeout <mads-timeout>]
[--mads_retries <mads-retries>] [-m|--map <map-file>]
[--vlr <file>] [-r|--routing] [--r_opt <[vs,][mcast,]>]
[--sa_dump <file>] [-u|--fat_tree]
[--scope <file.guid>] [--exclude_scope <file.guid>]
[-w|--write_topo_file <file name>]
[-t|--topo_file <file>] [--out_ibnl_dir <directory>]
[-o|--output_path <directory>]
Cable Diagnostic (Plugin)
[--get_cable_info] [--cable_info_disconnected]
Phy Diagnostic (Plugin)
[--get_phy_info] [--reset_phy_info]
```

Options

-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).

-p|--port <port-num> : Specifies the local device's port number used to connect to the IB fabric.

-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.

--skip <stage> : Skip the executions of the given stage.
Applicable skip stages (vs_cap_smp vs_cap_gmp | links | pm | speed_width_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).

--sc : Provides a report of Mellanox counters

--scr : Reset all the Mellanox counters (if --sc option selected).

--pc : Reset all the fabric PM counters.

-P|--counter <<PM>=<value>> : If any of the provided PM is greater than its provided value then print it.

--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).

--ber_test : Provides a BER test for each port.
Calculate BER for each port and check no BER value has exceeds the BER threshold. (default threshold="10^-12").

--ber_thresh <value> : Specifies the threshold value for the

BER test. The reciprocal number of the BER should be provided. Example: for 10^{-12} than value need to be 1000000000000 or 0xe8d4a51000 (10^{12}). If threshold given is 0 than all BER values for all ports will be reported.

--llr_active_cell <64 | 128> : Specifies the LLR active cell size for BER test, when LLR is active in the fabric.

--extended_speeds <dev-type> : Collect and test port extended speeds counters. dev-type: (sw | all).

--pm_per_lane : List all counters per lane (when available).

--ls <0 | 2.5 | 5 | 10 | 14 | 25 | 50 | 100 | FDR10> : Specifies the expected link speed.

--lw <1x | 4x | 8x | 12x> : Specifies the expected link width.

--screen_num_errs <num> : Specifies the threshold for printing errors to screen. (default=5).

--smp_window <num> : Max smp MADs on wire. (default=8).

--gmp_window <num> : Max gmp MADs on wire. (default=128).

--max_hops <max-hops> : Specifies the maximum hops for the discovery process. (default=64).

--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.

--write_capability <file name> : Write out an example file for capability masks configuration, and also the default capability masks for some devices.

--back_compat_db <version.sub_version> : Show ports section in "ibdiagnet2.db_csv" according to given version. Default version 2.0.

-V | --version : Prints the version of the tool.

-h | --help : Prints help information (without

plugins help if exists).

-H | --deep_help : Prints deep help information (including plugins help).

--virtual : Discover VPorts during discovery stage.

--mads_timeout <mads-timeout> : Specifies the timeout (in milliseconds) for sent and received mads. (default=500).

--mads_retries <mads-retries> : Specifies the number of retries for every timeout mad. (default=2).

-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".

--src_lid <src-lid> : source lid

--dest_lid <dest-lid> : destination lid

--dr_path <dr-path> : direct route path

-o | --output_path <directory> : Specifies the directory where the Output files will be placed.

(default="/var/tmp/ibdiagpath/").

Cable Diagnostic (Plugin)

--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.

Phy Diagnostic (Plugin)

--get_phy_info : Indicates to query all ports for phy information.

--reset_phy_info : Indicates to clear all ports phy information.

ibdiagpath

ibdiagpath scans the fabric using directed route packets and extracts all the available information regarding its connectivity and devices. It then produces the following files in the output directory (see below):

- "ibdiagnet2.log" - A log file with detailed information.
- "ibdiagnet2.db_csv" - A dump of the internal tool database.
- "ibdiagnet2.lst" - A list of all the nodes, ports and links in the fabric.
- "ibdiagnet2.pm" - A dump of all the nodes PM counters.
- "ibdiagnet2.mlnx_cntrs" - A dump of all the nodes Mellanox diagnostic counters.
- "ibdiagnet2.net_dump" - A dump of all the links and their features.

Cable Diagnostic (Plugin):

This plugin performs cable diagnostic. It can collect cable info (vendor, PN, OUI etc..) on each valid QSFP cable, if specified.

It produces the following files in the output directory (see below):

- "ibdiagnet2.cables" - In case specified to collect cable info, this file will contain all collected cable info.

Phy Diagnostic (Plugin)

This plugin performs phy diagnostic.

Load Plugins from:

```
/tmp/ibutils2/share/ibdiagnet2.1.1/plugins/
```

You can specify additional paths to be looked in with "IBDIAGNET_PLUGINS_PATH" env variableLoad plugins from:

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

Syntax

```
[-i|--device <dev-name>] [-p|--port <port-num>]
[-g|--guid <GUID in hex>] [--skip <stage>]
[--skip_plugin <library name>] [--sc]
[--scr] [--pc] [-P|--counter <<PM>=<value>>]
[--pm_pause_time <seconds>] [--ber_test]
[--ber_thresh <value>] [--llr_active_cell <64|128>]
[--extended_speeds <dev-type>] [--pm_per_lane]
[--ls <2.5|5|10|14|25|FDR10|EDR20>]
[--lw <1x|4x|8x|12x>] [--screen_num_errs <num>]
[--smp_window <num>] [--gmp_window <num>]
[--max_hops <max-hops>] [--read_capability <file name>]
[--write_capability <file name>]
[--back_compat_db <version.sub_version>]
[-V|--version] [-h|--help] [-H|--deep_help]
[--virtual] [--mads_timeout <mads-timeout>]
[--mads_retries <mads-retries>] [-m|--map <map-file>]
[--src_lid <src-lid>] [--dest_lid <dest-lid>]
[--dr_path <dr-path>] [-o|--output_path <directory>]
Cable Diagnostic (Plugin)
[--get_cable_info] [--cable_info_disconnected]
Phy Diagnostic (Plugin)
[--get_phy_info] [--reset_phy_info]
```

Options

-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).
-p --port <port-num>	:Specifies the local device's port number used to connect to the IB fabric.
-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.
--skip <stage>	:Skip the executions of the given stage. Applicable skip stages: (vs_cap_smp vs_cap_gmp links pm speed_width_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).
--sc	:Provides a report of Mellanox counters
--scr	:Reset all the Mellanox counters (if -sc option selected).
--pc	:Reset all the fabric PM counters.

-P --counter	:If any of the provided PM is greater then its provided value than print it.
<<PM>=	:Specifies the seconds to wait between first counters sample and second counters sample.
<value>>	If seconds given is 0 than no second counters sample will be done. (default=1).
--	:Provides a BER test for each port. Calculate BER for each port and check no BER value has exceeds the BER threshold.(default threshold="10^-12").
pm_pause_time	exceeds the BER threshold.(default threshold="10^-12").
<seconds>	:Specifies the threshold value for the BER test. The reciprocal number of the BER should be provided. Example: for 10^-12 than value need to be 1000000000000 or 0xe8d4a51000(10^12).If threshold given is 0 than all BER values for all ports will be reported.
--ber_test	
--ber_thresh	
<value>	
--llr_active_cell	:Specifies the LLR active cell size for BER test, when LLR is active in the fabric.
<64 128>	:Collect and test port extended speeds counters. dev-type: (sw all).
--	
extended_spee	:Specifies the expected link speed.
ds <dev-type>	:Specifies the expected link width.
--pm_per_lane	:Specifies the threshold for printing errors to screen. (default=5).
:List all	:Max smp MADs on wire. (default=8).
counters per	:Max gmp MADs on wire. (default=128).
lane (when	:Specifies the maximum hops for the discovery process.(default=64).
available).	:Specifies capability masks configuration file, giving capability mask configuration for the fabric. ibdiagnet will use this mapping for Vendor Specific MADs sending.
--ls	
<2.5 5 10 14	:Write out an example file for capability masks configuration, and also the default capability masks for some devices.
25 FDR10 EDR	
20>	:Show ports section in "ibdiagnet2.db_csv" according to given version. Default version 2.0.
--lw	:Prints the version of the tool.
<1x 4x 8x 12x	:Prints help information (without plugins help if exists).
>	:Prints deep help information (including plugins help).
--	:Discover VPorts during discovery stage.
screen_num_er	:Specifies the timeout (in milliseconds) for sent and received mads.(default=500).
rs <num>	:Specifies the number of retries for every timeout mad.(default=2).
--smp_window	:Specifies mapping file, that maps node guid to name (format: 0x[0-9a-fA-F]+ "name").
<num>	Mapping file can also be specified by environment variable
--gmp_window	"IBUTILS_NODE_NAME_MAP_FILE_PATH".
<num>	:source lid
--max_hops	destination lid
<max-hops>	:direct route path
--	:Specifies the directory where the output files will be placed.
read_capability	(default="/var/tmp/ibdiagpath/").
<file name>	
--	:Indicates to query all QSFP cables for cable information. Cable information will be stored in "ibdiagnet2.cables".
write_capability	
<file name>	:Get cable info on disconnected ports.
--	
back_compat_d	:Indicates to query all ports for phy information.
b	:Indicates to clear all ports phy information.
<version.sub_ve	
rsion>	
-V --version	
-h --help	
-H --deep_help	
--virtual	

```
--mads_timeout  
<mads-  
timeout>  
--mads_retries  
<mads-retries>  
-m|--map  
<map-file>  
--src_lid <src-  
lid>  
--dest_lid <dest-  
lid>  
--dr_path <dr-  
path>  
-o|--  
output_path  
<directory>  
Cable  
Diagnostic  
(Plugin)  
--get_cable_info  
--  
cable_info_disc  
onnected  
Phy Diagnostic  
(Plugin)  
--get_phy_info  
--reset_phy_info
```

© Copyright 2024, NVIDIA. PDF Generated on 06/06/2024