MLNX_VPI for Windows README

Rev 2.1.2

www.mellanox.com

NOTE:

THIS INFORMATION IS PROVIDED BY MELLANOX FOR INFORMATIONAL PURPOSES ONLY AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL MELLANOX BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS HARDWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Mellanox Technologies
350 Oakmead Parkway, Suite 100
Sunnyvale, CA 94085
U.S.A.
www.mellanox.com
Tel: (408) 970-3400
Fax: (408) 970-3403

Mellanox Technologies, Ltd.
PO Box 586 Hermon Building
Yokneam 20692
Israel
Tel: +972-4-909-7200
Fax: +972-4-959-3245

Table of Contents

Revision History

Rev 2.1.2 – October 10, 2010

- Removed section Debug Options.
- Updated Section 9, “Starting and Verifying the IB Fabric,” on page 30
- Added Section 12, “InfiniBand Fabric Diagnostic Utilities,” on page 33 and its subsections
- Added Section 13, “InfiniBand Fabric Performance Utilities,” on page 67 and its subsections

Rev 2.1.1.1 – July 14, 2010

- Removed all references of InfiniHost® adapter since it is not supported starting with WinOF VPI v2.1.1.

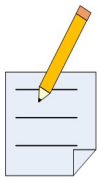
Rev 2.1.1 – May 2010

First release

Documentation Conventions

Typographical Conventions

Table 1 - Typographical Conventions

Description	Convention	Example
File names	file.extension	
Directory names	directory	
Commands and their parameters	command param1	mts3610-1 > show hosts
Required item	< >	
Optional item	[ ]	
Mutually exclusive parameters	{ p1, p2, p3 } or {p1 | p2 | p3}	
Optional mutually exclusive parameters	[p1 | p2 | p3]	
Prompt of a command in Standard mode	hostname >	mts3610-1 >
Prompt of a command in Enable mode	hostname #	mts3610-1 #
Prompt of a command in Config mode	hostname (config) #	mts3610-1 (config) #
Comments to explain command examples	//	// This is a comment
Variables for which users supply specific values	Italic font	*enable*
Emphasized words	Italic font	*These are emphasized words*
Note	<text> 	This is a note. 
Warning	<text> 	 Make sure to connect to the RS-232 RJ-45 port of the switch and not to the ETH port.

Common Abbreviations and Acronyms

Table 2 - Abbreviations and Acronyms

Abbreviation / Acronym	Whole Word / Description
B	(Capital) 'B' is used to indicate size in bytes or multiples of bytes (e.g., 1KB = 1024 bytes, and 1MB = 1048576 bytes)
b	(Small) 'b' is used to indicate size in bits or multiples of bits (e.g., 1Kb = 1024 bits)
Eth	Ethernet
FCoE	Fibre Channel over Ethernet
FW	Firmware
HCA	Host Channel Adapter
HW	Hardware
IB	InfiniBand
LSB	Least significant *byte*
lsb	Least significant *bit*
MSB	Most significant *byte*
msb	Most significant bit
NIC	Network Interface Card
SW	Software
VPI	Virtual Protocol Interconnect

1 Introduction

This is the README for the Mellanox WinOF VPI driver v2.1.2 package, distributed for Windows Server 2008 (x86 and x64) and Windows Server 2008 R2 (x64).

Mellanox WinOF VPI is composed of several software modules that contain an InfiniBand and/or,10Gb/s Ethernet network. The Mellanox WinOF VPI driver can be used in one of the following modes: 2 InfiniBand ports, 2 Ethernet ports, or 1 InfiniBand and 1 Ethernet port (that is, VPI mode).

Please refer to the MLNX_WinOF_IB_ReleaseNotes.txt file to check for known issues and fixed bugs for IB driver.

Please refer to the MLNX_WinOF_ETH_ReleaseNotes.txt file to check for known issues and fixed bugs for Ethernet driver.

1.1 Mellanox VPI Package Contents

The Mellanox WinOF for Windows package contains the following components:

- Core and ULPs
  - IB network adapter cards low-level drivers (mthca, mlx4)
  - IB Access Layer (IBAL)
  - Ethernet driver (ETH)
  - Upper Layer Protocols (ULPs):
    - IP over InfiniBand (IPoIB)
    - NetworkDirect (ND)
    - Winsock Direct (WSD)
    - Beta: Sockets Direct Protocol (SDP)
    - Beta: SCSI RDMA Protocol (SRP)
- Utilities
- SW Development Kit (SDK)
- Documentation

SDP and SRP are at Beta stage.

1.2 Hardware and Software Requirements

- Administrator privileges on your machine(s)
- Disk Space for installation: 100MB

1.3 Supported Network Adapter Cards and Firmware Versions

Mellanox WinOF VPI 2.1.2 supports the following Mellanox network adapter cards:

- VPI mode
  - ConnectX IB SDR/DDR/QDR (fw-25408 Rev 2.7.700)
  - ConnectX-2 IB SDR/DDR/QDR (fw-25408 Rev 2.7.9110)
- Ethernet only mode
  - ConnectX EN (fw-25408 Rev 2.7.700 or later)
  - ConnectX-2 EN (fw-25408 Rev 2.7.9110 or later)
- IB only mode
  - ConnectX (fw-25408 Rev 2.7.700 or later)
  - ConnectX-2 (fw-25408 Rev 2.7.9110 or later)
  - InfiniHost (fw-23108 Rev 3.5.000 or later)
  - InfiniHost III Ex (MemFree: fw-25218 Rev 5.3.000 or later; with memory: fw-25208 Rev 4.8.200 or later)
  - InfiniHost III Lx (fw-25204 Rev 1.2.000 or later)

Note: InfiniHost® adapters will be deprecated in the next MLNX_WinOF_VPI release.

Note: While installing MLNX_WinOF_VPI v2.1.2, please upgrade your firmware version to 2.7.9110 or higher. Please contact support@mellanox.com to get the binary file.

For official firmware versions please visit

http://www.mellanox.com-->Support > Firmware Download

1.4 Supported Operating Systems

Supported Operating Systems and Service Packs:

- Windows Server 2008 (x86, x64)
- Windows Server 2008 R2 (x64)
- Windows HPC Server 2008 (x64)
- Windows HPC Server 2008 R2(x64)

1.5 Managing Firmware

The adapter card may not have been shipped with the latest firmware version. This section describes how to update firmware.

1.5.1 Downloading the Firmware Tools Package

1. Download Mellanox Firmware Tools

Please download the current firmware tools package (MFT) from http://www.mellanox.com > Products > Software/Drivers > InfiniBand & VPI SW/Drivers > Firmware Tools.
The tools package to download is "MFT_SW for Windows" (WinMFT).

2. Install and Run WinMFT

   To install the WinMFT package, double click the MSI or run it from the command prompt.

On a Windows 2008 server, install the WinMFT package from the command line with administrator privileges.

Enter:

```
msiexec.exe /i WinMFT_<arch>_<version>.msi
```

3. Check the Device Status
   - To start the mst service (required by the tools), run > sc start mst
   - To check device status run > mst status

   If no card installation problems occur, the status command should produce the following output:

```
omt<device id>_pciconf0
omt<device id>_pci_cr0
```

   where device ID will be one of the supported PCI device IDs.

1.5.2 Download the Firmware Image of the Adapter Card

To download the correct card firmware image, please visit

http://www.mellanox.com > Support > Firmware Download

For help in identifying your adapter card, please visit

http://www.mellanox.comHome > Support > Firmware Downloads > Identifying Adapter Cards

1.5.3 Updating Adapter Card Firmware

Using a card specific binary firmware image file, enter the following command:

```
> flint -d mt<device id>_pci_cr0 -i <image_name.bin> burn
```

You may need to unzip the downloaded firmware image prior to burning.

For additional details, please check the MFT user's manual under

http://www.mellanox.com > Products > Adapter IB/VPI SW

2 Mellanox WinOF VPI Installation Process

Please refer to the Mellanox WinOF VPI Installation Guide for installation instructions.

3 Ethernet Driver

The Mellanox VPI WinOF driver release introduces the following capabilities:

- One or two ports
- Up to 16 Rx queues per port
- Rx steering mode (RSS)
- Hardware Tx/Rx checksum calculation
- Large Send Offload (i.e., TCP Segmentation Offload)
- Hardware multicast filtering
- Adaptive interrupt moderation
- Polling on send completion queue to decrease the number of interrupts (default: disabled)
- Polling on receive completion queue to decrease the number of interrupts (default: disabled)
- MSI-X support (only on Windows Server 2008 and higher)
- VLAN Tx/Rx acceleration (HW VLAN stripping/insertion)
- High Availability (HA) between ports and Mellanox NICs
- Load Balancing between ports and Mellanox NICs
- Quality of Service (QoS)
- HW VLAN filtering
- Tx arbitration mode: VLAN user-priority (off by default)

3.1 Performance Remarks

3.1.1 Performance Tuning

To improve performance, activate the performance tuning tool as follows:

1. Go to Control Panel.
2. Open Network Connections.
3. Right click on one of the entries "Mellanox ConnectX 10Gbit Ethernet Adapter" and select Properties.
4. Select the Performance tab.
5. Click General Tuning.

   Clicking the "General Tuning" button will change several registry entries (described below), and will check for system services that may decrease performance. It will also generate a log of the changes made. Users can refer to this log to restore the previous values.

The log path is:

%HOMEDRIVE%\windows\system32\logfiles\performancetunning.log.

This tuning is needed on one adapter only, and only once after the installation (as long as these entries are not changed directly in the registry, or by some other install or script).

You may need to reboot for the changes to take effect. You will be asked to reboot if necessary.

The registry entries that may be added/changed by this procedure are:

1. Windows 2003:

- Under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters:
  - TcpWindowSize, type REG_DWORD, value set to 512K.
  - Tcp1323Opts, type REG_DWORD, value set to 1.
  - SackOpts, type REG_DWORD, value set to 0.
  - EnableRss, type REG_DWORD, value set to 1.
  - RssBaseCpu, type REG_DWORD, value set to 1.
  - MaxNumRssCpus, type REG_DWORD, value set to 2.
- Under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AFD\Parameters:
  - FastSendDatagramThreshold, type REG_DWORD, value set to 64K. The following service is disabled:
  - "Windows Firewall/Internet Connection Sharing (ICS)"

2. Windows 2008 and Windows 2008-R2:

- Under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters:
  - SackOpts, type REG_DWORD, value set to 0.
- Under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AFD\Parameters:
  - FastSendDatagramThreshold, type REG_DWORD, value set to 64K.
- Under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Ndis\Parameters:
  - RssBaseCpu, type REG_DWORD, value set to 1.
  - MaxNumRssCpus, type REG_DWORD, value set to 4.

Note: In Windows 2008, RSS is enabled by default. However if you need to manually enable it, please run the "netsh int tcp set global rss = enabled" command.

Disable the time stamps on both sides. This is performed by means of the following command:

```
"netsh int tcp set global timestamps=disabled"
```

On some machines the following change may provide additional performance:

Change the send completion method from interrupt to polling as follows:

1. Open Device Manager
2. Right click the used Ethernet adapter (Mellanox ConnectX 10G Ethernet Adapter) and select Properties.
3. Select the Advanced tab.

4. Select Performance Options and then click Properties.
5. Select Send Completion Method.
6. Change the value to polling.
7. Click OK twice.

3.1.2 Known Performance Issues

- On Intel I/OAT supported systems, it is highly recommended to install and enable the latest I/OAT driver (download from www.intel.com).
- With I/OAT enabled, sending 256-byte messages or larger will activate I/OAT. This will cause a significant latency increase due to I/OAT algorithms. On the other hand, throughput will increase significantly when using I/OAT.
- On some systems, reducing the receive ring size ("Receive Ring Size" value under the Advanced tab) may improve performance.
- On some systems, changing the send completion method to polling ("Send Completion method" value under the Advanced tab) may improve performance.

3.2 Booting Windows from an iSCSI Target

Booting Windows from an iSCSI Target is supported on Windows Server 2008 and 2003 with the following limitations:

- 2008: Installing Windows Server 2008 directly to an iSCSI Target is not supported.
- 2003: Windows Server 2003 must be configured using a static IP address (and not through DHCP).

For more details on how to boot from a SAN using a Mellanox adapter card, please refer to http://www.etherboot.org/wiki/sanboot.

Also note that Mellanox has also tested the adapter card with a Windows iSCSI Target from StarWind (build 4.1).

3.3 Known Issues and Limitations

1. This release does not support installing MLNX_WinOF_VPI and other drivers such as MLNX_EN for Windows, Mellanox WinOF, WinOF, and Mellanox WinIB.
2. VLAN creation: VLANs on the same machine cannot be assigned dynamic IPs (using DHCP) of the same subnet.
3. Bundle creation (LBFO):
   a. After creating a bundle, all adapters in the bundle still appear in the Network Connections display.
   b. Once an adapter is part of a bundle, the following parameters cannot be changed: task offloading, RSS mode or MTU. To workaround this issue you need to (a) disassemble the bundle, (b) set the parameters for all the bundle adapters to the SAME values, and (c) reassemble the bundle.
   c. When creating a new bundle, it may take some time (up to one minute) until the OS presents the new bundle.

   d. When VLAN is present over the network adapter, LBFO will exclude this network adapter from the adapters list in LBFO GUI.

4. Replacing an adapter card may require reconfiguring LBFO bundles and/or VLAN adapters.
5. Uninstalling "Mellanox Virtual Miniport Driver" from a "Network properties" page may not work, and VLANs and LBFO bundles may still exist. Workaround:
   a. Open the device manager and select the Ethernet adapter that you are using (Mellanox ConnectX 10G Ethernet Adapter).
   b. Double-click on the adapter and select the VLAN tab, then remove all existing VLANs.
   c. Select the LBFO tab, then remove all existing bundles.

3.4 Troubleshooting

Issue # 1: The installation of MLNX_WinOF_VPI for Windows fails with the following (or a similar) error message:

This installation package is not supported by this processor type. Contact your product vendor."

Suggestion: This message is printed if you have downloaded and attempted to install an incorrect MSI -- for example, if you are trying to install a 64-bit MSI on a 32-bit machine (or vice versa).

Issue # 2: The performance is low.

Suggestion: This can be due to non-optimal system configuration. See the section "Performance Tuning" to take advantage of Mellanox 10 GBit NIC performance.

Issue # 3: The driver doesn't start.

Suggestion 1: This can happen due to an RSS configuration mismatch between the TCP stack and the Mellanox adapter. To confirm this scenario, open the event log and look under "System" for the "mlx4eth5" or "mlx4eth6" source. If found, enable RSS as follows:

1. For Windows 2003, in the TCP registry set the KEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\EnableRss registry value to 1.
2. For windows 2008, run the following command: "netsh int tcp set global rss = enabled".

Suggestion 2: This is a less recommended suggestion, and will cause low performance. Disable RSS on the adapter. To do this set RSS mode to "No Dynamic Rebalancing".

Issue # 4: The Ethernet driver fails to start. In the Event log, under the mlx4_bus source, the following error message appears: RUN_FW command failed with error -22

Suggestion: The error message indicates that the wrong firmware image has been programmed on the adapter card.

See http://www.mellanox.com > Support > Firmware Download

Issue # 5: The Ethernet driver fails to start. A yellow sign appears near the "Mellanox ConnectX 10Gb Ethernet Adapter" in the Device Manager display.

Suggestion: This can happen due to a hardware error. Try to disable and re-enable "Mellanox ConnectX Adapter" from the Device Manager display.

Issue # 6: No connectivity to a Fault Tolerance bundle while using network capture tools (e.g., Wireshark).

Suggestion: This can happen if the network capture tool captures the network traffic of the non-active adapter in the bundle. This is not allowed since the tool sets the packet filter to "promiscuous", thus causing traffic to be transferred on multiple interfaces. Close the network capture tool on the physical adapter card, and set it on the LBFO interface instead.

Issue # 7: No Ethernet connectivity on 1Gb/100Mb adapters after activating Performance Tuning (part of the installation).

Suggestion: This can happen due to adding a TcpWindowSize registry value. To resolve this issue, remove the value key under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\TcpWindowSize or set its value to 0xFFFF.

Issue # 8: System reboots on an I/OAT capable system on Windows Server 2008.

Suggestion: This may occur if you have an Intel I/OAT capable system with Direct Cache Access enabled, and 9K jumbo frames enabled. To resolve this issue, disable 9K jumbo frames.

Issue # 9: Packets are being lost.

Suggestion: This may occur if the port MTU has been set to a value higher than the maximum MTU supported by the switch.

Issue # 10: Issue(s) not listed above.

Suggestion: The MLNX_EN for Windows driver records events in the system log of the Windows event system. Using the event log you'll be able to identify, diagnose, and predict sources of system problems.

To see the log of events, open System Event Viewer as follows:

- Right click on My Computer, click Manage, and then click Event Viewer.

OR

- Click start-->Run and enter "eventvwr.exe".
- In Event Viewer, select the system log.

The following events are recorded:

- Mellanox ConnectX EN 10Gbit Ethernet Adapter <X> has been successfully initialized and enabled.
- Failed to initialize Mellanox ConnectX EN 10Gbit Ethernet Adapter.
- Mellanox ConnectX EN 10Gbit Ethernet Adapter <X> has been successfully initialized and enabled. The port's network address is <MAC Address>
- The Mellanox ConnectX EN 10Gbit Ethernet was reset.
- Failed to reset the Mellanox ConnectX EN 10Gbit Ethernet NIC. Try disabling then re-enabling the "Mellanox Ethernet Bus Driver" device via the Windows device manager.
- Mellanox ConnectX EN 10Gbit Ethernet Adapter <X> has been successfully stopped.
- Failed to initialize the Mellanox ConnectX EN 10Gbit Ethernet Adapter <X> because it uses old firmware version (<old firmware version>). You need to burn firmware version <new firmware version> or higher, and to restart your computer.
- Mellanox ConnectX EN 10Gbit Ethernet Adapter <X> device detected that the link connected to port <Y> is up, and has initiated normal operation.

- Mellanox ConnectX EN 10Gbit Ethernet Adapter <X> device detected that the link connected to port <Y> is down. This can occur if the physical link is disconnected or damaged, or if the other end-port is down.
- Mismatch in the configurations between the two ports may affect the performance. When Using MSI-X, both ports should use the same RSS mode. To fix the problem, configure the RSS mode of both ports to be the same in the driver GUI.
- Mellanox ConnectX EN 10Gbit Ethernet Adapter <X> device failed to create enough MSI-X vectors. The Network interface will not use MSI-X interrupts. This may affects the performance. To fix the problem, configure the number of MSI-X vectors in the registry to be at least <Y>.

4 IPoIB

IPoIB is a network driver implementation that enables transmitting IP and ARP protocol packets over an InfiniBand UD channel. The implementation conforms to the relevant IETF working group's RFCs (http://www.ietf.org).

4.1 IPoIB Drivers Overview

The Mellanox VPI WinOF driver release introduces the following capabilities:

- One or two ports
- Hardware Tx/Rx checksum calculation
- Large Send Offload (i.e., TCP Segmentation Offload)
- Hardware multicast filtering
- MSI-X support (only on Windows Server 2008 and higher)

4.2 IPoIB Setup

You may skip this section if you have configured one of the machines as a DHCP server for the IPoIB interface.

1. Go to Control Panel
2. Double-click Network Connections
3. Select the desired adapter (from Mellanox IPoIB Adapters), then right click and select Properties
4. Choose the General tab,
5. Select Internet Protocol (TCP/IP)
6. Click Properties
7. In the Internet Protocol (TCP/IP) Properties dialog box, click "Use the following IP address"
8. Enter the appropriate IP address and Subnet Mask. Use a different IP subnet for each IB port. IB ports IP subnet addresses must be different from Ethernet subnet addresses. In most cases the first number of an IP address is a constant, therefore it is common to assign new IPoIB addresses by changing the first number. For example:
   - Host Ethernet IP: 10.2.3.4
   - IPoIB IP address: 11.2.3.4

OpenSM must be active continuously on at least one machine in the cluster to allow proper IPoIB functioning.

4.3 Performance Remarks

4.3.1 Tunable Performance Parameters

The file IPoIB_registry_values.pdf provides the complete list of registry entries that may be added/changed by the performance tuning procedure described in "Performance Tuning" on page 20.

The following is a list of key parameters for performance tuning.

- Payload MTU

  The maximum available size of IPoIB transfer unit. It should be decremented by the size of an IPoIB header (=4B). For example, if the network adapter card supports a 4K MTU, the upper threshold for payload MTU is 4092B and not 4096B. A 4K MTU size also improves performance for short messages, since NDIS can coalesce a small message into a larger one.

4K MTU support is considered at beta level in the 2.1.2 release. Therefore, it is not advisable to enable both a 4K MTU and the Large Send Offload feature simultaneously (see below).

- Send and Receive checksum offload

  Possible values:

  - Disabled - No hardware checksum
  - Enabled - Try to offload if the device supports it (default)
  - Bypass - Always report success (checksum bypass)

- Large Send Offload (LSO)

  Disables/Enables the LSO feature (if supported by HW). This feature has a positive impact on overall performance.

4K MTU support is considered at beta level in the 2.1.2 release. Therefore, it is not advisable to enable both a 4K MTU and the Large Send Offload feature simultaneously.

4.3.2 Performance Tuning

To improve performance, activate the performance tuning tool as follows:

1. Start the "Device Manager" (open a command line window and enter: devmgmt.msc).

2. Open "Network Adapters".
3. Right click the relevant IPoIB adapter and select Properties.
4. Select the "Advanced" tab
5. Modify performance parameters (properties) as desired.

4.3.3 MAC Generation

IPoIB generates MAC addresses based on the GUID of the port. These MAC addresses are reported to Windows to enable normal communication. These addresses are replaced by the IPoIB driver before messages are sent on the wire, and are only for local usage. Mellanox cards are usually shipped with GUIDs of the form:

```
00-02-C9-02-00-XX-YY-ZZ or 00-02-C9-03-00-XX-YY-ZZ.
```

Since a GUID contains 8 bytes, the appropriate truncation should be done as illustrated in the following example:

- Mellanox Port GUID = "0002c90200XXYYZZ" => MAC = "0002c9XXYYZZ".
- Mellanox Port GUID = "0002c90300XXYYZZ" => MAC = "0002caXXYYZZ".

This release supports generic MAC address generation according to a user-defined bitwise GUID mask. A GUID mask is an 8-bit field that indicates which bytes of a GUID should be used in MAC address generation.

Since a MAC address has a fixed 6-byte length, the mask must contain exactly 6 non-zero bits.

- Examples of valid masks: 0xfc (binary: 1111 1100); 0x3f (binary: 0011 1111)
- Examples of invalid masks: 0xfd - contains 7 non-zero binary digits; 0x2d contains only 4 non-zero binary digits
- Example of MAC generation given a mask of 0xe7: Port GUID = "0002c90200112233" => (mask == 0xe7) => MAC = "0002c9112233".

To specify the mask, the user should change the appropriate registry value

(GUIDMask) located under

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\

{4D36E972-E325-11CE-BFC1-08002bE10318}\<IPoIB interface id>

This value is also accessible via the adapter's Properties user interface box.

IPoIB supports other companies' GUIDs such as Cisco, HP, SuperMicro,

SilverStorm and Voltaire. If the port GUID is not another company's GUID, or if it is not in one of the forms above, IPoIB will not be able to generate the correct MAC addresses from the HCA port GUID. In this case, the GUID that is generated will be an integer starting with the number 02-00-00-00-00-00.

Please use "ipconfig /all" to obtain the MAC that was reported to Windows.

If the installation was successful yet the DHCP did not assign an IP to the IPoIB interface, most likely the IPoIB driver did not recognize the port's GUID. You can run the utility 'guid2mac_checker.exe' which is available via www.mellanox.com > Products > InfiniBand SW/ Drivers > Mellanox WinOF.

The utility checks whether the port's GUID is recognized by the driver, and performs one of the following actions:

1. If the IPoIB driver recognizes the GUID, then it prints a confirmation message;
2. If the driver does not recognize the GUID but guid2mac_checker.exe recognizes it, then the utility writes an appropriate GUID mask to the registry;
3. If neither the driver nor guid2mac_checker.exe recognize the GUID, then the utility instructs the user how to create an appropriate GUID mask.

An invalid GUID mask will be rejected and IPoIB will return to its default flow.

It is not possible to change MAC address generation for known vendors like Cisco, HP, DELL etc.

To change network adapter card GUIDs, add the following flags when burning firmware:

- "-guid <GUID> -mac <mac>" for ConnectX HCA devices, and
- "-guid <GUID>" for the other (InfiniHost III family) HCA device.

See section "Updating Adapter Card Firmware" on page 11 for details.

Using the specified <GUID>, the following four parameters will be assigned:

- node GUID=<GUID>,
- port1 GUID=<GUID>+1
- port2 GUID=<GUID>+2
- system image GUID=<GUID>+3

For example, to burn firmware on a ConnectX network adapter, enter:

```
flint -d mt25418_pci_cr0 -i <image_name.bin> -guid 0002c90200123456 -mac
0002c9123457 ?burn
```

4.3.4 IGMP Configuration

Multicast traffic on IPoIB works only with IGMP v2 and not with IGMP v3 which is the default.

To configure your machine to use IGMP v2, please follow the instructions below.

- For Windows 2003 and Windows XP, run the following commands from the command line:
  - netsh routing ip igmp install
  - netsh routing ip igmp install add interface "interface name of IPoIB adapter" igmpprotocol-type=igmprtrv2

If after executing the commands above IGMP V3 remains in use, please follow the instructions on http://support.microsoft.com/default.aspx/kb/815752.

- For Windows 2008, run the following commands from the command line:
  - servermanagercmd.exe -install NPAS-RRAS-Services
  - netsh routing ip igmp install
  - netsh routing ip igmp install add interface "interface name of IPoIB adapter" igmpprotocol-type=igmprtrv2

5 SDP

SDP is a Beta code currently under development. Since it is a preliminary version of this ULP, it supports a limited set of API functions.

5.1 SDP Limitations

A limited set of API functions (w/w major flags) is supported by this version. These are: socket, connect, bind, listen, accept, send, WSASend, receive, WSARecv, select, AcceptEx, WSPShutdown and closesocket.

WSASend and WSARecv currently support all types of completion methods, including synchronous, completion routine, event and completion ports. Non-blocking IO is also supported.

Additionally:

getsockopt supports SO_PROTOCOL_INFOW and SO_CONNECT_TIME; and setsockopt supports SO_LINGER and SO_DONTLINGER WSPIoctl supports FIONBIO.

5.2 SDP Installation

SDP should be installed and activated at Mellanox WinOF VPI install time. If SDP is not installed, then please uninstall the Mellanox WinOF VPI package and reinstall it with SDP.

For further details, please see MLNX_WinOF_VPI_ReleaseNotes.txt.

5.3 Running Applications over SDP

- Run 'sc start sdp' to verify that the SDP service is running. This is needed after each reboot.
- Set the environment variable 'SdpApplications' with the name of the program to use SDP. If there is more than one program, separate the names using semi-colons.

  Examples:

```
SdpApplications=telnet.exe
SdpApplications=telnet.exe;ftp.exe
```

If this variable is not set, then only programs named SdpConnect.exe can use SDP to connect.

- Run the application using the IPoIB interface IP address.

5.4 Running an Application over SDP and Ethernet

In order to allow your program to run both SDP sockets and Ethernet sockets, perform the following:

1. Set the registry value MIXED_SDP_APPLICATIONS to 1. It is located under
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\sdp\Parameters
2. Restart the SDP driver.
3. Make sure that the SdpApplications is *NOT* set to the name of your application.
4. Your program will now use only TCP connections and not SDP. In the places that you do want to use SDP and not TCP replace the call s = socket (AF_INET_FAMILY , SOCK_STREAM, IPPROTO_TCP); with the call s = WSASocket(AF_INET_FAMILY, SOCK_STREAM, IPPROTO_TCP, NULL, 0, WSA_FLAG_OVERLAPPED | 0x40);
   and only that socket will use SDP.

5.5 Available Programs

The following applications were verified to work over SDP:

- Iometer: To obtain the program please refer to http://www.iometer.org
- iperf-2.0.1, iperf-1.7.0: These are test programs for 32-bit and 64-bit systems. To download them visit http://sourceforge.net/projects/iperf. Instructions for usage are included in the download package.
- TTcp.exe: Testing was conducted using the TTcp.exe version shipped with Windows XP SP2. Both synchronous and overlapped operations can be used.

Other TTcp.ext versions may also work.

- Ntttcp.exe: This is a benchmark developed by Microsoft. Please contact Microsoft to obtain the program.
- NetPipe: Used to measure latency. To download visit http://na-inet.jp/na/
- Microsoft CCS MPI
- SdpConnect.exe: This is a simple test program located under the SDP example directory. The program has two modes: client and server. In the server mode the program listens for connection; in the client mode the program connects to the server. The program can be used to test SDP with synchronous and overlapped operations.

  Example 1:
  - At node 1: SdpConnect.exe server 2222
  - At node 2: SdpConnect.exe client 11.4.8.63 2222 0 1 0 0 0 1 3000 16000

  Example 2:
  - At node 1: SdpConnect.exe server 2222
  - At node 2: SdpConnect.exe pingpong 11.4.8.63 2222 10000 10

For more options, enter: SdpConnect.exe

SdpConnect source code is included in the SDK component of Mellanox WinOF.

5.6 Troubleshooting

Issue # 1: How can I verify that SDP is being used?

Currently, there is no simple way to indicate SDP is being used. However, if you know that your program consumes a lot of bandwidth, then there is an indirect way to find out. Open the Task Manager and switch to the networking tab. If you see that network utilization is low, this means that SDP is being used. Alternatively, if the program is running (i.e., the two sides communicate), stop the SDP on one side (via "net stop sdp") then try to reconnect it. If it succeeds then SDP was NOT used; if it fails then SDP was used.

Issue # 2: My program does not seem to use SDP.

Suggestions:

1. Ping the remote node (ping <IP address of IPoIB interface>) to verify IPoIB is up.
2. Verify that the SDP driver is loaded (net start sdp).
3. Verify that the SdpApplications environment variable is correctly set (see Section Booting Windows from an iSCSI Target above).
4. Verify that the SDP provider is installed by running \Program Files\Mellanox\MLNX_WinOF\SDP\InstallSdpProvider.exe ?l The output of this command should include 'SDP provider'. Otherwise, install the SDP provider using <...>\InstallSdpProvider.exe ?i

Issue # 3: My system is experiencing instability and/or no network connectivity.

Suggestion: Remove the SDP provider using \Program Files\Mellanox\MLNX_WinOF\SDP\InstallSdpProvider.exe ?r then restart your computer.

Issue # 4: Interoperability with Linux SDP is broken on OFED 1.2.5, 1.3.0, and 1.3.1. A complete fix for the problem is only expected with the next OFED release. Until then, please use the following workaround:

1. Click Start->Run and enter regedit.
2. Go to:
   KEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\sdp\Parameters
3. Change the value for MaximumRecvBufferSize and MaximumSendBufferSize to 0x810. This will allow both stacks to work but with lower BW due to the small message size.

6 SRP

6.1 Overview

The SCSI RDMA Protocol (SRP) is designed to take full advantage of the protocol offload and RDMA features provided by the InfiniBand architecture. SRP allows a large body of SCSI software to be readily used on InfiniBand architecture.

The Mellanox WinOF VPI stack does not install the SRP driver by default. The installation package copies the SRP driver to the dir directory for future manual instalation if needed. To complete the SRP driver installation, an SRP target must be detected. This requires a Subnet Manager to be running in the InfiniBand subnet.

When an SRP target is detected, the "New Hardware Found" Wizard pops up.

1. Select “Browse for driver software on your computer” and click Next.
2. Click on the Have Disk button.
3. Click Browse and insert the driver’s directory path. Default path is: Program Files/Mellanox/MLNX_VPI/IB/SRP.

7 WSD

7.1 Running Applications over WSD

1. Install the WSD provider on both computers. Enter:
   \Program Files\Mellanox\MLNX_WinOF\IPoIB\installsp.exe -i
2. Check which providers are installed. Enter:
   \Program Files\Mellanox\MLNX_WinOF\IPoIB\installsp.exe -l
3. Run the application. Please note that WSD has a fall back option; thus, if the connection fails over WSD, the connection will be attempted over IPoIB.
4. Remove the WSD provider. Enter:
   \Program Files\Mellanox\MLNX_WinOF\IPoIB\installsp.exe -r

7.2 Performance

WSD has its performance counters.

1. Open perfmon and select add counters.
2. Locate the performance object called "IB winsock direct" and select "total sent bytes" or "total received bytes" -- this will display how much traffic is going through WSD (if any).

8 OpenSM

OpenSM v3.3.3 is an InfiniBand Subnet Manager. For Mellanox WinOF VPI to operate, OpenSM must be running on at least one host machine in the InfiniBand cluster.

OpenSM can either run as a Windows service which starts automatically during boot or can be started manually from the following directory: <installation_directory>\tools.

Please configure at least one machine to start the service automatically:

1. Right click on "My computer" and select Manage
2. Go to "Services and Applications" and select Services
3. Right click "OpenSM" and select Properties
4. Change "Startup type" to Automatic
5. Change service to start mode

OpenSM as a service will use the first port which is not in "down" state.

To run OpenSM manually, enter on the command line: opensm.exe

For additional run options, enter: opensm.exe -h

Notes

- For long term running, please avoid using the '-v' (verbosity) option to avoid exceeding disk quota.
- Running OpenSM on multiple servers may lead to incorrect OpenSM behavior.

  Please do not run OpenSM on more than 2 machines in the subnet.
- IBDiagnet cannot run on the same IB port that OpenSM is running on.

9 Starting and Verifying the IB Fabric

If you rebooted your machine after the installation process completed, then IB interfaces should be up.

1. Check that the IB driver is running on all nodes by using 'vstat'. The vstat utility located at <installation_directory>\tools, displays the status and capabilities of the network adaptor card(s).

   On the command line, enter “vstat” (use -h for options) to retrieve information about one or more adapter ports. The field port_state will be equal to:
   - PORT_DOWN - when there is no InfiniBand cable ("no link");
   - PORT_INITIALIZED - when the port is connected to some other port ("physical link");
   - PORT_ACTIVE - when the port is connected and OpenSM is running ("logical link").
2. Run OpenSM - see OpenSM operation instructions in the OpenSM section above.
3. Verify the status of ports by using vstat: All connected ports should report "PORT_ACTIVE" state.

10 Low level Performance Tests

The following performance tests are provided with the Mellanox WinOF VPI release under <installation_directory>\tools:

- Latency tests
  - ib_write_lat: RDMA write
  - ib_read_lat: RDMA read
  - ib_send_lat: UD, UC and RC (default) send
- Bandwidth tests
  - ib_write_bw: RDMA write
  - ib_read_bw: RDMA read
  - ib_send_bw: UD, UC and RC (default) send

For usage information, run: <test name> -h

For further information, please see Section 13.1, “Overview,” on page 67

Since the default MTU value is different per network adaptor card type, use "-m MTU" to set the MTU value on both the server and the client to the same value. This should be done only on heterogeneous systems (different network adaptor cards on different servers).

11 Driver Update and Uninstall Process

MLNX_WinOF_VPI v2.1.2 package supports driver update and installation utility. To uninstall the package, peform the following:

1. Go to Start > Control Panel > Programs and features
2. Uninstall the MLNX_VPI package

To update the driver, peform the following:

1. Rerun the new MLNX_VPI package. The driver is automatically updated.

12 InfiniBand Fabric Diagnostic Utilities

12.1 Overview

The diagnostic utilities described in this chapter provide means for debugging the connectivity and status of InfiniBand (IB) devices in a fabric. The tools are:

12.2 Utilities Usage

This section first describes common configuration, interface, and addressing for all the tools in the package. Then it provides detailed descriptions of the tools themselves including: operation, synopsis and options descriptions, error codes, and examples.

12.2.1 Common Configuration, Interface and Addressing

Topology File (Optional)

An InfiniBand fabric is composed of switches and channel adapter (HCA/TCA) devices. To identify devices in a fabric (or even in one switch system), each device is given a GUID (a MAC equivalent). Since a GUID is a non-user-friendly string of characters, it is better to alias it to a meaningful, user-given name. For this objective, the IB Diagnostic Tools can be provided with a "topology file", which is an optional configuration file specifying the IB fabric topology in user-given names.

For diagnostic tools to fully support the topology file, the user may need to provide the local system name (if the local hostname is not used in the topology file).

To specify a topology file to a diagnostic tool use one of the following two options:

1. On the command line, specify the file name using the option '`-t <topology file name>`'
2. Define the environment variable IBDIAG_TOPO_FILE

To specify the local system name to an diagnostic tool use one of the following two options:

1. On the command line, specify the system name using the option '`-s <local system name>`'
2. Define the environment variable IBDIAG_SYS_NAME

12.2.2 IB Interface Definition

The diagnostic tools installed on a machine connect to the IB fabric by means of an HCA port through which they send MADs. To specify this port to an IB diagnostic tool use one of the following options:

1. On the command line, specify the port number using the option '-p <local port number>' (see below)
2. Define the environment variable IBDIAG_PORT_NUM

In case more than one HCA device is installed on the local machine, it is necessary to specify the device's index to the tool as well. For this use on of the following options:

1. On the command line, specify the index of the local device using the following option: '-i <index of local device>'
2. Define the environment variable IBDIAG_DEV_IDX

12.2.3 Addressing

This section applies to the `ibdiagpath` tool only. A tool command may require defining the destination device or port to which it applies.

The following addressing modes can be used to define the IB ports:

- Using a Directed Route to the destination: (Tool option '-d')

  This option defines a directed route of output port numbers from the local port to the destination.

- Using port LIDs: (Tool option '-l'):

  In this mode, the source and destination ports are defined by means of their LIDs. If the fabric is configured to allow multiple LIDs per port, then using any of them is valid for defining a port.

- Using port names defined in the topology file: (Tool option '-n')

  This option refers to the source and destination ports by the names defined in the topology file. (Therefore, this option is relevant only if a topology file is specified to the tool.) In this mode, the tool uses the names to extract the port LIDs from the matched topology, then the tool operates as in the '-l' option.

12.3 ibdiagnet (of ibutils) - IB Net Diagnostic

This version of ibdiagnet is included in the ibutils package, and it is run by default after installing Mellanox OFED. To use this ibdiagnet version, run: `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 (which is defined by the -o option described below).

12.3.1 SYNOPSYS

```
ibdiagnet [-c <count>] [-v] [-r] [-o <out-dir>]
    [-t <topo-file>] [-s <sys-name>] [-i <dev-index>] [-p <port-num>]
    [-pm] [-pc] [-P <<PM counter>=<Trash Limit>>]
    [-lw <1x|4x|12x>] [-ls <2.5|5|10>]
    [-skip <dup_guids|zero_guids|pm|logical_state>]
```

12.3.2 OPTIONS

Flag	Description
-c <count>	Min number of packets to be sent across each link (default = 10
-v	Enable verbose mode
-r	Provides a report of the fabric qualities
-o <out-dir>	Specifies the directory where the output files will be placed (default = /tmp)
-t <topo-file>	Specifies the topology file name
-s <sys-name>	Specifies the local system name. Meaningful only if a topology file is specified
-i <dev-index>	Specifies the index of the device of the port used to connect to the IB fabric (in case of multiple devices on the local system)
-p <port-num>	Specifies the local device's port num used to connect to the IB fabric
-pm	Dump all the fabric links, pm Counters into ibdiagnet.pm
-pc	Reset all the fabric links pmCounters
-P <PM=<Trash>>	If any of the provided pm is greater then its provided value, print it to screen
-lw <1x|4x|12x>	Specifies the expected link width
-ls <2.5|5|10>	Specifies the expected link speed
-skip <skip-option(s)>	Skip the executions of the selected checks. Skip options (one or more can be specified): dup_guids zero_guids pm logical_state part ipoib all

12.3.3 Output Files

Table 3 - ibdiagnet (of ibutils) Output Files

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

In addition to generating the files above, the discovery phase also checks for duplicate node/port GUIDs in the IB fabric. If such an error is detected, it is displayed on the standard output. After the discovery phase is completed, directed route packets are sent multiple times (according to the -c option) to detect possible problematic paths on which packets may be lost. Such paths are explored, and a report of the suspected bad links is displayed on the standard output.

After scanning the fabric, if the `-r` option is provided, a full report of the fabric qualities is displayed. This report includes:

- SM report
- Number of nodes and systems
- Hop-count information: maximal hop-count, an example path, and a hop-count histogram
- All CA-to-CA paths traced
- Credit loop report
- mgid-mlid-HCAs multicast group and report
- Partitions report
- IPoIB report

In case the IB fabric includes only one CA, then CA-to-CA paths are not reported. Furthermore, if a topology file is provided, ibdiagnet uses the names defined in it for the output reports.

12.3.4 ERROR CODES

```
1 - Failed to fully discover the fabric
2 - Failed to parse command line options
3 - Failed to intract with IB fabric
4 - Failed to use local device or local port
5 - Failed to use Topology File
6 - Failed to load requierd Package
```

12.4 ibdiagpath - IB diagnostic path

`ibdiagpath` traces a path between two end-points and provides information regarding the nodes and ports traversed along the path. It utilizes device specific health queries for the different devices along the path.

The way `ibdiagpath` operates depends on the addressing mode used on the command line. If directed route addressing is used (-d flag), the local node is the source node and the route to the destination port is known apriori. On the other hand, if LID-route (or by-name) addressing is employed, then the source and destination ports of a route are specified by their LIDs (or by the names defined in the topology file). In this case, the actual path from the local port to the source port, and from the source port to the destination port, is defined by means of Subnet Management Linear Forwarding Table queries of the switch nodes along that path. Therefore, the path cannot be predicted as it may change.

ibdiagpath should not be supplied with contradicting local ports by the -p and -d flags (see synopsis descriptions below). In other words, when ibdiagpath is provided with the options -p and -d together, the first port in the direct route must be equal to the one specified in the “-p” option. Otherwise, an error is reported.

When ibdiagpath queries for the performance counters along the path between the source and destination ports, it always traverses the LID route, even if a directed route is specified. If along the LID route one or more links are not in the ACTIVE state, ibdiagpath reports an error.

Moreover, the tool allows omitting the source node in LID-route addressing, in which case the local port on the machine running the tool is assumed to be the source.

12.4.1 SYNOPSYS

```
ibdiagpath
    {-n <[src-name,]dst-name>|-l <[src-lid,]dst-lid>|-d <p1,p2,p3,...>}
    [-c <count>] [-v] [-o <out-dir>] [-smp]
    [-t <topo-file>] [-s <sys-name>] [-i <dev-index>] [-p <port-num>]
    [-pm] [-pc] [-P <<PM counter>=<Trash Limit>>]
    [-lw <1x|4x|12x>] [-ls <2.5|5|10>] [-sl <service level>]
```

12.4.2 OPTIONS

Flag	Description
-n <[src-name,]dst-name>	Names of the source and destination ports (as defined in the topology file; source may be omitted -> local port is assumed to be the source)
-l <[src-lid,]dst-lid>	Source and destination LIDs (source may be omitted --> the local port is assumed to be the source)
-c <count>	The minimal number of packets to be sent across each link (default = 100)
-v	Enable verbose mode
-o <out-dir>	Specifies the directory where the output files will be placed (default = /tmp)
-smp	
-t <topo-file>	Specifies the topology file name
-s <sys-name>	Specifies the local system name. Meaningful only if a topology file is specified
-i <dev-index>	Specifies the index of the device of the port used to connect to the IB fabric (in case of multiple devices on the local system)
-p <port-num>	Specifies the local device's port number used to connect to the IB fabric
-pm	Dump all the fabric links, pm Counters into ibdiagnet.pm
-pc	Reset all the fabric links pmCounters
-P <PM=<Trash>>	If any of the provided pm is greater then its provided value, print it to screen
-lw <1x|4x|12x>	Specifies the expected link width
-ls <2.5|5|10>	Specifies the expected link speed
-sl	

12.4.3 Output Files

Table 4 - ibdiagpath Output Files

Output File	Description
ibdiagpath.log	A dump of all the application reports generated according to the provided flags
ibdiagnet.pm	A dump of the Performance Counters values, of the fabric links

12.4.4 ERROR CODES

```
1 - The path traced is un-healthy
2 - Failed to parse command line options
3 - More then 64 hops are required for traversing the local port to the
    "Source" port and then to the "Destination" port
4 - Unable to traverse the LFT data from source to destination
5 - Failed to use Topology File
6 - Failed to load required Package
```

12.5 ibportstate

Enables querying the logical (link) and physical port states of an InfiniBand port. It also allows adjusting the link speed that is enabled on any InfiniBand port.

If the queried port is a *swich* port, then ibportstate can be used to

- disable, enable or reset the port
- validate the port's link width and speed against the peer port

12.5.1 Applicable Hardware

All InfiniBand devices.

12.5.2 Synopsis

```
ibportstate [-d] [-e] [-v] [-V] [-D] [-L] [-G] [-s <smlid>] \
   [-C <ca_name>] [-P <ca_port>] [-u] [-t <timeout_ms>] \
   [<dest dr_path|lid|guid>] <portnum> [<op> [<value>]]
```

12.5.3 Options

The table below lists the various flags of the command.

Table 5 - ibportstate Flags and Options

Flag	Description
-h/--help	Print the help menu
-d/--debug	Raise the IB debug level. May be used several times for higher debug levels (-ddd or -d -d -d)
-e(rr_show)	Show send and receive errors (timeouts and others)
-v(erbose)	Increase verbosity level. May be used several times for additional verbosity (-vvv or -v -v -v)
-V(ersion)	Show version info
-D(irect)	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, ...
-L/--Lid	Use Lid address argument
-G(uid)	Use GUID address argument. In most cases, it is the Port GUID. Example: '0x08f1040023'
-s <smlid>	Use <smlid> as the target lid for SM/SA queries
-C <ca_name>	Use the specified channel adapter or router
-P <ca_port>	Use the specified port
-u/--usage	Usage message

Table 5 - ibportstate Flags and Options (Continued)

Flag	Description
-t <timeout_ms>	Override the default timeout for the solicited MADs [msec]
<dest dr_path | lid | guid>	Destination's directed path, LID, or GUID.
<portnum>	Destination's port number
<op> [<value>]	Define the allowed port operations: enable, disable, reset, speed, and query

In case of multiple channel adapters (CAs) or multiple ports without a CA/port being specified, a port is chosen by the utility according to the following criteria:

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

Examples

1. Query the status of Port 1 of CA mlx4_0 (using ibstatus) and use its output (the LID – 3 in this case) to obtain additional link information using ibportstate.

```
>  ibstatus mlx4_0:1
Infiniband device 'mlx4_0' port 1 status:
       default gid:     fe80:0000:0000:0000:0000:0000:9289:3895
        base lid:        0x3
        sm lid:          0x3
        state:           2: INIT
        phys state:      5: LinkUp
        rate:            20 Gb/sec (4X DDR)

> ibportstate -C mlx4_0 3 1 query
PortInfo:
# Port info: Lid 3 port 1
LinkState:.....................Initialize
PhysLinkState:.................LinkUp
LinkWidthSupported:............1X or 4X
LinkWidthEnabled:..............1X or 4X
LinkWidthActive:...............4X
LinkSpeedSupported:............2.5 Gbps or 5.0 Gbps
LinkSpeedEnabled:..............2.5 Gbps or 5.0 Gbps
LinkSpeedActive:...............5.0 Gbps
```

2. Query the status of two channel adapters using directed paths.

```
>  ibportstate -C mlx4_0 -D 0 1
PortInfo:
# Port info: DR path slid 65535; dlid 65535; 0 port 1
LinkState:.....................Initialize
PhysLinkState:.................LinkUp
LinkWidthSupported:............1X or 4X
LinkWidthEnabled:..............1X or 4X
LinkWidthActive:...............4X
LinkSpeedSupported:............2.5 Gbps or 5.0 Gbps
LinkSpeedEnabled:..............2.5 Gbps or 5.0 Gbps
LinkSpeedActive:...............5.0 Gbps

>  ibportstate -C mthca0 -D 0 1
PortInfo:
# Port info: DR path slid 65535; dlid 65535; 0 port 1
LinkState:.....................Down
PhysLinkState:.................Polling
LinkWidthSupported:............1X or 4X
LinkWidthEnabled:..............1X or 4X
LinkWidthActive:...............4X
LinkSpeedSupported:............2.5 Gbps
LinkSpeedEnabled:..............2.5 Gbps
LinkSpeedActive:...............2.5 Gbps
```

3. Change the speed of a port.

```
# First query for current configuration
>  ibportstate -C mlx4_0 -D 0 1
PortInfo:
# Port info: DR path slid 65535; dlid 65535; 0 port 1
LinkState:.....................Initialize
PhysLinkState:.................LinkUp
LinkWidthSupported:............1X or 4X
LinkWidthEnabled:..............1X or 4X
LinkWidthActive:...............4X
LinkSpeedSupported:............2.5 Gbps or 5.0 Gbps
LinkSpeedEnabled:..............2.5 Gbps or 5.0 Gbps
LinkSpeedActive:...............5.0 Gbps

# Now change the enabled link speed
>  ibportstate -C mlx4_0 -D 0 1 speed 2
ibportstate -C mlx4_0 -D 0 1 speed 2
Initial PortInfo:
# Port info: DR path slid 65535; dlid 65535; 0 port 1
LinkSpeedEnabled:..............2.5 Gbps

After PortInfo set:
# Port info: DR path slid 65535; dlid 65535; 0 port 1
LinkSpeedEnabled:..............5.0 Gbps (IBA extension)

# Show the new configuration
>  ibportstate -C mlx4_0 -D 0 1
PortInfo:
# Port info: DR path slid 65535; dlid 65535; 0 port 1
LinkState:.....................Initialize
PhysLinkState:.................LinkUp
LinkWidthSupported:............1X or 4X
LinkWidthEnabled:..............1X or 4X
LinkWidthActive:...............4X
LinkSpeedSupported:............2.5 Gbps or 5.0 Gbps
LinkSpeedEnabled:..............5.0 Gbps (IBA extension)
LinkSpeedActive:...............5.0 Gbps
```

12.6 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 to FDBTop.

12.6.1 Applicable Hardware

InfiniBand switches.

12.6.2 Synopsis

```
ibroute [-h] [-d] [-v] [-V] [-a] [-n] [-D] [-G] [-M] [-L] [-e] [-u] [-s <smlid>] \
   [-C <ca_name>] [-P <ca_port>] [-t <timeout_ms>] \
   [<dest dr_path|lid|guid> [<startlid> [<endlid>]]]
```

12.6.3 Options

The table below lists the various flags of the command.

Table 6 - ibportstate Flags and Options

Flag	Description
-h(help)	Print the help menu
-d(ebug)	Raise the IB debug level. May be used several times for higher debug levels (-ddd or -d -d -d)
-a(ll)	Show all LIDs in range, including invalid entries
-v(erbose)	Increase verbosity level. May be used several times for additional verbosity (-vvv or -v -v -v)
-V(ersion)	Show version info
-a(ll)	Show all LIDs in range, including invalid entries
-n(o_dests)	Do not try to resolve destinations
-D(irect)	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(uid)	Use GUID address argument. In most cases, it is the Port GUID. Example: ‘0x08f1040023’
-M(ulticast)	Show multicast forwarding tables. The parameters <startlid> and <endlid> specify the MLID range.
-L/--Lid	Use Lid address argument
-u/--usage	Usage message
-e(rr_show)	Show send and receive errors (timeouts and others)
-s <smlid>	Use <smlid> as the target LID for SM/SA queries
-C <ca_name>	Use the specified channel adapter or router
-P <ca_port>	Use the specified port
-t <timeout_ms>	Override the default timeout for the solicited MADs [msec]
<dest dr_path | lid | guid>	Destination’s directed path, LID, or GUID

Table 6 - ibportstate Flags and Options

Flag	Description
<startlid>	Starting LID in an MLID range
<endlid>	Ending LID in an MLID range

Examples

1. Dump all Lids with valid out ports of the switch with Lid 2.

```
> ibroute 2
Unicast lids [0x0-0x8] of switch Lid 2 guid 0x0002c902fffff00a
(MT47396 Infiniscale-III Mellanox Technologies):
  Lid  Out   Destination
       Port     Info
0x0002 000 : (Switch portguid 0x0002c902fffff00a: 'MT47396
Infiniscale-III Mellanox Technologies')
0x0003 021 : (Switch portguid 0x000b8cffff004016: 'MT47396
Infiniscale-III Mellanox Technologies')
0x0006 007 : (Channel Adapter portguid 0x0002c90300001039:
'sw137 HCA-1')
0x0007 021 : (Channel Adapter portguid 0x0002c9020025874a:
'sw157 HCA-1')
0x0008 008 : (Channel Adapter portguid 0x0002c902002582cd:
'sw136 HCA-1')
5 valid lids dumped
```

2. Dump all Lids with valid out ports of the switch with Lid 2.

```
> ibroute 2
Unicast lids [0x0-0x8] of switch Lid 2 guid 0x0002c902fffff00a
(MT47396 Infiniscale-III Mellanox Technologies):
  Lid  Out   Destination
       Port     Info
0x0002 000 : (Switch portguid 0x0002c902fffff00a: 'MT47396
Infiniscale-III Mellanox Technologies')
0x0003 021 : (Switch portguid 0x000b8cffff004016: 'MT47396
Infiniscale-III Mellanox Technologies')
0x0006 007 : (Channel Adapter portguid 0x0002c90300001039:
'sw137 HCA-1')
0x0007 021 : (Channel Adapter portguid 0x0002c9020025874a:
'sw157 HCA-1')
0x0008 008 : (Channel Adapter portguid 0x0002c902002582cd:
'sw136 HCA-1')
5 valid lids dumped
```

3. Dump all Lids in the range 3 to 7 with valid out ports of the switch with Lid 2.

```
> ibroute 2 3 7
Unicast lids [0x3-0x7] of switch Lid 2 guid 0x0002c902fffff00a
(MT47396 Infiniscale-III Mellanox Technologies):
  Lid  Out   Destination
       Port     Info
0x0003 021 : (Switch portguid 0x000b8cffff004016: 'MT47396
Infiniscale-III Mellanox Technologies')
0x0006 007 : (Channel Adapter portguid 0x0002c90300001039:
'sw137 HCA-1')
0x0007 021 : (Channel Adapter portguid 0x0002c9020025874a:
'sw157 HCA-1')
3 valid lids dumped
```

4. Dump all Lids with valid out ports of the switch with portguid `0x000b8cffff004016`.

```
> ibroute -G 0x000b8cffff004016
Unicast lids [0x0-0x8] of switch Lid 3 guid 0x000b8cffff004016
(MT47396 Infiniscale-III Mellanox Technologies):
  Lid  Out   Destination
       Port     Info
0x0002 023 : (Switch portguid 0x0002c902fffff00a: 'MT47396
Infiniscale-III Mellanox Technologies')
0x0003 000 : (Switch portguid 0x000b8cffff004016: 'MT47396
Infiniscale-III Mellanox Technologies')
0x0006 023 : (Channel Adapter portguid 0x0002c90300001039:
'sw137 HCA-1')
0x0007 020 : (Channel Adapter portguid 0x0002c9020025874a:
'sw157 HCA-1')
0x0008 024 : (Channel Adapter portguid 0x0002c902002582cd:
'sw136 HCA-1')
5 valid lids dumped
```

5. Dump all non-empty mlids of switch with Lid 3.

```
> ibroute -M 3
Multicast mlids [0xc000-0xc3ff] of switch Lid 3 guid
0x000b8cffff004016 (MT47396 Infiniscale-III Mellanox Technolo-
gies):
            0                   1                   2
     Ports: 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4
 MLid
0xc000                                                  x
0xc001                                                  x
0xc002                                                  x
0xc003                                                  x
0xc020                  x
0xc021                  x
0xc022                  x
0xc023                  x
0xc024                  x
0xc040                  x
0xc041                  x
0xc042                  x
12 valid mlids dumped
```

12.7 smpquery

Provides a basic subset of standard SMP queries to query Subnet management attributes such as node info, node description, switch info, and port info.

12.7.1 Applicable Hardware

All InfiniBand devices.

12.7.2 Synopsys

```
smpquery [-h] [-d] [-e] [-c] [-v] [-D] [-G] [-s <smlid>] [-L] [-u] [-V]
         [-C <ca_name>] [-P <ca_port>] [-t <timeout_ms>]
         [--node-name-map <node-name-map>]
         <op> <dest dr_path|lid|guid> [op params]
```

12.7.3 Options

The table below lists the various flags of the command.

Table 7 - smpquery Flags and Options

Flag	Description
-h(help)	Print the help menu

Table 7 - smpquery Flags and Options

Flag	Description
-d(ebug)	Raise the IB debug level. May be used several times for higher debug levels (-ddd or -d -d -d)
-e(rr_show)	Show send and receive errors (timeouts and others)
-v(erbose)	Increase verbosity level. May be used several times for additional verbosity (-vvv or -v -v -v)
-D(irect)	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(uid)	Use GUID address argument. In most cases, it is the Port GUID. Example: ‘0x08f1040023’
-s <smlid>	Use <smlid> as the target LID for SM/SA queries
-V(ersion)	Show version info
-L/--Lid	Use Lid address argument
-c--combined	Use combined route address argument
-u/--usage	Usage message
-C <ca_name>	Use the specified channel adapter or router
-P <ca_port>	Use the specified port
-t <timeout_ms>	Override the default timeout for the solicited MADs [msec]
<op>	Supported operations: • NodeInfo (NI) <addr> • NodeDesc (ND) <addr> • PortInfo (PI) <addr> [<portnum>] • SwitchInfo (SI) <addr> • PKeyTable (PKeys) <addr> [<portnum>] • SL2VLTable (SL2VL) <addr> [<portnum>] • VLArbitration (VLArb) <addr> [<portnum>] • GUIDInfo (GI) <addr>
<dest dr_path | lid | guid>	Destination’s directed path, LID, or GUID

Examples

1. Query PortInfo by LID, with port modifier.

```
> smpquery portinfo 1 1
# Port info: Lid 1 port 1
Mkey:............................0x0000000000000000
GidPrefix:.......................0xfe80000000000000
Lid:.............................0x0001
SMLid:...........................0x0001
CapMask:.........................0x251086a
                                   IsSM
                                   IsTrapSupported
                                   IsAutomaticMigrationSupported
                                   IsSLMappingSupported
                                   IsSystemImageGUIDsupported
                                 IsCommunicatonManagementSupported
                                   IsVendorClassSupported
                                  IsCapabilityMaskNoticeSupported
                                   IsClientRegistrationSupported
DiagCode:........................0x0000
MkeyLeasePeriod:.................0
LocalPort:.......................1
LinkWidthEnabled:................1X or 4X
LinkWidthSupported:..............1X or 4X
LinkWidthActive:.................4X
LinkSpeedSupported:..............2.5 Gbps or 5.0 Gbps
LinkState:.......................Active
PhysLinkState:...................LinkUp
LinkDownDefState:................Polling
ProtectBits:.....................0
LMC:.............................0
LinkSpeedActive:.................5.0 Gbps
LinkSpeedEnabled:................2.5 Gbps or 5.0 Gbps
NeighborMTU:.....................2048
SMSL:............................0
VLCap:...........................VL0-7
InitType:........................0x00
VLHighLimit:.....................4
VLArbHighCap:....................8
VLArbLowCap:.....................8
InitReply:.......................0x00
MtuCap:..........................2048
VLStallCount:....................0
HoqLife:.........................31
OperVLs:.........................VL0-3
PartEnforceInb:..................0
PartEnforceOutb:.................0
```

2. Query SwitchInfo by GUID.

```
> smpquery -G switchinfo 0x000b8cffff004016
# Switch info: Lid 3
LinearFdbCap:...................49152
RandomFdbCap:...................0
McastFdbCap:....................1024
LinearFdbTop:...................8
DefPort:........................0
DefMcastPrimPort:...............0
DefMcastNotPrimPort:............0
LifeTime:.......................18
StateChange:....................0
LidsPerPort:....................0
PartEnforceCap:.................32
InboundPartEnf:.................1
OutboundPartEnf:................1
FilterRawInbound:...............1
FilterRawOutbound:..............1
EnhancedPort0:..................0
```

3. Query NodeInfo by direct route.

```
> smpquery -D nodeinfo 0
# Node info: DR path slid 65535; dlid 65535; 0
BaseVers:.......................1
ClassVers:......................1
NodeType:.......................Channel Adapter
NumPorts:.......................2
SystemGuid:.....................0x0002c9030000103b
Guid:...........................0x0002c90300001038
PortGuid:.......................0x0002c90300001039
PartCap:........................128
DevId:..........................0x634a
Revision:.......................0x000000a0
LocalPort:......................1
VendorId:.......................0x0002c9
```

12.8 perfquery

Queries InfiniBand ports' performance and error counters. Optionally, it displays aggregated counters for all ports of a node. It can also reset counters after reading them or simply reset them.

12.8.1 Applicable Hardware

All InfiniBand devices.

12.8.2 Synopsys

perfquery [-h] [-d] [-G] [-xmtsl, -X] [--xmtdisc, -D] [--rcvsl, -S] [--rcverr, -E] **[**-smplctl, -c] **[-a] [**--Lid, -L**] [**--sm_port, -s <lid>**] [**--errors, -e**] [**--verbose, -v**] [**--usage, -u**][-l] [-r] [-C <ca_name>] [-P <ca_port>] [-R][-t <timeout_ms>] [-V] [<lid|guid> [[port][reset_mask]]]**

The table below lists the various flags of the command.

Table 8 - perfquery Flags and Options

Flag	Description
-h(help)	Print the help menu
-d(ebug)	Raise the IB debug level. May be used several times for higher debug levels (-ddd or -d -d -d)
-G(uid)	Use GUID address argument. In most cases, it is the Port GUID. Example: ‘0x08f1040023’
--xmtsl, -X	Show Xmt SL port counters
--rcvsl, -S	Show Rcv SL port counters
--xmtdisc, -D	Show Xmt Discard Details
--rcverr, -E	Show Rcv Error Details
--smplctl, -c	Show samples control
-a	Apply query to all ports
--Lid, -L	Use LID address argument
--sm_port, -s <lid>	SM port lid
--errors, -e	Show send and receive errors
--verbose, -v	Increase verbosity level
--usage, -u	Usage message
-l	Loop ports
-r	Reset the counters after reading them
-C <ca_name>	Use the specified channel adapter or router
-P <ca_port>	Use the specified port
-R	Reset the counters
-t <timeout_ms>	Override the default timeout for the solicited MADs [msec]
-V(ersion)	Show version info
<lid | guid> [[port][reset_mask]]	LID or GUID

Examples

```
perfquery -r 32 1# read performance counters and reset
perfquery -e -r 32 1# read extended performance counters and reset
perfquery -R 0x20 1# reset performance counters of port 1 only
perfquery -e -R 0x20 1# reset extended performance counters of port 1 only
perfquery -R -a 32# reset performance counters of all ports
perfquery -R 32 2 0x0fff# reset only error counters of port 2
perfquery -R 32 2 0xf000# reset only non-error counters of port 2
```

1. Read local port's performance counters.

```
> perfquery
# Port counters: Lid 6 port 1
PortSelect:......................1
CounterSelect:...................0x1000
SymbolErrors:....................0
LinkRecovers:....................0
LinkDowned:......................0
RcvErrors:.......................0
RcvRemotePhysErrors:.............0
RcvSwRelayErrors:................0
XmtDiscards:.....................0
XmtConstraintErrors:.............0
RcvConstraintErrors:.............0
LinkIntegrityErrors:.............0
ExcBufOverrunErrors:.............0
VL15Dropped:.....................0
XmtData:.........................55178210
RcvData:.........................55174680
XmtPkts:.........................766366
RcvPkts:.........................766315
```

2. Read performance counters from LID 2, all ports.

```
> smpquery -a 2
# Port counters: Lid 2 port 255
PortSelect:......................255
CounterSelect:...................0x0100
SymbolErrors:....................65535
LinkRecovers:....................255
LinkDowned:......................16
RcvErrors:.......................657
RcvRemotePhysErrors:.............0
RcvSwRelayErrors:................70
XmtDiscards:.....................488
XmtConstraintErrors:.............0
RcvConstraintErrors:.............0
LinkIntegrityErrors:.............0
ExcBufOverrunErrors:.............0
VL15Dropped:.....................0
XmtData:.........................129840354
RcvData:.........................129529906
XmtPkts:.........................1803332
RcvPkts:.........................1799018
```

3. Read then reset performance counters from LID 2, port 1.

```
> perfquery  -r 2 1
# Port counters: Lid 2 port 1
PortSelect:......................1
CounterSelect:...................0x0100
SymbolErrors:....................0
LinkRecovers:....................0
LinkDowned:......................0
RcvErrors:.......................0
RcvRemotePhysErrors:.............0
RcvSwRelayErrors:................0
XmtDiscards:.....................3
XmtConstraintErrors:.............0
RcvConstraintErrors:.............0
LinkIntegrityErrors:.............0
ExcBufOverrunErrors:.............0
VL15Dropped:.....................0
XmtData:.........................0
RcvData:.........................0
XmtPkts:.........................0
RcvPkts:.........................0
```

12.9 ibping

ibping uses vendor mads to validate connectivity between IB nodes. On exit, (IP) ping like output is shown. ibping is run as a client/server, however the default is to run it as a client. Note also that a default ping server is implemented within the kernel.

12.9.1 Synopsys

```
ibping [-d(ebug)] [-e(rr_show)] [-v(erbose)] [-G(uid)] [-C ca_name] [-P ca_port]
[-s smlid] [-t(imeout)timeout_ms] [-V(ersion)] [-L(id)][-u(sage)] [-c ping_count]
[-f(lood)] [-o oui] [-S(erver)] [-h(elp)] <dest lid | guid>
```

12.9.2 Options

The table below lists the various flags of the command.

Table 9 - ibping Flags and Options

Flag	Description
-c	Stops after count packets
-f, (--flood)	Floods destination: send packets back to back without delay
-o, (--oui)	Uses specified OUI number to multiplex vendor mads
-S, (--Serve)r	Starts in server mode (do not return)
-d/-ddd/-d -d -d	Raises the IB debugging level
-e	Shows send and receive errors (timeouts and others)
-h	Shows the usage message
-v/-vvv/-v -v -v	Increases the application verbosity level
-V	Shows the version info
--Lid, -L	Use LID address argument
--usage, -u	Usage message
-G	Uses GUID address argument. In most cases, it is the Port GUID. For example: "0x08f1040023"
-s <smlid>	Uses 'smlid' as the target lid for SM/SA queries
-C <ca_name>	Uses the specified ca_name
-P <ca_port>	Uses the specified ca_port
-t <timeout_ms>	Overrides the default timeout for the solicited mads

12.10 ibnetdiscover

ibnetdiscover performs IB subnet discovery and outputs a human readable topology file. GUIDs, node types, and port numbers are displayed as well as port LIDs and NodeDescriptions. All nodes (and links) are dis-played (full topology). Optionally, this utility can be used to list the cur-

rent connected nodes by node-type. The output is printed to standard output unless a topology file is specified.

12.10.1Synopsys

```
ibnetdiscover [-d(ebug)] [-e(rr_show)] [-v(erbose)] [-s(how)] [-l(ist)] [-
g(rouping)] [-H(ca_list)][-S(witch_list)] [-R(outer_list)] [-C ca_name] [-P
ca_port] [-t(imeout) timeout_ms] [-V(ersion)] [--outstanding_smps -o <val>] [-
u(sage)] [--node-name-map <node-name-map>] [--cache <filename>] [--load-cache
<filename>] [-p(orts)] [-m(ax_hops)]
[-h(elp)] [<topology-file>]
```

12.10.2Options

The table below lists the various flags of the command.

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

Table 10 - ibnetdiscover Flags and Options

Flag	Description
-l, --list	List of connected nodes
-g, --grouping	Show grouping. Grouping correlates IB nodes by different vendor specific schemes. It may also show the switch external ports correspondence.
-H, --Hca_list	List of connected CAs
-S, --Switch_list	List of connected switches
-R, --Router_list	List of connected routers
-s, --show	Show progress information during discovery
--node-name-map <node-name-map>	Specify a node name map. The node name map file maps GUIDs to more user friendly names. See "Topology File Format" on page 56.
--cache <filename>	Cache the ibnetdiscover network data in the specified filename. This cache may be used by other tools for later analysis
--load-cache <filename>	Load and use the cached ibnetdiscover data stored in the specified filename. May be useful for outputting and learning about other fabrics or a previous state of a fabric
--diff <filename>	Load cached ibnetdiscover data and do a diff comparison to the current network or another cache. A special diff output for ibnetdiscover output will be displayed showing differences between the old and current fabric. By default, the following are compared for differences: switches, channel adapters, routers, and port connections
--diffcheck <key(s)>	Specify what diff checks should be done in the --diff option above. Comma separate multiple diff check key(s). The available diff checks are: sw = switches, ca = channel adapters, router = routers, port = port connections, lid = lids, nodedesc = node descriptions. Note that port, lid, and nodedesc are checked only for the node types that are specified (e.g. sw, ca, router). If port is specified alongside lid or nodedesc, remote port lids and node descriptions will also be com-pared

Table 10 - ibnetdiscover Flags and Options

Flag	Description
-p, --ports	Obtain a ports report which is a list of connected ports with relevant information (like LID, port-num, GUID, width, speed, and NodeDescription)
-m, --max_hops	Report max hops discovered
-d/-ddd/-d -d -d	Raise the IB debugging level
-e	Show send and receive errors (timeouts and others)
-h	Show the usage message
-v/-vv/-v -v -v	Increase the application verbosity level
-V	Show the version info
--outstanding_smps -o <val>	Specify the number of outstanding SMPs which should be issued during the scan
-u (sage)	Usage message
-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

12.10.3 Topology File Format

The topology file format is largely intuitive. Most identifiers are given textual names like vendor ID (vendid), device ID (device ID), GUIDs of various types (sysimgguid, caguid, switchguid, etc.). PortGUIDs are shown in parentheses (). For switches, this is shown on the switchguid line. For CA and router ports, it is shown on the connectivity lines. The IB node is identified followed by the number of ports and a quoted the node GUID. On the right of this line is a comment (#) followed by the NodeDescription in quotes. If the node is a switch, this line also contains whether switch port 0 is base or enhanced, and the LID and LMC of port 0. Subsequent lines pertaining to this node show the connectivity. On the left is the port number of the current node. On the right is the peer node (node at other end of link). It is identified in quotes with nodetype followed by - followed by NodeGUID with the port number in square brackets. Further on the right is a comment (#). What follows the comment is dependent on the node type. If it it a switch node, it is followed by the NodeDescription in quotes and the LID of the peer node. If it is a CA or router node, it is followed by the local LID and LMC and then followed by the NodeDescription in quotes and the LID of the peer node. The active link width and speed are then appended to the end of this output line.

Example

```
# Topology file: generated on Tue Jun  5 14:15:10 2007
#
# Max of 3 hops discovered
# Initiated from node 0008f10403960558 port 0008f10403960559
```

12.10.3.1 Non-Chassis Nodes

```
vendid=0x8f1
```

```
devid=0x5a06
sysimgguid=0x5442ba00003000
switchguid=0x5442ba00003080(5442ba00003080)
Switch  24 "S-005442ba00003080"        # "ISR9024 Voltaire" base port 0 lid 6 lmc 0
[22]    "H-0008f10403961354"[1](8f10403961355)        # "MT23108 InfiniHost Mella-
nox Technologies" lid 4 4xSDR
[10]    "S-0008f10400410015"[1]         # "SW-6IB4 Voltaire" lid 3 4xSDR
[8]     "H-0008f10403960558"[2](8f1040396055a)        # "MT23108 InfiniHost Mella-
nox Technologies" lid 14 4xSDR
[6]     "S-0008f10400410015"[3]         # "SW-6IB4 Voltaire" lid 3 4xSDR
[12]    "H-0008f10403960558"[1](8f10403960559)        # "MT23108 InfiniHost Mella-
nox Technologies" lid 10 4xSDR
vendid=0x8f1
devid=0x5a05
switchguid=0x8f10400410015(8f10400410015)
Switch  8 "S-0008f10400410015"         # "SW-6IB4 Voltaire" base port 0 lid 3 lmc 0
[6]     "H-0008f10403960984"[1](8f10403960985)        # "MT23108 InfiniHost Mella-
nox Technologies" lid 16 4xSDR
[4]     "H-005442b100004900"[1](5442b100004901)       # "MT23108 InfiniHost Mella-
nox Technologies" lid 12 4xSDR
[1]     "S-005442ba00003080"[10]               # "ISR9024 Voltaire" lid 6 1xSDR
[3]     "S-005442ba00003080"[6]         # "ISR9024 Voltaire" lid 6 4xSDR
vendid=0x2c9
devid=0x5a44
caguid=0x8f10403960984
Ca      2 "H-0008f10403960984"         # "MT23108 InfiniHost Mellanox Technologies"
[1](8f10403960985)     "S-0008f10400410015"[6]         # lid 16 lmc 1 "SW-6IB4 Vol-
taire" lid 3 4xSDR
vendid=0x2c9
devid=0x5a44
caguid=0x5442b100004900
Ca      2 "H-005442b100004900"         # "MT23108 InfiniHost Mellanox Technologies"
[1](5442b100004901)     "S-0008f10400410015"[4]        # lid 12 lmc 1 "SW-6IB4 Vol-
taire" lid 3 4xSDR
vendid=0x2c9
devid=0x5a44
caguid=0x8f10403961354
Ca      2 "H-0008f10403961354"         # "MT23108 InfiniHost Mellanox Technologies"
[1](8f10403961355)     "S-005442ba00003080"[22]               # lid 4 lmc 1
"ISR9024 Voltaire" lid 6 4xSDR
vendid=0x2c9
devid=0x5a44
caguid=0x8f10403960558
Ca      2 "H-0008f10403960558"         # "MT23108 InfiniHost Mellanox Technologies"
[2](8f1040396055a)     "S-005442ba00003080"[8]         # lid 14 lmc 1 "ISR9024 Vol-
taire" lid 6 4xSDR
```

```
[1](8f10403960559)     "S-005442ba00003080"[12]                # lid 10 lmc 1
"ISR9024 Voltaire" lid 6 1xSDR
```

When grouping is used, IB nodes are organized into chasses which are numbered. Nodes which cannot be determined to be in a chassis are displayed as "Non-Chassis Nodes". External ports are also shown on the connectivity lines.

12.10.3.2 Node Name Map File Format

The node name map is used to specify user friendly names for nodes in the output. GUIDs are used to perform the lookup.

```
# comment
<guid> "<name>"
```

Example

```
# IB1
# Line cards
0x0008f104003f125c "IB1 (Rack 11 slot 1   ) ISR9288/ISR9096 Voltaire sLB-24D"
0x0008f104003f125d "IB1 (Rack 11 slot 1   ) ISR9288/ISR9096 Voltaire sLB-24D"
0x0008f104003f10d2 "IB1 (Rack 11 slot 2   ) ISR9288/ISR9096 Voltaire sLB-24D"
0x0008f104003f10d3 "IB1 (Rack 11 slot 2   ) ISR9288/ISR9096 Voltaire sLB-24D"
0x0008f104003f10bf "IB1 (Rack 11 slot 12  ) ISR9288/ISR9096 Voltaire sLB-24D"
# Spines
0x0008f10400400e2d "IB1 (Rack 11 spine 1   ) ISR9288 Voltaire sFB-12D"
0x0008f10400400e2e "IB1 (Rack 11 spine 1   ) ISR9288 Voltaire sFB-12D"
0x0008f10400400e2f "IB1 (Rack 11 spine 1   ) ISR9288 Voltaire sFB-12D"
0x0008f10400400e31 "IB1 (Rack 11 spine 2   ) ISR9288 Voltaire sFB-12D"
0x0008f10400400e32 "IB1 (Rack 11 spine 2   ) ISR9288 Voltaire sFB-12D"
# GUID   Node Name
0x0008f10400411a08 "SW1  (Rack  3) ISR9024 Voltaire 9024D"
0x0008f10400411a28 "SW2  (Rack  3) ISR9024 Voltaire 9024D"
0x0008f10400411a34 "SW3  (Rack  3) ISR9024 Voltaire 9024D"
0x0008f104004119d0 "SW4  (Rack  3) ISR9024 Voltaire 9024D"
```

12.11 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.

12.11.1 Synopsys

```
ibtracert [-d(ebug)] [-v(erbose)] [-D(irect)] [-L(id)] [-e(rrors)] [-u(sage)] [-
G(uids)] [-f(orce)] [-n(o_info)] [-m mlid] [-s smlid] [-C ca_name][-P ca_port] [-
t(imeout) timeout_ms] [-V(ersion)] [--node-name--map <node-name-map>] [-h(elp)]
[<dest dr_path|lid|guid> [<startlid> [<endlid>]]
```

12.11.2Options

The table below lists the various flags of the command.

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

Table 11 - ibtracert Flags and Options

Flag	Description
-f (orce)	Force
-n, --no_info	Simple format; do not show additional information
-m	Show the multicast trace of the specified mlid
--node-name-map <node-name-map>	Specify a node name map. The node name map file maps GUIDs to more user friendly names. See "Topology File Format" on page 56.
-d/-ddd/-d -d -d	Raise the IB debugging level
-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, ...
--Lid, -L	Use LID address argument
--errors, -e	Show send and receive errors
--usage, -u	Usage message
-G	Use GUID address argument. In most cases, it is the Port GUID. Example: "0x08f1040023"
-s <smlid>	Use 'smlid' as the target lid for SM/SA queries
-h	Show the usage message
-v/-vv/-v -v -v	Increase the application verbosity level
-V	Show the version info
-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

Examples

- Unicast examples

```
ibtracert 4 16              # show path between lids 4 and 16
ibtracert -n 4 16           # same, but using simple output format
ibtracert -G 0x8f1040396522d 0x002c9000100d051  # use guid addresses
```

- Multicast example

```
ibtracert -m 0xc000 4 16    # show multicast path of mlid 0xc000 between lids 4 and
16
```

12.12 sminfo

Optionally sets and displays the output of a sminfo query in a 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.

Using sminfo for any purposes other then simple query may be very dangerous, and may result in a malfunction of the target SM.

12.12.1 Synopsys

```
sminfo [-d(ebug)] [-e(rr_show)] [-s state] [-p prio] [-a activity] [-D(irect)]
[-L(id)] [-u(sage)] [-G(uid)] [-C ca_name] [-P ca_port] [-t(imeout) timeout_ms] [-
V(ersion)] [-h(elp)] sm_lid | sm_dr_path [modifier]
```

12.12.2 Options

The table below lists the various flags of the command.

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

Table 12 - sminfo Flags and Options

Flag	Description
-s	Set SM state • 0 - not active • 1 - discovering • 2 - standby • 3 - master
-p	Set priority (0-15)
-a	Set activity count
-d/-ddd/-d -d -d	Raise the IB debugging level
-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, ...
--Lid, -L	Use LID address argument
--usage, -u	Usage message
-e	Show send and receive errors (timeouts and others)

Table 12 - sminfo Flags and Options

Flag	Description
-G	Use GUID address argument. In most cases, it is the Port GUID. Example: "0x08f1040023"
-s <smlid>	Use 'smlid' as the target lid for SM/SA queries
-h	Show the usage message
-v/-vv/-v -v -v	Increase the application verbosity level
-V	Show the version info
-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

Examples

```
sminfo                  # local ports sminfo
sminfo 32               # show sminfo of lid 32
sminfo  -G 0x8f1040023  # same but using guid address
```

12.13 ibclearerrors

ibclearerrors is a script which clears the PMA error counters in PortCounters by either walking the IB subnet topology or using an already saved topology file.

12.13.1 Synopsys

```
ibclearerrors [-h] [-N | -nocolor] [<topology-file> | -C ca_name -P ca_port -t(ime-
out) timeout_ms]
```

12.13.2 Options

The table below lists the various flags of the command.

Table 13 - ibclearerrors Flags and Options

Flag	Description
-N | -nocolor	Use mono rather than color mode
-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

12.14 ibstat

ibstat is a binary which displays basic information obtained from the local IB driver. Output includes LID, SMLID, port state, link width active, and port physical state.

12.14.1 Synopsys

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

12.14.2 Options

The table below lists the various flags of the command.

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

Table 14 - ibstat Flags and Options

Flag	Description
-l, --list_of_cas	List all IB devices
-s, --short	Short output
-p, --port_list	Show port list
ca_name	InfiniBand device name
portnum	Port number of InfiniBand device
-d/-ddd/-d -d -d	Raise the IB debugging level
-h	Show the usage message
-v/-vv/-v -v -v	Increase the application verbosity level
-V	Show the version info

Examples

```
ibstat            # display status of all ports on all IB devices
ibstat -l         # list all IB devices
ibstat -p         # show port guids
ibstat mthca0 2   # show status of port 2 of 'mthca0'
```

12.15 vstat

vstat is a binary which displays information on the HCA attributes.

12.15.1 Synopsys

```
vstat [-v] [-c]
```

12.15.2 Options

The table below lists the various flags of the command..

Table 15 - ibstat Flags and Options

Flag	Description
-v -	Verbose mode
-c	HCA error/statistic counters

12.16 part_man

part_man is an application which allows creating, deleting and viewing existing host partitions.

12.16.1 Synopsys

```
part_man.exe <show|add|rem> <port_guid> <pkey1 pkey2 ...>
```

12.16.2 Options

The table below lists the various flags of the command..

Table 16 - ibstat Flags and Options

Flag	Description
show	Shows the existing partitions. The output format is: port_guid1 pkey1 pkey2 pkey3 pkey4 pkey5 pkey6 pkey7 pkey8 where *port_guid* is a port guid in hexadecimal format, and pkeys are the values of the partition key (in hex format) of this port. The default partition key (0xFFFF) is not shown and cannot be created by the part_man.exe.
add	Creates new partition(s) on the specified port. The output format is: port_guid add <port_guid> <pkey1> <pkey2>
rem	Removes partition key of the specified port. The output format is: part_man.exe rem <port_guid> <pkey1> <pkey2>

12.17 osmtest

osmtest is a test program to validate InfiniBand subnet manager and administration (SM/SA). Default is to run all flows with the exception of the QoS flow. osmtest provides a test suite for opensm. osmtest has the following capabilities and testing flows:

- It creates an inventory file of all available Nodes, Ports, and PathRecords, including all their fields.
- It verifies the existing inventory, with all the object fields, and matches it to a presaved one.
- A Multicast Compliancy test.
- An Event Forwarding test.

- A Service Record registration test.
- An RMPP stress test.
- A Small SA Queries stress test.

It is recommended that after installing opensm, the user should run "osmtest -f c" to generate the inventory file, and immediately afterwards run "osmtest -f a" to test OpenSM.

Additionally, it is recommended to create the inventory when the IB fabric is stable, and occasionally run "osmtest -v" to verify that nothing has changed.

12.17.1Synopsys

```
osmtest [-f(low) <c|a|v|s|e|f|m|q|t>] [-w(ait) <trap_wait_time>] [-d(ebug) <number>] [-m(ax_lid) <LID in hex>] [-g(uid)[=]<GUID in hex>] [-p(ort)] [-i(nventory) <filename>] [-s(tress)] [-M(ulticast_Mode)] [-t(imeout) <milliseconds>] [-l | --log_file] [-v] [-vf <flags>] [-h(elp)]
```

12.17.2Options

The table below lists the various flags of the command.

Table 17 - osmtest Flags and Options

Flag	Description
-f, --flow	This option directs osmtest to run a specific flow. The following is the flow's description: • c = create an inventory file with all nodes, ports and paths • a = run all validation tests (expecting an input inventory) • v = only validate the given inventory file • s = run service registration, deregistration, and lease test • e = run event forwarding test • f = flood the SA with queries according to the stress mode • m = multicast flow • q = QoS info: dump VLArb and SLtoVL tables • t = run trap 64/65 flow (this flow requires running of external tool, default is all flows except QoS)
-w, --wait	This option specifies the wait time for trap 64/65 in seconds It is used only when running -f t - the trap 64/65 flow (default to 10 sec)
-d, --debug	This option specifies a debug option. These options are not normally needed. The number following -d selects the debug option to enable as follows: OPT Description --- ---------- -d0 - Ignore other SM nodes -d1 - Force single threaded dispatching -d2 - Force log flushing after each log message -d3 - Disable multicast support
-m, --max_lid	This option specifies the maximal LID number to be searched for during inventory file build (default to 100)

Table 17 - osmtest Flags and Options

Flag	Description
-g, --guid	This option specifies the local port GUID value with which OpenSM should bind. OpenSM may be bound to 1 port at a time. If GUID given is 0, OpenSM displays a list of possible port GUIDs and waits for user input. Without -g, OpenSM trys to use the default port
-p, --port	This option displays a menu of possible local port GUID values with which osmtest could bind
-i, --inventory	This option specifies the name of the inventory file Normally, osmtest expects to find an inventory file, which osmtest uses to validate real-time information received from the SA during testing If -i is not specified, osmtest defaults to the file osmtest.dat See -c option for related information
-s, --stress	This option runs the specified stress test instead of the normal test suite Stress test options are as follows: OPT Description --- ---------- -s1 - Single-MAD (RMPP) response SA queries -s2 - Multi-MAD (RMPP) response SA queries -s3 - Multi-MAD (RMPP) Path Record SA queries -s4 - Single-MAD (non RMPP) get Path Record SA queries Without -s, stress testing is not performed
-M, --Multicast_Mode	This option specify length of Multicast test: OPT Description --- ---------- -M1 - Short Multicast Flow (default) - single mode -M2 - Short Multicast Flow - multiple mode -M3 - Long Multicast Flow - single mode -M4 - Long Multicast Flow - multiple mode • Single mode - Osmtest is tested alone, with no other apps that interact with OpenSM MC • Multiple mode - Could be run with other apps using MC with OpenSM. Without -M, default flow testing is performed
-t, --timeout	This option specifies the time in milliseconds used for transaction timeouts. Specifying -t 0 disables timeouts. Without -t, OpenSM defaults to a timeout value of 200 milliseconds.
-l, --log_file	This option defines the log to be the given file. By default the log goes to std-out.
-v, --verbose	This option increases the log verbosity level. The -v option may be specified multiple times to further increase the verbosity level. See the -vf option for more information about. log verbosity.
-V	This option sets the maximum verbosity level and forces log flushing. The -V is equivalent to '-vf0xFF -d 2'. See the -vf option for more information about. log verbosity.

Table 17 - osmtest Flags and Options

Flag	Description
-vf	This option sets the log verbosity level. A flags field must follow the -D option. A bit set/clear in the flags enables/disables a specific log level as follows: BIT LOG LEVEL ENABLED ---- ---------- 0x01 - ERROR (error messages) 0x02 - INFO (basic messages, low volume) 0x04 - VERBOSE (interesting stuff, moderate volume) 0x08 - DEBUG (diagnostic, high volume) 0x10 - FUNCS (function entry/exit, very high volume) 0x20 - FRAMES (dumps all SMP and GMP frames) 0x40 - ROUTING (dump FDB routing information) 0x80 - currently unused. Without -vf, osmtest defaults to ERROR + INFO (0x3) Specifying -vf 0 disables all messages Specifying -vf 0xFF enables all messages (see -V) High verbosity levels may require increasing the transaction timeout with the -t option
-h, --help	Display this usage info then exit.

13 InfiniBand Fabric Performance Utilities

13.1 Overview

The performance utilities described in this chapter are intended to be used as a performance micro-benchmark. The tools are:

13.2 ib_read_bw

ib_read_bw calculats the BW of RDMA read between a pair of machines. One acts as a server and the other as a client. The client RDMA reads the server memory and calculate the BW by sampling the CPU each time it receive a successfull completion. The test supports features such as Bidirectional, in which they both RDMA read from each other memory's at the same time,change of mtu size, tx size, number of iteration, message size and more. Read is availible only in RC connection mode (as specified in IB spec).

13.2.1 Synopsys

```
ib_read_bw [-i(b_port) ib_port] [-m(tu) mtu_size] [-s(ize) message_size] [-n
iteration_num] [-p(ort) PDT_port] [-b(idirectional)] [-o(uts) outstanding reads]
[-a(ll)] [-V(ersion)]
```

13.2.2 Options

The table below lists the various flags of the command.

Table 18 - ib_read_bw Flags and Options

Flag	Description
-p, --port=<port>	Listens on/connect to port <port> (default 18515)
-d, --ib-dev=<dev>	Uses IB device <device guid> (default first device found)
-i, --ib-port=<port>	Uses port <port> of IB device (default 1)
-m, --mtu=<mtu>	The mtu size (default 1024)

Table 18 - ib_read_bw Flags and Options

Flag	Description
-o, --outs=<num>	The number of outstanding read/atom(default 4)
-s, --size=<size>	The size of message to exchange (default 65536)
-a, --all	Runs sizes from 2 till 2^23
-t, --tx-depth=<dep>	The size of tx queue (default 100)
-n, --iters=<iters>	The number of exchanges (at least 2, default 1000)
-b, --bidirectional	Measures bidirectional bandwidth (default unidirectional)
-V, --version	Displays version number

13.3 ib_read_lat

ib_read_lat calculats the latency of RDMA read operation of message_sizeB between a pair of machines. One acts as a server and the other as a client. They perform a ping pong benchmark on which one side RDMA reads the memory of the other side only after the other side have read his memory. Each of the sides samples the CPU clock each time they read the other side memory , in order to calculate latency. Read is availible only in RC connection mode (as specified in IB spec).

13.3.1 Synopsys

```
ib_read_lat [-i(b_port) ib_port] [-m(tu) mtu_size] [-s(ize) message_size] [-t(x-
depth) tx_size] [-n iteration_num] [-p(ort) PDT_port] [-o(uts) outstanding reads]
[-a(ll)] [-V(ersion)] [-C report cycles] [-H report histogram] [-U report unsorted]
```

13.3.2 Options

The table below lists the various flags of the command.

Table 19 - ib_read_lat Flags and Options

Flag	Description
-p, --port=<port>	Listens on/connect to port <port> (default 18515)
-d, --ib-dev=<dev>	Uses IB device <device guid> (default first device found)
-i, --ib-port=<port>	Uses port <port> of IB device (default 1)
-m, --mtu=<mtu>	The mtu size (default 1024)
-o, --outs=<num>	The number of outstanding read/atom(default 4)
-s, --size=<size>	The size of message to exchange (default 65536)
-a, --all	Runs sizes from 2 till 2^23
-t, --tx-depth=<dep>	The size of tx queue (default 100)
-n, --iters=<iters>	The number of exchanges (at least 2, default 1000)
-C, --report-cycles	Reports times in cpu cycle units (default microseconds)

Table 19 - ib_read_lat Flags and Options

Flag	Description
-H, --report-histogram	Print out all results (default print summary only)
-U, --report-unsorted (implies -H)	Print out unsorted results (default sorted)
-V, --version	Displays version number

13.4 ib_send_bw

ib_send_bw calculats the BW of SEND between a pair of machines. One acts as a server and the other as a client. The server receive packets from the client and they both calculate the throughput of the operation. The test supports features such as Bidirectional, on which they both send and receive at the same time, change of mtu size, tx size, number of iteration, message size and more. Using the "-a" provides results for all message sizes.

13.4.1 Synopsys

```
ib_send_bw [-i(b_port) ib_port]  [-c(onnection_type) RC\UC\UD] [-m(tu) mtu_size]
[-s(ize) message_size] [-t(x-depth) tx_size] [-n iteration_num] [-p(ort) PDT_port]
[-b(idirectional)] [-a(ll)] [-V(ersion)]
```

13.4.2 Options

The table below lists the various flags of the command.

Table 20 - ib_send_bw Flags and Options

Flag	Description
-p, --port=<port>	Listens on/connect to port <port> (default 18515)
-d, --ib-dev=<dev>	Uses IB device <device guid> (default first device found)
-i, --ib-port=<port>	Uses port <port> of IB device (default 1)
-m, --mtu=<mtu>	The mtu size (default 1024)
-c, --connection=<RC/UC>	Connection type RC/UC/UD (default RC)
-s, --size=<size>	The size of message to exchange (default 65536)
-a, --all	Runs sizes from 2 till 2^23
-t, --tx-depth=<dep>	The size of tx queue (default 100)
-n, --iters=<iters>	The number of exchanges (at least 2, default 1000)
-b, --bidirectional	Measures bidirectional bandwidth (default unidirectional)
-V, --version	Displays version number

13.5 ib_send_lat

ib_send_lat calculats the latency of sending a packet in message_sizeB between a pair of machines. One acts as a server and the other as a client. They perform a ping pong benchmark on which you send packet only if you receive one. Each of the sides samples the CPU each time they receive a packet in order to calculate the latency.

13.5.1 Synopsys

```
ib_send_lat [-i(b_port) ib_port]  [-c(onnection_type) RC\UC\UD] [-m(tu) mtu_size]
[-s(ize) message_size] [-t(x-depth) tx_size] [-n iteration_num] [-p(ort) PDT_port]
[-a(ll)] [-V(ersion)] [-C report cycles] [-H report histogram] [-U report unsorted]
```

13.5.2 Options

The table below lists the various flags of the command.

Table 21 - ib_send_lat Flags and Options

Flag	Description
-p, --port=<port>	Listens on/connect to port <port> (default 18515)
-d, --ib-dev=<dev>	Uses IB device <device guid> (default first device found)
-i, --ib-port=<port>	Uses port <port> of IB device (default 1)
-m, --mtu=<mtu>	The mtu size (default 1024)
-c, --connection=<RC/UC>	Connection type RC/UC/UD (default RC)
-s, --size=<size>	The size of message to exchange (default 65536)
-l, --signal	Signal completion on each msg
-a, --all	Runs sizes from 2 till 2^23
-t, --tx-depth=<dep>	The size of tx queue (default 100)
-n, --iters=<iters>	The number of exchanges (at least 2, default 1000)
-C, --report-cycles	Reports times in cpu cycle units (default microseconds)
-H, --report-histogram	Print out all results (default print summary only)
-U, --report-unsorted (implies -H)	Print out unsorted results (default sorted)
-V, --version	Displays version number

13.6 ib_write_bw

ib_write_bw calculats the BW of RDMA write between a pair of machines. One acts as a server and the other as a client. The client RDMA writes to the server memory and calculate the BW by sampling the CPU each time it receive a successfull completion. The test supports features such as Bidirectional, in which they both RDMA write to each other at the same time, change of mtu size, tx size, number of iteration, message size and more. Using the "-a" flag provides results for all message sizes.

13.6.1 Synopsys

```
ib_write_bw [-q num of qps] [-c(onnection_type) RC\UC\UD] [-i(b_port) ib_port] [-
m(tu) mtu_size] [-s(ize) message_size] [-t(x-depth) tx_size] [-n iteration_num] [-
p(ort) PDT_port] [-b(idirectional)] [-a(ll)] [-V(ersion)]
```

13.6.2 Options

The table below lists the various flags of the command.

Table 22 - ib_write_bw Flags and Options

Flag	Description
-p, --port=<port>	Listens on/connect to port <port> (default 18515)
-d, --ib-dev=<dev>	Uses IB device <device guid> (default first device found)
-i, --ib-port=<port>	Uses port <port> of IB device (default 1)
-m, --mtu=<mtu>	The mtu size (default 1024)
-c, --connection=<RC/UC>	Connection type RC/UC/UD (default RC)
-s, --size=<size>	The size of message to exchange (default 65536)
-a, --all	Runs sizes from 2 till 2^23
-t, --tx-depth=<dep>	The size of tx queue (default 100)
-n, --iters=<iters>	The number of exchanges (at least 2, default 1000)
-b, --bidirectional	Measures bidirectional bandwidth (default unidirectional)
-V, --version	Displays version number
-g, --post=<num of posts>	The number of posts for each qp in the chain (default tx_depth)
-q, --qp=<num of qp's>	The number of qp's(default 1)

13.7 ib_write_lat

ib_write_lat calculats the latency of RDMA write operation of message_sizeB between a pair of machines. One acts as a server and the other as a client. They perform a ping pong benchmark on which one side RDMA writes to the other side memory only after the other side wrote on his memory. Each of the sides samples the CPU clock each time they write to the other side memory, in order to calculate latency.

13.7.1 Synopsys

```
ib_write_lat [-i(b_port) ib_port]  [-c(onnection_type) RC\UC\UD] [-m(tu) mtu_size]
[-s(ize) message_size] [-t(x-depth) tx_size] [-n iteration_num] [-p(ort) PDT_port]
[-a(ll)] [-V(ersion)] [-C report cycles] [-H report histogram] [-U report unsorted]
```

13.7.2 Options

The table below lists the various flags of the command.

Table 23 - ib_write_lat Flags and Options

Flag	Description
-p, --port=<port>	Listens on/connect to port <port> (default 18515)
-d, --ib-dev=<dev>	Uses IB device <device guid> (default first device found)
-i, --ib-port=<port>	Uses port <port> of IB device (default 1)
-m, --mtu=<mtu>	The mtu size (default 1024)
-c, --connection=<RC/UC>	Connection type RC/UC/UD (default RC)
-s, --size=<size>	The size of message to exchange (default 65536)
-f, --freq=<dep>	How often the time stamp is taken
-a, --all	Runs sizes from 2 till 2^23
-t, --tx-depth=<dep>	The size of tx queue (default 100)
-n, --iters=<iters>	The number of exchanges (at least 2, default 1000)
-C, --report-cycles	Reports times in cpu cycle units (default microseconds)
-H, --report-histogram	Print out all results (default print summary only)
-U, --report-unsorted (implies -H)	Print out unsorted results (default sorted)
-V, --version	Displays version number

13.8 ibv_read_bw

This is a more advanced version of ib_read_bw and contains more flags and featurs than the older version and also improved algorithms. ibv_read_bw Calculats the BW of RDMA read between a pair of machines. One acts as a server, and the other as a client. The client RDMA reads the server memory and calculate the BW by sampling the CPU each time it receive a successfull completion. The test supports a large variety of features as described below, and has better performance than ib_send_bw in Nahelem systems. Read is availible only in RC connection mode (as specified in the InfiniBand spec).

13.8.1 Synopsys

```
ibv_read_bw [-i(b_port) ib_port] [-d ib device] [-o(uts) outstanding reads] [-
m(tu) mtu_size] [-s(ize) message_size] [-t(x-depth) tx_size] [-n iteration_num] [-
p(ort) PDT_port] [-u qp timeout] [-S(l) sl type] [-x gid index] [-e(vents) use
events] [-F CPU freq fail] [-b(idirectional)] [-a(ll)] [-V(ersion)]
```

13.8.2 Options

The table below lists the various flags of the command.

Table 24 - ibv_read_bw Flags and Options

Flag	Description
-p, --port=<port>	Listens on/connect to port <port> (default 18515)
-d, --ib-dev=<dev>	Uses IB device <device guid> (default first device found)
-i, --ib-port=<port>	Uses port <port> of IB device (default 1)
-m, --mtu=<mtu>	The mtu size (default 1024)
-o, --outs=<num>	The number of outstanding read/atom(default for hermon 16 (others 4)
-s, --size=<size>	The size of message to exchange (default 65536)
-a, --all	Runs sizes from 2 till 2^23
-t, --tx-depth=<dep>	The size of tx queue (default 100)
-n, --iters=<iters>	The number of exchanges (at least 2, default 1000)
-u, --qp-timeout=<timeout>	QP timeout. The timeout value is 4 usec * 2 ^(timeout), default 14
-S, --sl=<sl>	The service level (default 0)
-x, --gid-index=<index>	Test uses GID with GID index taken from command line (for RDMAoE index should be 0)
-b, --bidirectional	Measures bidirectional bandwidth (default unidirectional)
-V, --version	Displays version number
-g, --post=<num of posts>	The number of posts for each qp in the chain (default tx_depth)
-e, --events	Inactive during CQ events (default poll)
-F, --CPU-freq	The CPU frequency test. It is active even if the cpufreq_ondemand module is loaded

13.9 ibv_read_lat

This is a more advanced version of ib_read_lat ,and contains more flags and featurs than the older version and also improved algorithms. ibv_read_lat calculats the latency of RDMA read operation of message_sizeB between a pair of machines. One acts as a server and the other as a client. They perform a ping pong benchmark on which one side RDMA reads the memory of the other side only after the other side have read his memory. Each of the sides samples the CPU clock each time they read the other side memory, to calculate latency. Read is availible only in RC connection mode (as specified in InfiniBand spec).

13.9.1 Synopsys

```
ibv_read_lat [-i(b_port) ib_port] [-m(tu) mtu_size] [-s(ize) message_size] [-t(x-
depth) tx_size] [-I(nline_size) inline size] [-u qp timeout] [-S(L) sl type] [-d
ib_device name] [-x gid index] [-n iteration_num] [-o(uts) outstanding reads] [-
```

```
e(vents) use events] [-p(ort) PDT_port] [-a(ll)] [-V(ersion)] [-C report cycles]
[-H report histogram] [-U report unsorted] [-F CPU freq fail]
```

13.9.2 Options

The table below lists the various flags of the command.

Table 25 - ibv_read_lat Flags and Options

Flag	Description
-p, --port=<port>	Listens on/connect to port <port> (default 18515)
-d, --ib-dev=<dev>	Uses IB device <device guid> (default first device found)
-i, --ib-port=<port>	Uses port <port> of IB device (default 1)
-m, --mtu=<mtu>	The mtu size (default 1024)
-o, --outs=<num>	The number of outstanding read/atom(default for hermon 16 (others 4)
-s, --size=<size>	The size of message to exchange (default 65536)
-a, --all	Runs sizes from 2 till 2^23
-t, --tx-depth=<dep>	The size of tx queue (default 100)
-n, --iters=<iters>	The number of exchanges (at least 2, default 1000)
-u, --qp-timeout=<timeout>	QP timeout. The timeout value is 4 usec * 2 ^(timeout), default 14
-S, --sl=<sl>	The service level (default 0)
-x, --gid-index=<index>	Test uses GID with GID index taken from command line (for RDMAoE index should be 0)
-C, --report-cycles	Reports times in cpu cycle units (default microseconds)
-H, --report-histogram	Print out all results (default print summary only)
-U, --report-unsorted (implies -H)	Print out unsorted results (default sorted)
-V, --version	Displays version number
-e, --events	Inactive during CQ events (default poll)
-F, --CPU-freq	The CPU frequency test. It is active even if the cpufreq_ondemand module is loaded

13.10 ibv_send_bw

This is a more advanced version of ib_send_bw and contains more flags and featurs than the older version and also improved algorithms. ibv_send_bw calculats the BW of SEND between a pair of machines. One acts as a server and the other as a client. The server receive packets from the client and they both calculate the throughput of the operation.The test supports a large variety of features as described below, and has better performance than ib_send_bw in Nahelem systems.

13.10.1Synopsys

```
ibv_send_bw [-i(b_port) ib_port] [-d ib device]  [-c(onnection_type) RC\UC\UD] [-
m(tu) mtu_size] [-s(ize) message_size] [-t(x-depth) tx_size] [-r(x_dpeth) rx_size]
[-n iteration_num] [-p(ort) PDT_port] [-I(nline_size) inline size] [-u qp timeout]
[-S(l) sl type] [-x gid index] [-e(vents) use events] [-N(o_peak) use peak calc]
[-F CPU freq fail] [-g num of qps in mcast group] [-M mcast gid] [-b(idirec-
tional)] [-a(ll)] [-V(ersion)]
```

13.10.2Options

The table below lists the various flags of the command.

Table 26 - ibv_send_bw Flags and Options

Flag	Description
-p, --port=<port>	Listens on/connect to port <port> (default 18515)
-d, --ib-dev=<dev>	Uses IB device <device guid> (default first device found)
-i, --ib-port=<port>	Uses port <port> of IB device (default 1)
-m, --mtu=<mtu>	The mtu size (default 1024)
-c, --connection=<RC/UC/UD>	Connection type RC/UC/UD (default RC)
-s, --size=<size>	The size of message to exchange (default 65536)
-a, --all	Runs sizes from 2 till 2^23
-t, --tx-depth=<dep>	The size of tx queue (default 100)
-n, --iters=<iters>	The number of exchanges (at least 2, default 1000)
-u, --qp-timeout=<timeout>	QP timeout. The timeout value is 4 usec * 2 ^(timeout), default 14
-S, --sl=<sl>	The service level (default 0)
-x, --gid-index=<index>	Test uses GID with GID index taken from command line (for RDMAoE index should be 0)
-b, --bidirectional	Measures bidirectional bandwidth (default unidirectional)
-V, --version	Displays version number
-g, --post=<num of posts>	The number of posts for each qp in the chain (default tx_depth)
-e, --events	Inactive during CQ events (default poll)
-F, --CPU-freq	The CPU frequency test. It is active even if the cpufreq_ondemand module is loaded
-r, --rx-depth=<dep>	Makes rx queue bigger than tx (default 600)
-I, --inline_size=<size>	The maximum size of message to be sent in “inline mode” (default 0)
-N, --no peak-bw	Cancels peak-bw calculation (default with peak-bw)
-g, --mcg=<num_of_qps>	Sends messages to multicast group with <num_of_qps> qps attached to it.

Table 26 - ibv_send_bw Flags and Options

Flag	Description
-M, --MGID=<multicast_gid>	In case of multicast, uses <multicast_gid> as the group MGID. The format must be '255:1:X:X:X:X:X:X:X:X:X:X:X:X:X:X', where X is a vlaue within [0,255]

13.11 ibv_send_lat

This is a more advanced version of ib_send_lat and contains more flags and featurs than the older version and also improved algorithms. ibv_send_lat calculats the latency of sending a packet in message_sizeB between a pair of machines. One acts as a server and the other as a client. They perform a ping pong benchmark on which you send packet only after you receive one. Each of the sides samples the CPU clock each time they receive a send packet, in order to calculate the latency.

13.11.1 Synopsys

```
ibv_send_lat [-i(b_port) ib_port]  [-c(onnection_type) RC\UC\UD] [-d ib_device
name] [-m(tu) mtu_size] [-s(ize) message_size] [-t(x-depth) tx_size] [-
I(nline_size) inline size] [-u qp timeout] [-S(L) sl type] [-x gid index] [-
e(events) use events] [-n iteration_num] [-g num of qps in mcast group] [-p(ort)
PDT_port] [-a(ll)] [-V(ersion)] [-C report cycles] [-H report histogram] [-U
report unsorted] [-F CPU freq fail]
```

13.11.2 Options

The table below lists the various flags of the command.

Table 27 - ibv_send_lat Flags and Options

Flag	Description
-p, --port=<port>	Listens on/connect to port <port> (default 18515)
-d, --ib-dev=<dev>	Uses IB device <device guid> (default first device found)
-i, --ib-port=<port>	Uses port <port> of IB device (default 1)
-m, --mtu=<mtu>	The mtu size (default 1024)
-c, --connection=<RC/UC/UD>	Connection type RC/UC/UD (default RC)
-s, --size=<size>	The size of message to exchange (default 65536)
-l, --signal	Signal completion on each msg
-a, --all	Runs sizes from 2 till 2^23
-t, --tx-depth=<dep>	The size of tx queue (default 100)
-n, --iters=<iters>	The number of exchanges (at least 2, default 1000)
-u, --qp-timeout=<timeout>	QP timeout. The timeout value is 4 usec * 2 ^(timeout), default 14
-S, --sl=<sl>	The service level (default 0)

Table 27 - ibv_send_lat Flags and Options

Flag	Description
-x, --gid-index=<index>	Test uses GID with GID index taken from command line (for RDMAoE index should be 0)
-C, --report-cycles	Reports times in cpu cycle units (default microseconds)
-H, --report-histogram	Print out all results (default print summary only)
-U, --report-unsorted (implies -H)	Print out unsorted results (default sorted)
-V, --version	Displays version number
-F, --CPU-freq	The CPU frequency test. It is active even if the cpufreq_ondemand module is loaded
-g, --post=<num of posts>	The number of posts for each qp in the chain (default tx_depth)
-I, --inline_size=<size>	The maximum size of message to be sent in “inline mode” (default 0)
-e, --events	Inactive during CQ events (default poll)
-g, --mcg=<num_of_qps>	Sends messages to multicast group with <num_of_qps> qps attached to it.
-M, --MGID=<multicast_gid>	In case of multicast, uses <multicast_gid> as the group MGID. The format must be '255:1:X:X:X:X:X:X:X:X:X:X:X:X:X:X', where X is a vlaue within [0,255]. You must specify a different MGID on both sides to avoid loopback.

13.12 ibv_write_bw

This is a more advanced version of ib_write_bw ,and contains more flags and featurs than the older version and also improved algorithms. ibv_write_bw calculats the BW of RDMA write between a pair of machines. One acts as a server and the other as a client. The client RDMA writes to the server memory and calculate the BW by sampling the CPU each time it receive a successfull completion. The test supports a large variety of features as described below, and has better performance than ib_send_bw in Nahelem systems.

13.12.1 Synopsys

```
ibv_write_bw [-i(b_port) ib_port] [-d ib device]  [-c(onnection_type) RC\UC\UD] [-
m(tu) mtu_size] [-s(ize) message_size] [-t(x-depth) tx_size] [-n iteration_num] [-
p(ort) PDT_port] [-I(nline_size) inline size] [-u qp timeout] [-S(l) sl type] [-x
gid index] [-e(vents) use events] [-N(o_peak) use peak calc] [-F CPU freq fail] [-
g num of posts] [-q num of qps] [-b(idirectional)] [-a(ll)] [-V(ersion)]
```

13.12.2 Options

The table below lists the various flags of the command.

Table 28 - ibv_write_bw Flags and Options

Flag	Description
-p, --port=<port>	Listens on/connect to port <port> (default 18515)

Table 28 - ibv_write_bw Flags and Options

Flag	Description
-d, --ib-dev=<dev>	Uses IB device <device guid> (default first device found)
-i, --ib-port=<port>	Uses port <port> of IB device (default 1)
-m, --mtu=<mtu>	The mtu size (default 1024)
-c, --connection=<RC/UC>	Connection type RC/UC(default RC)
-s, --size=<size>	The size of message to exchange (default 65536)
-a, --all	Runs sizes from 2 till 2^23
-t, --tx-depth=<dep>	The size of tx queue (default 100)
-n, --iters=<iters>	The number of exchanges (at least 2, default 1000)
-u, --qp-timeout=<timeout>	QP timeout. The timeout value is 4 usec * 2 ^(timeout), default 14
-S, --sl=<sl>	The service level (default 0)
-x, --gid-index=<index>	Test uses GID with GID index taken from command line (for RDMAoE index should be 0)
-b, --bidirectional	Measures bidirectional bandwidth (default unidirectional)
-V, --version	Displays version number
-g, --post=<num of posts>	The number of posts for each qp in the chain (default tx_depth)
-F, --CPU-freq	The CPU frequency test. It is active even if the cpufreq_ondemand module is loaded
-q, --qp=<num of qp's>	The number of qp's (default 1)
-I, --inline_size=<size>	The maximum size of message to be sent in “inline mode” (default 0)
-N, --no peak-bw	Cancels peak-bw calculation (default with peak-bw)

13.13 ibv_write_lat

This is a more advanced version of ib_write_lat and contains more flags and featurs than the older version and also improved algorithms. ibv_write_lat calculats the latency of RDMA write operation of message_sizeB between a pair of machines. One acts as a server, and the other as a client. They perform a ping pong benchmark on which one side RDMA writes to the other side memory only after the other side wrote on his memory. Each of the sides samples the CPU clock each time they write to the other side memory to calculate latency.

13.13.1 Synopsys

```
ibv_write_lat [-i(b_port) ib_port]  [-c(onnection_type) RC\UC\UD] [-m(tu)
mtu_size] [-s(ize) message_size] [-t(x-depth) tx_size] [-I(nline_size) inline
size] [-u qp timeout] [-S(L) sl type] [-d ib_device name] [-x gid index] [-n
iteration_num] [-p(ort) PDT_port] [-a(ll)] [-V(ersion)] [-C report cycles] [-H
report histogram] [-U report unsorted]
```

13.13.2Options

The table below lists the various flags of the command.

Table 29 - ibv_write_lat Flags and Options

Flag	Description
-p, --port=<port>	Listens on/connect to port <port> (default 18515)
-d, --ib-dev=<dev>	Uses IB device <device guid> (default first device found)
-i, --ib-port=<port>	Uses port <port> of IB device (default 1)
-m, --mtu=<mtu>	The mtu size (default 1024)
-c, --connection=<RC/UC>	Connection type RC/UC (default RC)
-s, --size=<size>	The size of message to exchange (default 65536)
-l, --signal	Signal completion on each msg
-a, --all	Runs sizes from 2 till 2^23
-t, --tx-depth=<dep>	The size of tx queue (default 100)
-n, --iters=<iters>	The number of exchanges (at least 2, default 1000)
-u, --qp-timeout=<timeout>	QP timeout. The timeout value is 4 usec * 2 ^(timeout), default 14
-S, --sl=<sl>	The service level (default 0)
-x, --gid-index=<index>	Test uses GID with GID index taken from command line (for RDMAoE index should be 0)
-C, --report-cycles	Reports times in cpu cycle units (default microseconds)
-H, --report-histogram	Print out all results (default print summary only)
-U, --report-unsorted (implies -H)	Print out unsorted results (default sorted)
-V, --version	Displays version number
-F, --CPU-freq	The CPU frequency test. It is active even if the cpufreq_ondemand module is loaded
-I, --inline_size=<size>	The maximum size of message to be sent in “inline mode” (default 0)

14 Documentation

- Under <installation_directory>\documents:
  - Release Notes for: core (IBAL), IPoIB, WSD
  - README and user manuals for: opensm, SDP
- Under <installation_directory>:
  - License file
  - This document
- Under <installation_directory>\SDK:
  - core (IBAL) API HTML documentation (in SDK package)
  - hello_world code example (in SDK package): This is a 'two-sided' code example built by the DDK environment

    Activation:
    - Side A: hello_world.exe -d [daemon options]
    - Side B: hello_world.exe --ip=<daemon_host_ip> [client options]

    For options, enter: hello_world.exe --help