WinOF VPI for Windows

User Manual

Rev 3.0.0

www.mellanox.com

NOTE:

THIS HARDWARE, SOFTWARE OR TEST SUITE PRODUCT (“PRODUCT(S)”) AND ITS RELATED DOCUMENTATION ARE PROVIDED BY MELLANOX TECHNOLOGIES “AS-IS” WITH ALL FAULTS OF ANY KIND AND SOLELY FOR THE PURPOSE OF AIDING THE CUSTOMER IN TESTING APPLICATIONS THAT USE THE PRODUCTS IN DESIGNATED SOLUTIONS. THE CUSTOMER'S MANUFACTURING TEST ENVIRONMENT HAS NOT MET THE STANDARDS SET BY MELLANOX TECHNOLOGIES TO FULLY QUALIFY THE PRODUCTO(S) AND/OR THE SYSTEM USING IT. THEREFORE, MELLANOX TECHNOLOGIES CANNOT AND DOES NOT GUARANTEE OR WARRANT THAT THE PRODUCTS WILL OPERATE WITH THE HIGHEST QUALITY. ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT ARE DISCLAIMED. IN NO EVENT SHALL MELLANOX BE LIABLE TO CUSTOMER OR ANY THIRD PARTIES FOR ANY DIRECT, INDIRECT, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES OF ANY KIND (INCLUDING, BUT NOT LIMITED TO, PAYMENT FOR 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 FROM THE USE OF THE PRODUCT(S) AND RELATED DOCUMENTATION 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.
Beit Mellanox
PO Box 586 Yokneam 20692
Israel
www.mellanox.com
Tel: +972 (0)4 909 7200 ; +972 (0)74 723 7200
Fax: +972 (0)4 959 3245

Table of Contents

List of Tables

Document Revision History

Table 1 - Document Revision History

Document Revision	Date	Changes
Rev 3.0.0	February 08, 2012	• Added section RDMA over Converged Ethernet (RoCE) and its subsections • Added section Hyper-V with VMQ • Added section Network Driver Interface Specification (NDIS) • Added section Header Data Split • Added section Auto Sensing • Added section Adapter Teaming • Added section Port Protocol Configuration • Added section Advanced Configuration for InfiniBand Driver • Added section Advanced Configuration for Ethernet Driver • Added section Updated section Tunable Performance Parameters • Added section Merged Ethernet and InfiniBand features sections • Removed section Sockets Direct Protocol and its subsections • Removed section Winsock Direct and Protocol and its subsections • Removed section Added ConnectX®-3 support • Removed section IPoIB Drivers Overview • Removed section Booting Windows from an iSCSI Target
Rev 2.1.3	January 28. 2011	Complete restructure
Rev 2.1.2	October 10, 2010	• Removed section Debug Options. • Updated Section 3, “Uninstalling Mellanox VPI Driver,” on page 11 • Added Section 6, “InfiniBand Fabric,” on page 38 and its subsections • Added Section 6.3, “InfiniBand Fabric Performance Utilities,” on page 71 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

1 About this Manual

1.1 Scope

The document describes WinOF Rev 3.0.0 features, content and configuration. Additionally, this document provides information on various performance tools supplied with this version.

1.2 Intended Audience

This manual is intended for system administrators responsible for the installation, configuration, management and maintenance of the software and hardware of VPI (InfiniBand, Ethernet) adapter cards. It is also intended for application developers.

1.3 Documentation Conventions

Table 2 - Documentation 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]	
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>	 May result in system instability.

1.3.1 Common Abbreviations and Acronyms

Table 3 - 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)
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
IPoIB	IP over InfiniBand
PFC	Priority Flow Control
PR	Path Record
RDS	Reliable Datagram Sockets
RoCE	RDMA over Converged Ethernet
SL	Service Level
MPI	Message Passing Interface
EoIB	Ethernet over Infiniband
QoS	Quality of Service
ULP	Upper Level Protocol
VL	Virtual Lane

2 Introduction

This User Manual addresses the Mellanox WinOF VPI driver Rev 3.0.0 package distributed for Windows Server 2008 (x86 and x64), Windows Server 2008 R2 (x64) and Windows 7 (x86 and x64).

Mellanox WinOF VPI is composed of several software modules that contain an InfiniBand and Ethernet driver. The Mellanox WinOF VPI driver supports Infiniband and 10GB Ethernet ports. The port type is determined upon boot based on card's capability and user setting.

2.1 Mellanox VPI Package Contents

The Mellanox WinOF for Windows package contains the following components:

- Core and ULPs
  - IB network adapter cards low-level drivers (mlx4)
  - IB Access Layer (IBAL)
  - Ethernet driver (ETH)
  - IP over InfiniBand (IPoIB)
  - Upper Layer Protocols (ULPs):
    - NetworkDirect (ND)
- Utilities
- SW Development Kit (SDK)
- Documentation

2.2 Hardware and Software Requirements

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

2.3 Supported Network Adapter Cards and Firmware Versions

Mellanox WinOF VPI Rev 3.0.0 supports the following Mellanox network adapter cards:

IB

- ConnectX®-2 EN IB SDR/DDR/QDR (fw-25408 Rev 2.9.1000)

VPI / Ethernet

- ConnectX® / ConnectX®-2 / ConnectX® EN / IB SDR/DDR/QDR (fw-25408 Rev 2.9.1000)
- ConnectX®-3 FDR/SDR/QDR (fw-25408 Rev 2.10.0000 and higher)

We recommend upgrading ConnectX and ConnectX-2 adapter cards to firmware 2.9.1000 or higher to enable improved functionality while using this WinOF release.
For further information, see Section 2.4.1, “Downloading the Firmware Tools Package,” on page 10.

2.4 Managing Firmware

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

2.4.1 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 Software for Windows_x86" for x86 architecture and "MFT Software for Windows_x64" for x64 architecture.

2. Install and Run WinMFT

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

   Install the WinMFT package from the command line with administrator privileges.

   Enter:

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

3. Check the Device Status
   - start/stop mst is automatically done by the tools > C:\Users\herod\Desktop>mst start
   - To check device status run > mst status

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

   ```
   mt<device id>_pciconf0

   mt<device id>_pci_cr0
   ```

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

2.4.2 Downloading the Firmware Image of the Adapter Card

- To download the correct card firmware image, please visit

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

- To identify your adapter card, please visit
  http://www.mellanox.com > Support > Firmware Downloads > Identifying Adapter Cards

2.4.3 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
```

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

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

3 Driver Features

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
- MSI-X support (only on Windows Server 2008 and higher)
- Auto Sensing
- RoCE

Ethernet Only:

- High Availability (HA) between ports and Mellanox NICs
- Load Balancing between ports and Mellanox NICs
- HW VLAN filtering
- Hyper-V
- Header Data Split

For the complete list of Ethernet and InfiniBand Know Issues and Limitation, see MLNX_WinVPI_ReleaseNotes.txt.

3.1 RDMA over Converged Ethernet (RoCE)

3.1.1 RoCE Overview

Remote Direct Memory Access (RDMA) is the remote memory management capability that allows server to server data movement directly between application memory without any CPU involvement. RDMA over Ethernet (RoCE) is a mechanism to provide this efficient data transfer with very low latencies on loss-less Ethernet networks. With advances in data center convergence over reliable Ethernet, ConnectX®-2/ConnectX®-3 EN with RoCE uses the proven and efficient RDMA transport to provide the platform for deploying RDMA technology in mainstream data center application at 10GigE and 40GigE link-speed. ConnectX®-2/ConnectX®-3 EN with its hardware offload support takes advantage of this efficient RDMA transport (InfiniBand) services over Ethernet to deliver ultra low latency for performance-critical and transaction intensive applications such as financial, data base, storage, and content delivery networks.

RoCE encapsulates IB transport and GRH headers in Ethernet packets bearing a dedicated ether type. While the use of GRH is optional within IB subnets, it is mandatory when using RoCE. Applications written over IB verbs should work seamlessly, but they require provisioning of GRH

information when creating address vectors. The library and driver are modified to provide mapping from GID to MAC addresses required by the hardware.

3.1.2 Ported Applications

The following applications are ported with RoCE:

- Network Direct (ND) applications work without any change
- Performance tests

3.1.3 Reading Port Counters Statistics

RoCE port statistics are not shown in the Windows network counters associated with Etherent interface. It is possible to read port statistics in the same way it is done for regular InfiniBand ports. The information is available by running `vstat -c`.

3.1.4 Setting RoCE

» To set the RoCE, please perform the following:

Step 1. Open Device Manager and expand System devices display pane.

Step 2. Right-click the Mellanox ConnectX VPI (MT26428) - PCIe 2.0 5GT/s, IB QDR / 10GigE Network Adapter

entry and left-click Properties.

Step 3. Click the Port Protocol tab and check RoCE check box.

Step 4. Click OK. It's will cause to driver restart

3.1.5 Setting RoCE MTU

Ethernet packet uses the general MTU value whereas the RoCE packet uses the RoCE MTU.

All devices that run the RoCE protocol must have the same MTU, otherwise packets larger than the minimum MTU are dropped and not transferred.

When RoCE is enabled, you can configure the MTU that can be sent by the RoCE protocol.

- The valid RoCE MTU values are: 256, 512, 1024, 2048

  When using MTU 2048, the administrator should configure the switches to support MTU 2048 or higher.

- The default MTU is 1024

» To set the RoCE MTU, please perform the following:

Step 1 Open Device Manager and expand Network Adapters in the device display pane.

Step 2. Right-click the Mellanox ConnectX 10Gb Ethernet Adapter entry and left-click Properties.

Step 3. Click the Advanced tab and modify the desired properties.

Step 4. Select RoCE Options and click Properties to modify the settings as needed.

Step 5. Click OK

3.2 Hyper-V with VMQ

Mellanox WinOF Rev 3.0.0 includes a virtual machine queue (VMQ) interface to support Microsoft Hyper-V network performance improvements and security enhancement.

VMQ interface supports:

- Classification of received packets by using the destination MAC address to route the packets to different receive queues
- NIC ability to use DMA to transfer packets directly to a Hyper-V child-partition's shared memory
- Scaling to multiple processors, by processing packets for different virtual machines on different processors.

VMQ is disabled by default for Windows 2008 R2.

3.2.1 Enabling Virtual Machine Queue on Windows 2008 R2

To enable VMQ on Windows 2008 R2 with 10 Gbps physical network adapters, set the registry keys as follow:

Step 1. Open Command Prompt window, Click Start--> All Programs.

Step 2. Click Accessories, right-click Command Prompt and then click Run as administrator.

Step 3. Type reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\VMSMP\Parameters\TenGigVmqEnabled /t REG_DWORD /d 1 /f

Step 4. Click ENTER.

Step 5. Reboot

3.3 Header Data Split

The header-data split feature improves network performance by splitting the headers and data in received Ethernet frames into separate buffers. The feature is disabled by default and can be enabled in the Advanced tab (Performance Options) from the Properties sheet.

For further information, please refer to the MSDN library:

http://msdn.microsoft.com/en-us/library/windows/hardware/ff553723(v=VS.85).aspx

3.4 Receive Side Scaling (RSS)

Mellanox WinOF Rev 3.0.0 IPoIB and Ethernet drivers use NDIS 6.2 new RSS capabilities. The main changes are:

- Supports unlimited number of processors (previously 64)
- Individual network adapter RSS configuration usage

To set the RSS capability for individual adapter instead of global setting, and to improve RSS on Windows 2008 R2 and Windows 7, set the registry keys listed in the table below:

Table 4 - Registry Keys Setting

Sub-key	Description
HKLM\SYSTEM\CurrentControlSet\Control\Class\{XXXXX72-XXX}\<network adapter number>*MaxRSSProcessors	**Maximum number of CPUs allotted.** Sets the desired maximum number of processors for each interface. The number can be different for each interface. Note: Restart the network adapter when you change this registry key.
HKLM\SYSTEM\CurrentControlSet\Control\Class\{XXXXX72-XXX}\<network adapter number>*RssBaseProcNumber	**Base CPU number.** Sets the desired base CPU number for each interface. The number can be different for each interface. This allows partitioning of CPUs across network adapters. Note: Restart the network adapter when you change this registry key.
HKLM\SYSTEM\CurrentControlSet\Control\Class\{XXXXX72-XXX}\<network adapter number>*NumaNodeID	**NUMA node affinitization**
HKLM\SYSTEM\CurrentControlSet\Control\Class\{XXXXX72-XXX}\<network adapter number>*RssBaseProcGroup	Sets the RSS base processor group for systems with more than 64 processors.

3.5 Port Configuration

After MLNX_VPI installation, it is possible to modify the network protocol that runs on each port of VPI adapter cards. Each port can be set to run as InfiniBand, Ethernet or Auto Sensing.

3.5.1 Auto Sensing

Auto Sensing enables the NIC to automatically sense the link type (InfiniBand or Ethernet) based on the cable connected to the port and load the appropriate driver stack (InfiniBand or Ethernet).

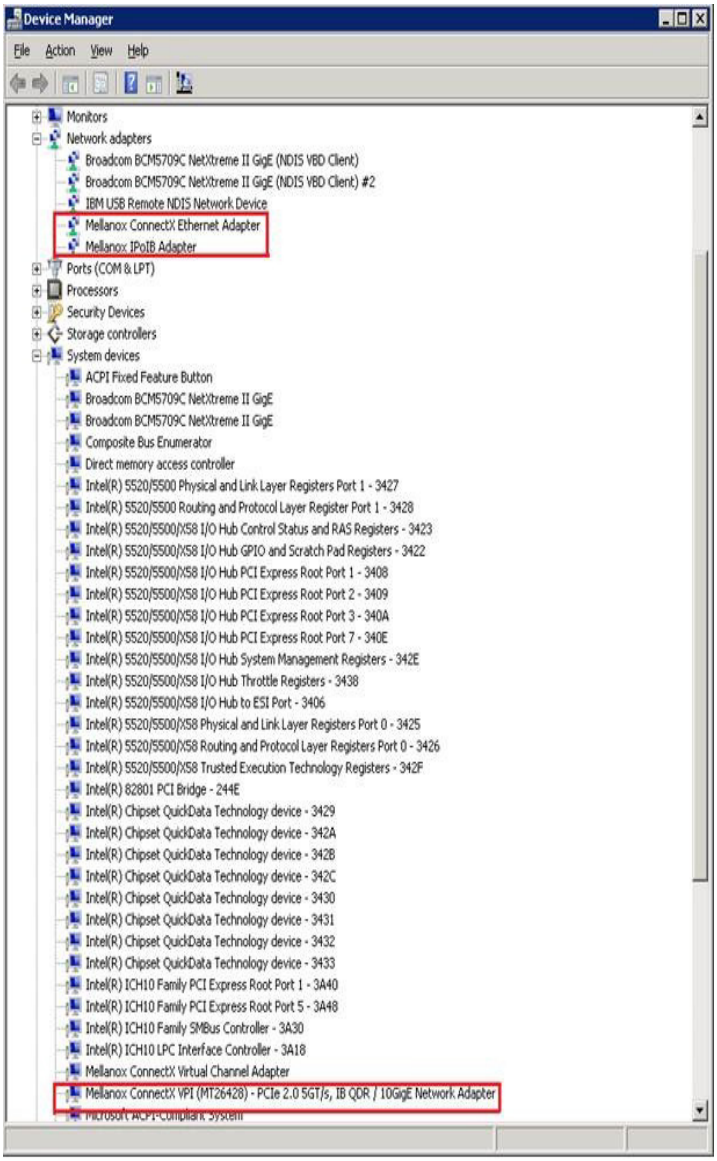
For example, if the first port is connected to an InfiniBand switch and the second to Ethernet switch, the NIC will automatically load the first port as InfiniBand and the second as Ethernet.

Auto Sensing is performed only when rebooting the machine or after disabling/enabling the mlx4_bus interface from the Device Manager. Hence, if you replace cables during the runtime, the NIC will not perform Auto Sensing.

For further information on how to configure it, please refer to Section 3.5.2, “Port Protocol Configuration,” on page 16.

3.5.2 Port Protocol Configuration

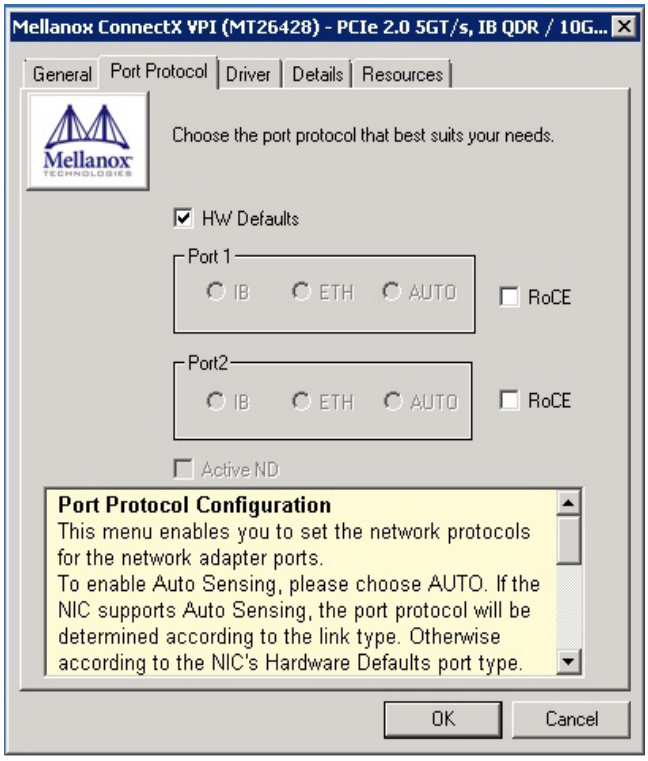
Step 1 Display the Device Manager and expand “Network adapters”.



Step 2. Right-click on the Mellanox ConnectX VPI network adapter and left-click Properties. Select the Port Protocol tab from the Properties sheet.

The "Port Protocol" tab is displayed only if the NIC is a VPI (IB and ETH).

The figure below is an example of the displayed Port Protocol sheet for a dual port VPI adapter card.



Step 3. In this step, you can perform the following functions:

- Choose HW Defaults option. If you choose the HW Defaults option, the port protocols will be determine according to the NIC's hardware default values.
- Choose the desired port protocol for the available port(s). If you choose IB or ETH, both ends of the connection must be of the same type (IB or ETH).
- Enable Auto Sensing by checking the AUTO checkbox. If the NIC does not support Auto Sensing, the AUTO option will be grayed out.
- Enable InfiniBand application over Ethernet by checking the RoCE checkbox. RoCE can be selected on the second port (Port2) only if Port 1 is set as either IB or Ethernet with RoCE enabled.
- Installing Network Direct (ND) by checking the Active ND checkbox

IB must be always the first port in Port 1. If you choose ETH as your first port in Port 1, then the second port in Port2 can be only ETH.

3.6 Load Balancing, Fail-Over (LBFO) and VLAN

3.6.1 Adapter Teaming

Adapter teaming can group a group of ports inside a network adapters or a number of physical network adapters into virtual adapters that provide the fault-tolerance and load-balancing functions. Depending on the teaming mode, one or more interfaces can be active. The non active interfaces in a team are in a standby mode and will take over the network traffic in the event of a link failure in the active interfaces. All of the active interfaces in a team participate in load-balancing operations by sending and receiving a portion of the total network traffic.

3.6.1.1 Teaming (Bundle) Modes

1. Fault Tolerance

   Provides automatic redundancy for the server's network connection. If the primary adapter fails, the secondary adapter (currently in a standby mode) takes over. Fault Tolerance is the basis for each of the following teaming types and is inherent in all teaming modes.

2. Switch Fault Tolerance

   Provides a failover relationship between two adapters when each adapter is connected to a separate switch.

3. Send Load Balancing

   Provides load balancing of transmit traffic and fault tolerance. The load balancing is perform only on the send port.

4. Load Balancing (Send & Receive)

   Provides load balancing of transmit and receive traffic and fault tolerance. The load balancing splits the transmit and receive traffic statically among the team adapters (without changing the base of the traffic loading) based on the source/destination MAC and IP addresses.

5. Adaptive Load Balancing

   The same functionality as Load Balancing (Send & Receive). In case of traffic load in one of the adapters, the load balancing channels the traffic between the other team adapter.

6. Dynamic Link Aggregation (802.3ad)

   Provides dynamic link aggregation allowing creation of one or more channel groups using same-speed or mixed-speed server adapters.

7. Static Link Aggregation (802.3ad)

   Provides increased transmission and reception throughput in a team comprised of two to eight adapter ports through static configuration.

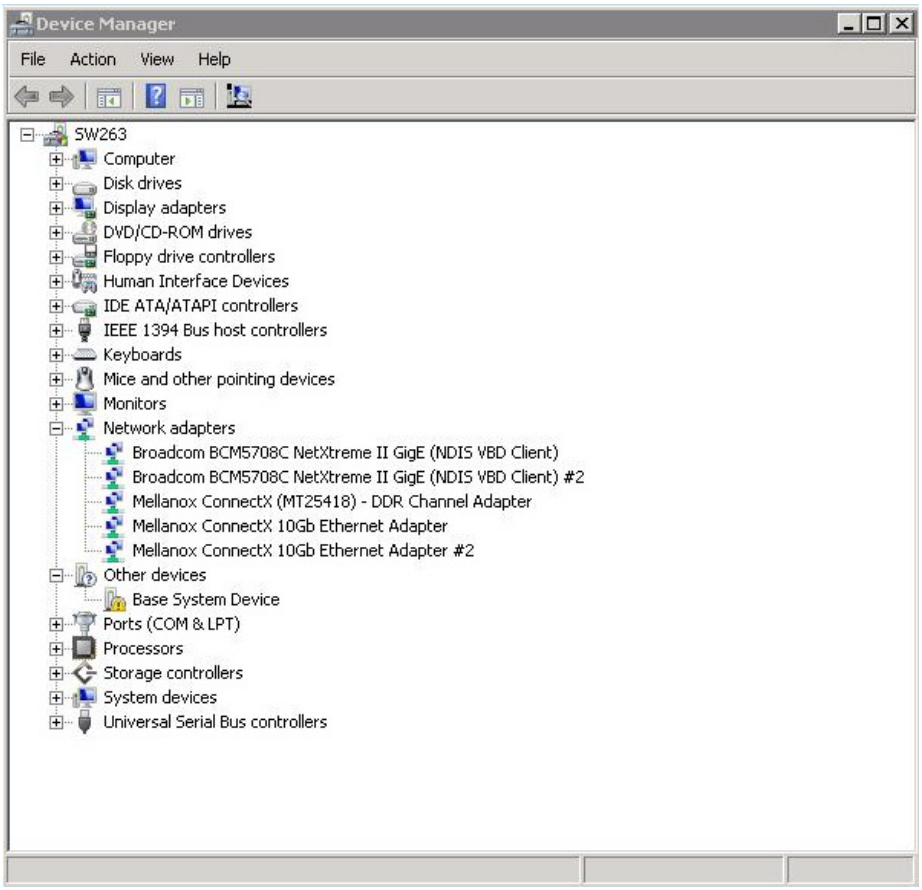
   If the switch connected to the HCA supports 802.3ad the recommended setting is teaming mode 6.

3.6.2 Creating a Load Balancing and Fail-Over (LBFO) Bundle

LBFO is used to balance the workload of packet transfers by distributing the workload over a bundle of network instances and to set a secondary network instance to take over packet indications and information requests if the primary network instance fails.

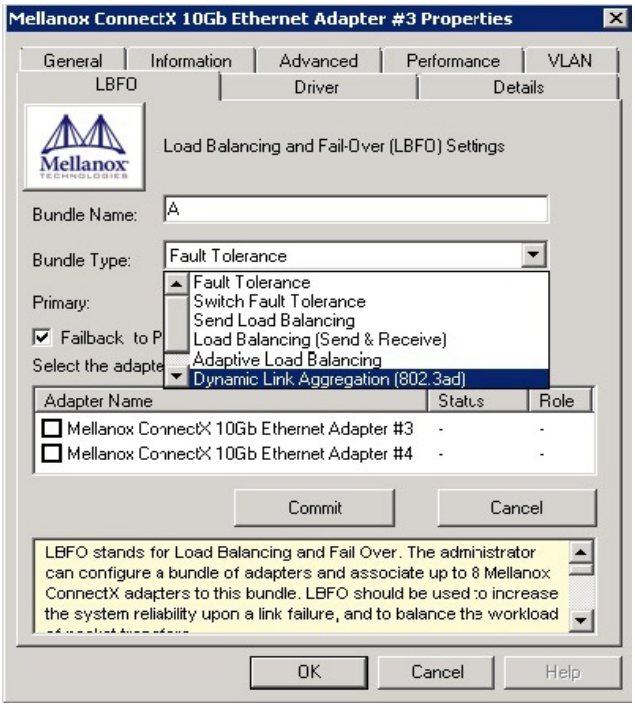
The following steps describe the process of creating an LBFO bundle.

Step 1 Display the Device Manager.



Step 2. Right-click a Mellanox ConnectX 10Gb Ethernet adapter (under "Network adapters" list) and left-click Properties. Select the LBFO tab from the Properties sheet.

It is ***not*** recommended to open the Properties sheet of more than one adapter simultaneously.



Step 3. The LBFO dialog enables creating, modifying or removing a bundle.

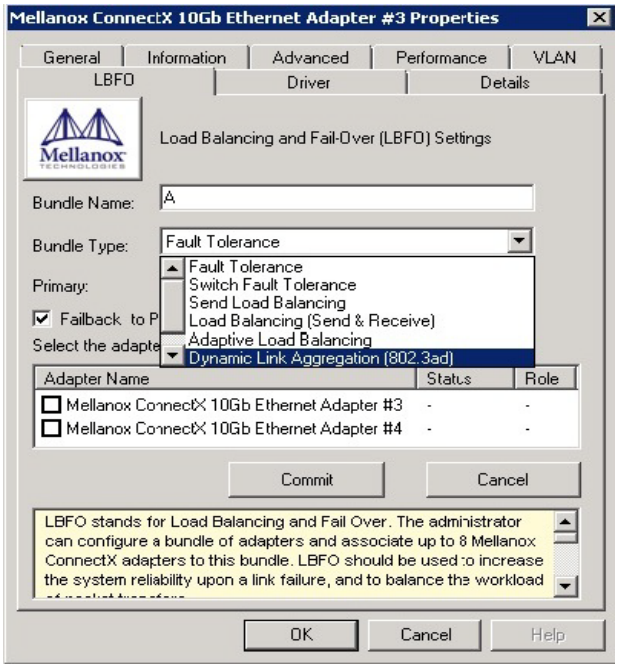
Only Mellanox Technologies adapters can be part of the LBFO.

To *create* a new bundle, perform the following:

1. Click the Create button.
2. Enter a (unique) bundle name.
3. Select a bundle type.
4. Select the adapters to be included in the bundle (that have not been associated with a VLAN).
5. [Optional] Select Primary Adapter.

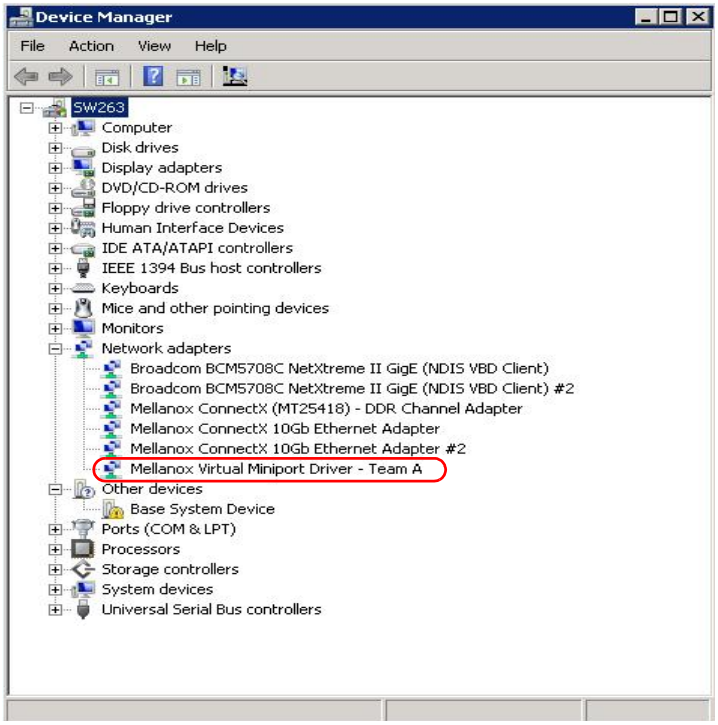
   An active-passive scenario used for data transfer of link disconnecting. In such scenario, the system uses one of the other interfaces.

   When the primary link comes up, the LBFO interface returns to transfer data using the primary interface. If the primary adapter is not selected, the primary interface is selected randomly.
6. [Optional] Failback to Primary
7. Click the Commit button.



The newly created virtual Mellanox adapter representing the bundle will be displayed by the Device Manager under "Network adapters" in the following format (see figure below):

`Mellanox Virtual Miniport Driver - Team <bundle_name>`



To *modify* an existing bundle, perform the following:

1. Select the desired bundle and click Modify
2. Modify the bundle name, its type, and/or the participating adapters in the bundle
3. Click the Commit button

To *remove* an existing bundle, select the desired bundle and click Remove. You will be prompted to approve this action.

Notes on this step:

a. Each adapter that participates in a bundle has two properties:

- Status: Connected/Disconnected/Disabled
- Role: Active or Backup

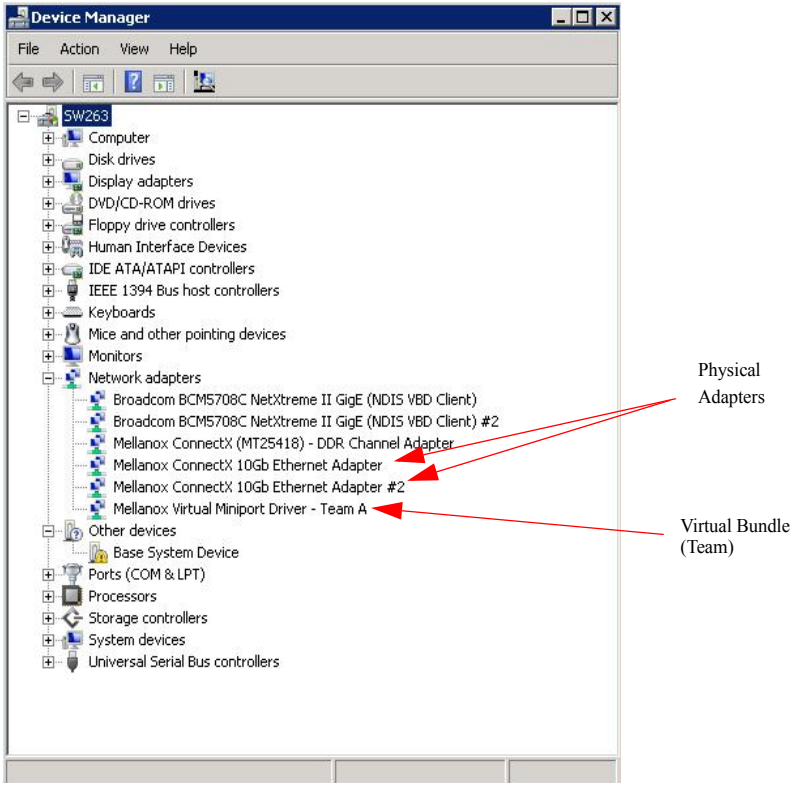
b. Each network adapter that is added or removed from a bundle gets refreshed (i.e., disabled then enabled). This may cause a temporary loss of connection to the adapter.

c. In case a bundle loses one or more network adapters by a "create" or "modify" operation, the remaining adapters in the bundle are automatically notified of the change.

3.6.3 Creating a Port VLAN

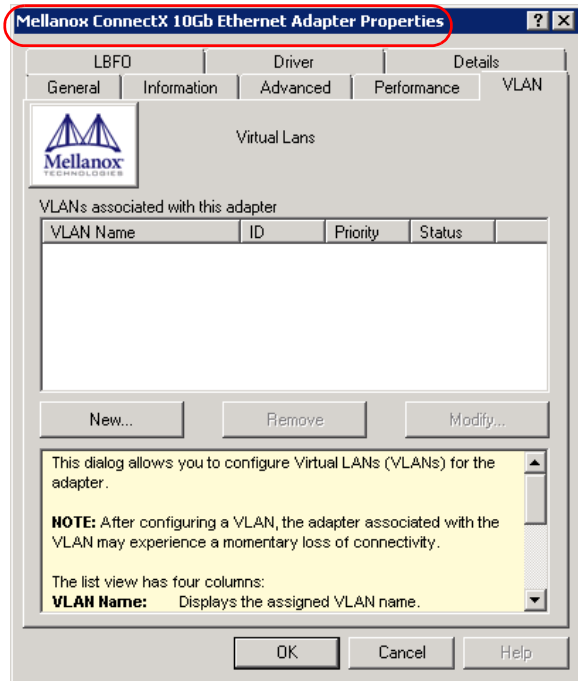
You can create a Port VLAN either on a *physical* Mellanox ConnectX EN adapter or a *virtual* bundle (team). The following steps describe how to create a port VLAN.

Step 1 Display the Device Manager.

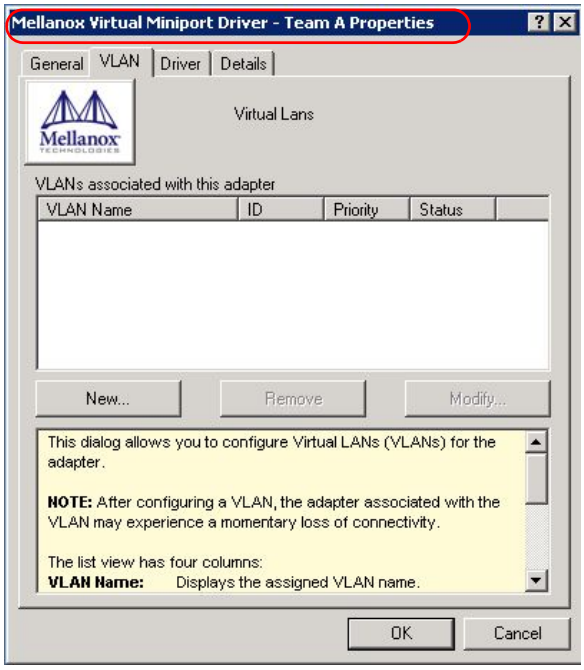


Step 2. Right-click a Mellanox network adapter (under "Network adapters" list) and left-click Properties. Select the VLAN tab from the Properties sheet.

Physical Adapter

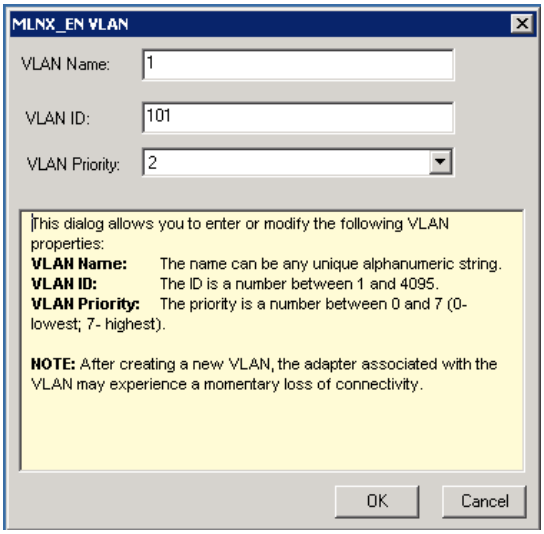


Virtual Bundle (Team)



If a physical adapter has been added to a bundle (team), then the VLAN tab will not be displayed.

Step 3. Click New to open a VLAN dialog window. Enter the desired VLAN Name and VLAN ID, and select the VLAN Priority.

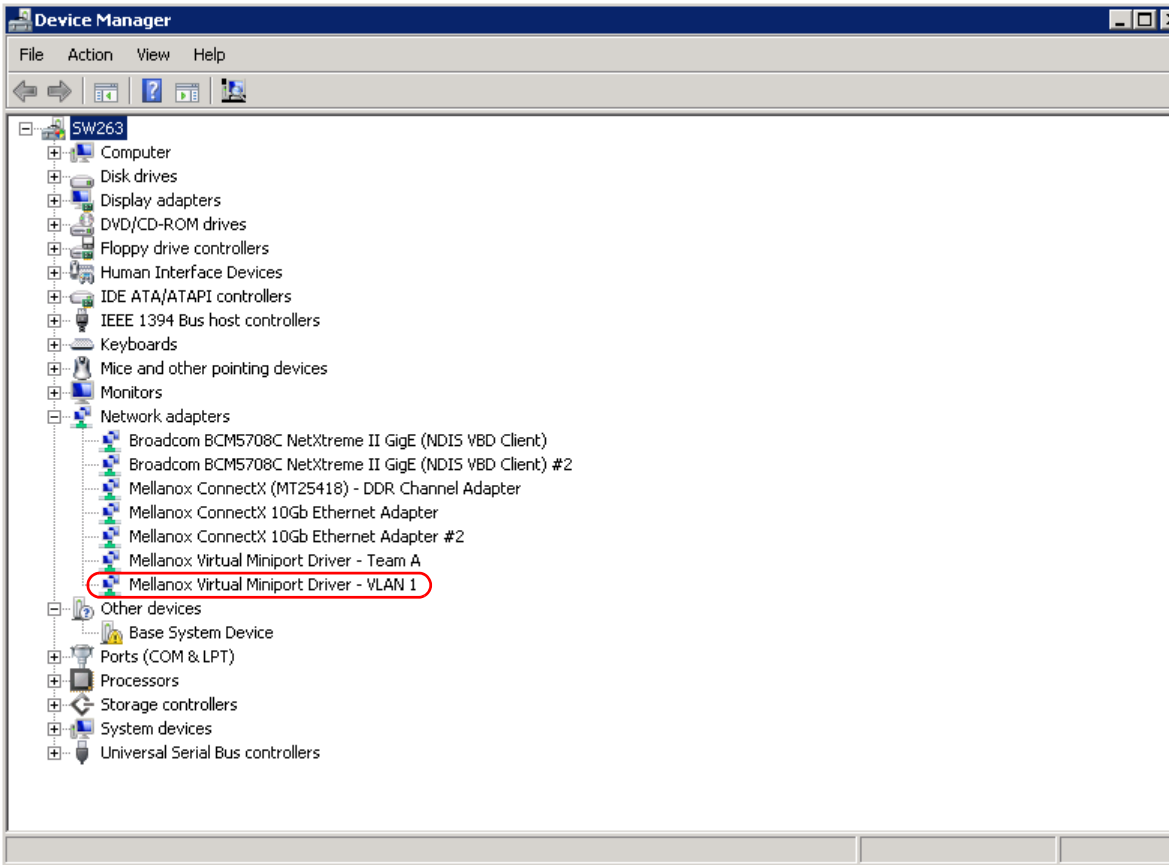


After installing the first virtual adapter (VLAN) on a specific port, the port becomes disabled. This means that it is not possible to bind to this port until all the virtual adapters associated with it are removed.

When using a VLAN, the network address is configured using the VLAN ID. Therefore, the VLAN ID on both ends of the connection must be the same.

Step 4. Verify the new VLAN(s) by opening the Device Manager window or the Network Connections window. The newly created VLAN will be displayed in the following format:

```
Mellanox Virtual Miniport Driver - VLAN <name>
```



3.6.4 Removing a Port VLAN

To remove a port VLAN, perform the following steps:

Step 1 In the Device Manager window, right-click the network adapter from which the port VLAN was created.

Step 2. Left-click Properties.

Step 3. Select the VLAN tab from the Properties sheet.

Step 4. Select the VLAN to be removed.

Step 5. Click Remove and confirm the operation.

4 Driver Configuration

Once you have installed Mellanox WinOF VPI package, you can perform various modifications to your driver to make it suitable for your system's needs

Changes made to the Windows registry happen immediately, and no backup is automatically made.
Do ***not*** edit the Windows registry unless you are confident regarding the changes.

4.1 Configuring the InfiniBand Driver

4.1.1 Modifying Mellanox HCA Configuration

To modify HCA configuration after installation, perform the following steps:

Step 1 Open the Registry editor by clicking Start->Run and entering '`regedit`'.

Step 2. In the navigation pane, expand HKEY_LOCAL_MACHINE->SYSTEM->CurrentControlSet->Services.

Step 3. Expand (in the navigation pane) the HCA driver service entry:

- 'mtcha' for the InfiniHost family
- 'mlx4_hca' and 'mlx4_bus' for the ConnectX family

Step 4. Click the Parameters entry in the expanded driver service entry to display HCA parameters.

Step 5. Double click the desired HCA parameter and modify it. Repeat this step for all the parameters you wish to modify.

Step 6. Close the Registry editor after completing all modifications.

Step 7. Open Device Manager and expand the correct InfiniBand Channel Adapters entry (i.e., the adapter with modified parameters).

Step 8. Right click the expanded HCA entry and left-click Disable. This disables the device.

Step 9. Right click the expanded HCA entry and left-click Enable. This re-enables the device.

For the changes to take effect, you must disable and re-enable the HCA (Steps 8and 9 above).

4.1.2 Modifying IPoIB Configuration

To modify the IPoIB configuration after installation, perform the following steps:

Step 1 Open Device Manager and expand Network Adapters in the device display pane.

Step 2. Right-click the Mellanox IPoIB Adapter entry and left-click Properties.

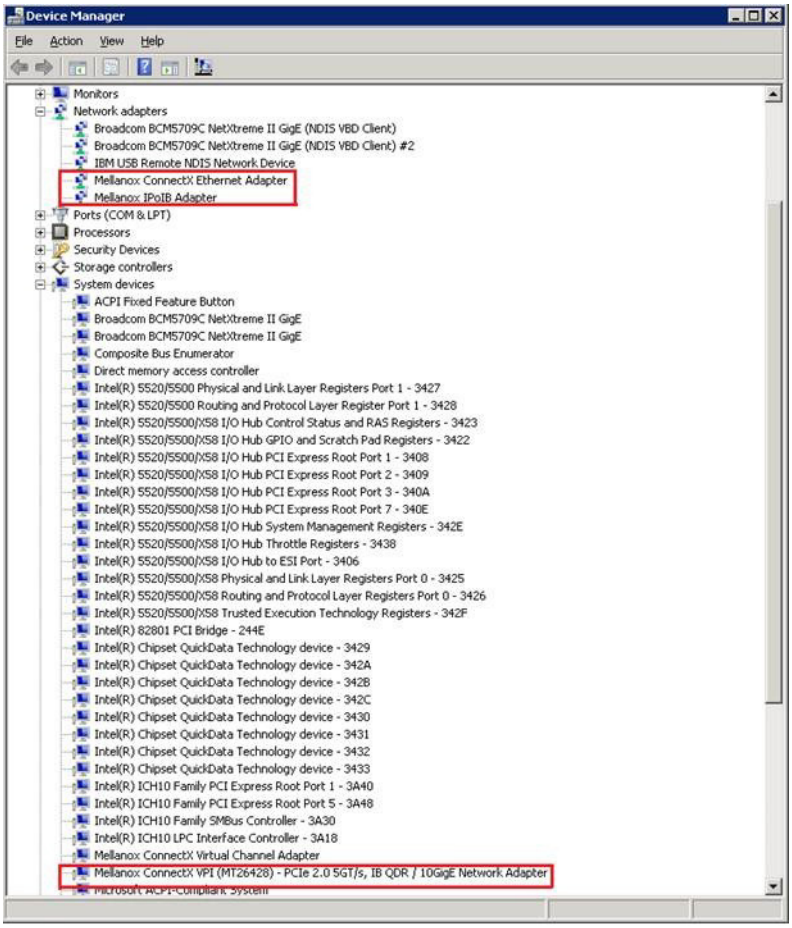
Step 3. Click the Advanced tab and modify the desired properties.

The IPoIB network interface is automatically restarted once you finish modifying IPoIB parameters. Consequently, it might affect any running traffic.

4.1.3 Displaying Adapter Related Information

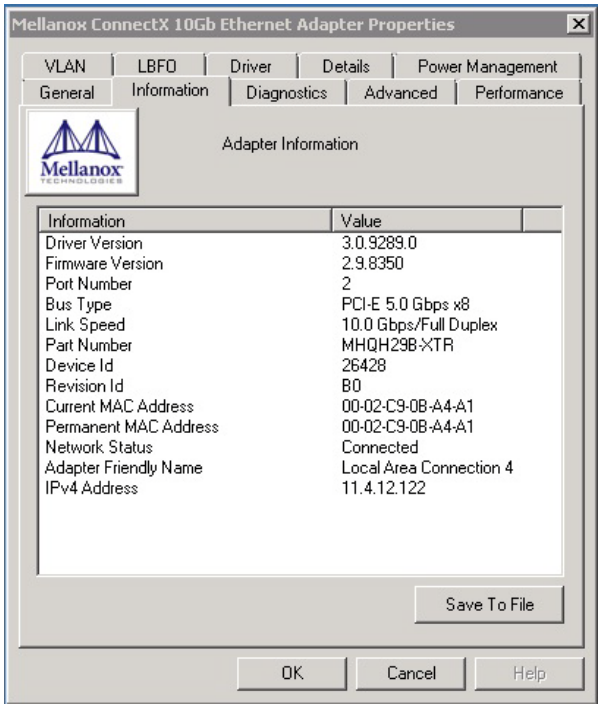
To display a summary of network adapter software, firmware- and hardware-related information such as driver version, firmware version, bus interface, adapter identity, and network port link information, perform the following steps:

Step 1 Display the Device Manager.



Step 2. Right-click a Mellanox ConnectX VPI adapter (under “System devices” list) and left-click Properties.

Step 3. Select the Information tab from the Properties sheet.

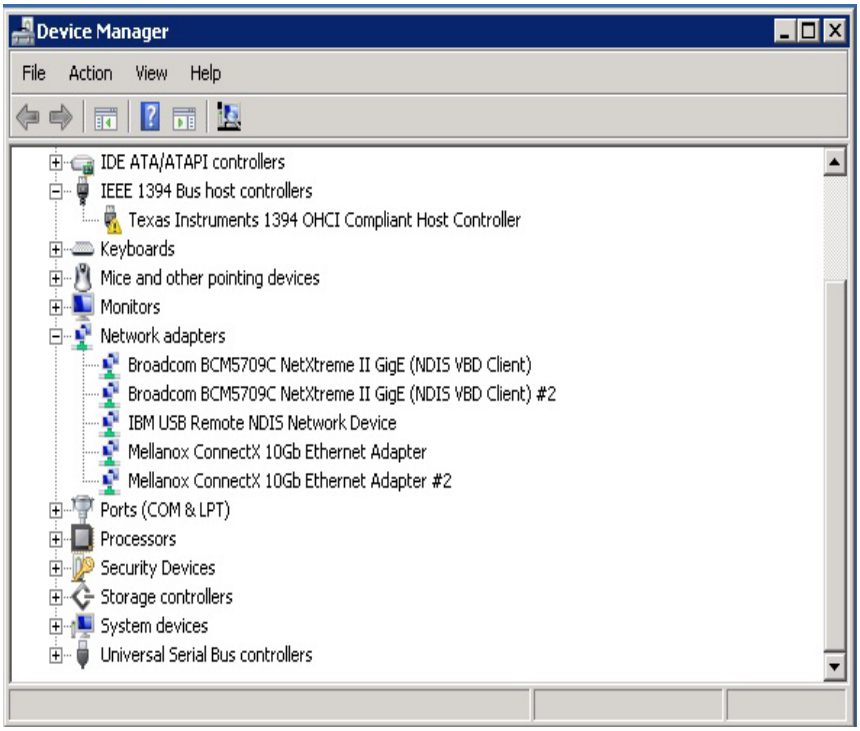


To save this information for debug purposes, click **Save To File** and provide the output file name.

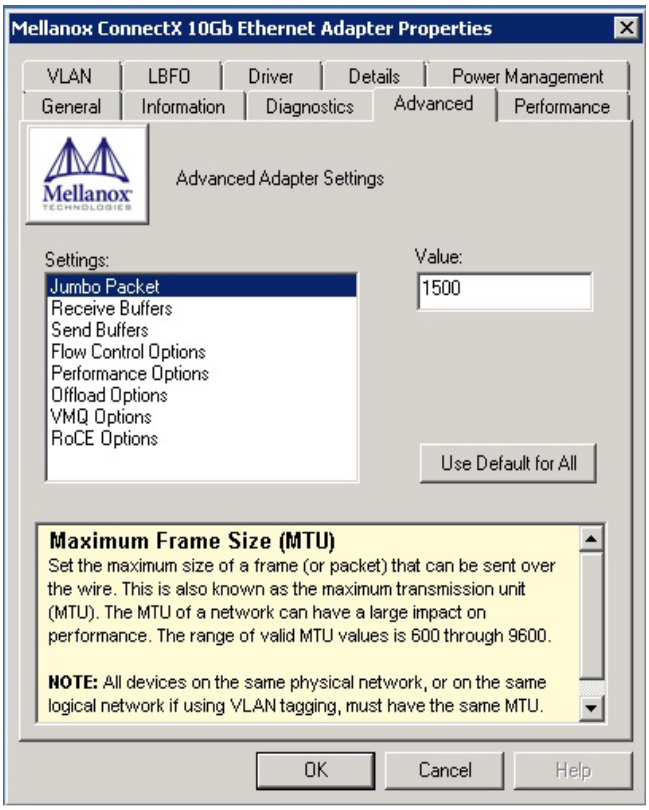
4.2 Configuring the Ethernet Driver

The following steps describe how to configure advanced features.

Step 1 Display the Device Manager.



Step 2. Right-click a Mellanox network adapter (under "Network adapters" list) and left-click Properties. Select the Advanced tab from the Properties sheet.



Step 3. Modify configuration parameters to suit your system.

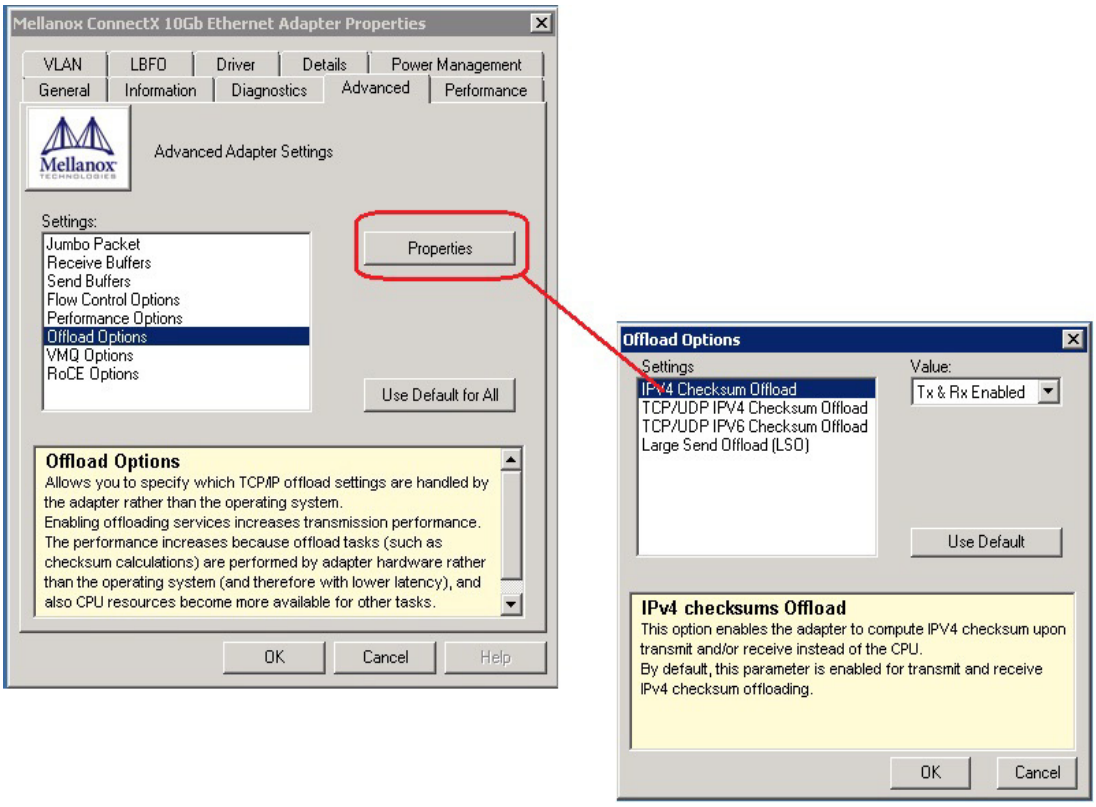
Please note the following:

a. For help on a specific parameter/option, check the help window at the bottom of the dialog.

b. If you select one of the entries Offload Options, Performance Options, or Flow Control Options, you'll need to click the Properties button to modify parameters via a pop-up dialog. See example in the two figures below.

c. A "Use Default for All" button appears on the Advanced dialog. Click this button to set all entries (and their sub-entries) to the Mellanox Ethernet driver default values. You will be prompted to approve this action.

d.If you press Cancel, then the last settings will be restored.



5 Performance

5.1 General Performance Optimization and Tuning

To achieve the best performance for Windows using 10GigE adapters, you may need to modify some of the Windows registries.

5.1.1 Registry Tuning

The registry entries that may be added/changed by this “General Tuning” procedure are:

Under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters:

- Disable TCP selective acks option for better cpu utilization:

```
SackOpts, type REG_DWORD, value set to 0.
```

Under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AFD\Parameters:

- Enable fast datagram sending for UDP traffic:

```
FastSendDatagramThreshold,  type REG_DWORD, value set to 64K.
```

Under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Ndis\Parameters:

- Set RSS parameters:

```
RssBaseCpu, type REG_DWORD, value set to 1.
```

5.1.2 Enable RSS

Enabling Receive Side Scaling (RSS) is performed by means of the following command:

```
“netsh int tcp set global rss = enabled”
```

5.1.3 Tuning the Network Adapter

The Network Adapter tuning can be performed either during installation by modifying some of Windows registries as explained in section“Registry Tuning” on page 32. or can be set post-installation manually. To improve the network adapter 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. Select Mellanox Ethernet adapter, right click and select Properties.
4. Select the “Performance tab”.
5. Click on “General Tuning” button.

Clicking the “General Tuning” button will change several registry entries (described below), and will check for system services that may decrease network performance. It will also generate a log including the applied changes.

Users can view this log to restore the previous values. The log path is:

```
%HOMEDRIVE%\Windows\System32\LogFiles\PerformanceTunning.log
```

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

Please note that a reboot may be required for the changes to take effect.

5.2 Application Specific Optimization and Tuning

5.2.1 Ethernet Performance Tuning

The user can configure the Ethernet adapter by setting some registry keys. The registry keys may affect Ethernet performance.

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 Ethernet adapter and select Properties.
4. Select the "Advanced" tab and select Performance Options
5. Modify performance parameters (properties) as desired.

5.2.1.1 Performance Known 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.

5.2.2 IPoIB Performance Tuning

The user can configure the IPoIB adapter by setting some registry keys. The registry keys may affect IPoIB performance.

For the complete list of registry entries that may be added/changed by the performance tuning procedure, see the IPoIB_registry_values.pdf file.

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.

5.3 Tunable Performance Parameters

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

- **Jumbo Packet**

  The maximum available size of the transfer unit, also known as the Maximum Transmission Unit (MTU). For IPoIB, the MTU should not include the size of the 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. The MTU of a network can have a substantial impact on performance. A 4K MTU size improves performance for short messages, since NDIS can coalesce a small message into a larger one.

  Valid MTU values range is between 600 and 9600.

  All devices on the same physical network, or on the same logical network, must have the same MTU.

- **Receive Buffers**

  The number of receive buffers (default 1024).

- **Send Buffers**

  The number of sent buffers (default 2048).

- **Performance Options**

  Configures parameters that can improve adapter performance.

  - Interrupt Moderation

    Moderates or delays the interrupts' generation. Hence, optimizes network throughput and CPU utilization (default Enabled).
    - When the interrupt moderation is enabled, the system accumulates interrupts and sends a single interrupt rather than a series of interrupts. An interrupt is generated after receiving 5 packets or after 10ms from the first packet received. It improves performance and reduces CPU load however, it increases latency.
    - When the interrupt moderation is disabled, the system generates an interrupt each time a packet is received or sent. In this mode, the CPU utilization data rates increase, as the system handles a larger number of interrupts. However, the latency decreases as the packet is handled faster.
  - Receive Side Scaling (RSS Mode)

    Improves incoming packet processing performance. RSS enables the adapter port to utilize the multiple CPUs in a multi-core system for receiving incoming packets and steering them to the designated destination. RSS can significantly improve the number of transactions, the number of connections per second, and the network throughput.
    This parameter can be set to one of the following values:

- Enabled (default): Set RSS Mode
- Disabled: The hardware is configured once to use the Toeplitz hash function, and the indirection table is never changed.

IOAT is not used while in RSS mode.

- Receive Completion Method

  Sets the completion methods of the received packets, and can affect network throughput and CPU utilization.
  - **Polling Method**

Increases the CPU utilization as the system polls the received rings for the incoming packets. However, it may increase the network performance as the incoming packet is handled faster.
  - **Interrupt Method**

Optimizes the CPU as it uses interrupts for handling incoming messages. However, in certain scenarios it can decrease the network throughput.
  - **Adaptive (Default Settings)**

A combination of the interrupt and polling methods dynamically, depending on traffic type and network usage. Choosing a different setting may improve network and/or system performance in certain configurations.

- Interrupt Moderation RX Packet Count

  Number of packets that need to be received before an interrupt is generated on the receive side (default 5).
- Interrupt Moderation RX Packet Time

  Maximum elapsed time (in usec) between the receiving of a packet and the generation of an interrupt, even if the moderation count has not been reached (default 10).
- Rx Interrupt Moderation Type

  Sets the rate at which the controller moderates or delays the generation of interrupts making it possible to optimize network throughput and CPU utilization. The default setting (Adaptive) adjusts the interrupt rates dynamically depending on the traffic type and network usage. Choosing a different setting may improve network and system performance in certain configurations.
- Send completion method

  Sets the completion methods of the Send packets and it may affect network throughput and CPU utilization.

- Interrupt Moderation TX Packet Count

  Number of packets that need to be sent before an interrupt is generated on the send side (default 0).
- Interrupt Moderation TX Packet Time

  Maximum elapsed time (in usec) between the sending of a packet and the generation of an interrupt even if the moderation count has not been reached (default 0).
- Bus-master DMA Operations

  Sets the addressing type: NDIS DMA addressing (UseDma=Enabled) or physical addressing (UseDma=Disabled) (default Disabled).

- **Offload Options**

  Allows you to specify which TCP/IP offload settings are handled by the adapter rather than the operating system.

  Enabling offloading services increases transmission performance as the offload tasks are performed by the adapter hardware rather than the operating system. Thus, freeing CPU resources to work on other tasks.

  - IPv4 Checksums Offload

    Enables the adapter to compute IPv4 checksum upon transmit and/or receive instead of the CPU (default Enabled).
  - TCP/UDP Checksum Offload for IPv4 packets

    Enables the adapter to compute TCP/UDP checksum over IPv4 packets upon transmit and/or receive instead of the CPU (default Enabled).
  - TCP/UDP Checksum Offload for IPv6 packets

    Enables the adapter to compute TCP/UDP checksum over IPv6 packets upon transmit and/or receive instead of the CPU (default Enabled).
  - Large Send Offload (LSO)

    Allows the TCP stack to build a TCP message up to 64KB long and sends it in one call down the stack. The adapter then re-segments the message into multiple TCP packets for transmission on the wire with each pack sized according to the MTU. This option offloads a large amount of kernel processing time from the host CPU to the adapter.

- **IB Options**

  Configures parameters related to InfiniBand functionality.

  - SA Query Retry Count

    Sets the number of SA query retries once a query fails. The valid values are 1 - 64 (default 10).
  - SA Query Timeout

    Sets the waiting timeout (in millisecond) of an SA query completion. The valid values are 500 - 60000 (default 1000 ms).

This document describes how to modify Windows registry parameters in order to improve performance.

Please note that modifying the registry incorrectly might lead to serious problems, including the loss of data, system hang, and you may need to reinstall Windows. As such it is recommended to back up the registry on your system before implementing recommendations included in this document. If the modifications you apply lead to serious problems, you will be able to restore the original registry state. For more details about backing up and restoring the registry, please visit www.microsoft.com.

6 OpenSM - Subnet Manager

OpenSM v3.3.11 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.

Please use the embedded OpenSM in the WinOF package for testing purpose and small cluster. Otherwise, we recommend using OpenSM from FabricIT EFM™ or UFM™.

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.

To start OpenSM automatically, please perform the following:

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 more than a single instance of OpenSM in the subnet.
- IBDiagnet cannot run on the same IB port that OpenSM is running on.

7 InfiniBand Fabric

7.1 Network Direct Interface

The Network Direct Interface (NDI) architecture provides application developers with a networking interface that enables zero-copy data transfers between applications, kernel-bypass I/O generation and completion processing, and one-sided data transfer operations.

NDI is supported by Microsoft and is the recommended method to write InfiniBand application. NDI exposes the advanced capabilities of the Mellanox networking devices and allows applications to leverage advances of InfiniBand.

For further information please refer to:

http://msdn.microsoft.com/en-us/library/cc904397(v=vs.85).aspx

7.2 InfiniBand Fabric Diagnostic Utilities

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:

7.2.1 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.

7.2.1.1 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

7.2.1.2 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

7.2.1.3 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.

7.2.1.4 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>]
```

7.2.1.5 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

7.2.1.6 Output Files

Table 5 - 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

Table 5 - ibdiagnet (of ibutils) Output Files

Output File	Description
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.

7.2.1.7 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
```

7.2.1.8 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>]
```

7.2.1.9 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	

7.2.1.10 Output Files

Table 6 - 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

7.2.1.11 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
```

7.2.2 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

7.2.2.1 Applicable Hardware

All InfiniBand devices.

7.2.2.2 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>]]
```

7.2.2.3 Options

The table below lists the various flags of the command.

Table 7 - 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)

Table 7 - ibportstate Flags and Options (Continued)

Flag	Description
-e/--errors	Show send and receive errors (timeouts and others)
-v/--verbose	Increase verbosity level. May be used several times for additional verbosity (-vvv or -v -v -v)
-V/--version	Show version info
-D/--Direct	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/--Guid	Use GUID address argument. In most cases, it is the Port GUID. Example: '0x08f1040023'
-s/--sm_port	Use <smlid> as the target lid for SM/SA queries
-C/--Ca	Use the specified channel adapter or router
-P/--Port	Use the specified port
-u/--usage	Usage message
-t/--timeout	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
```

7.2.3 ibroute

Uses SMPs to display the forwarding tables for 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 of 1 to FDBTop.

7.2.3.1 Applicable Hardware

InfiniBand switches.

7.2.3.2 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> [<star-
tlid> [<endlid>]]]
```

7.2.3.3 Options

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

Table 8 - ibroute 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)
-a/--all	Show all LIDs in range, including invalid entries
-v/--verbose	Increase verbosity level. May be used several times for additional verbosity (-vvv or -v -v -v)
-V/--version	Show version info
-n/--no_dests	Do not try to resolve destinations
-D/--Direct	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/--Guid	Use GUID address argument. In most cases, it is the Port GUID. Example: ‘0x08f1040023’
-M/--Multicast	Show multicast forwarding tables. The parameters <startlid> and <endlid> specify the MLID range.
-L/--Lid	Use Lid address argument
-u/--usage	Usage message
-e/--errors	Show send and receive errors (timeouts and others)
-s/--sm_port <smlid>	Use <smlid> as the target LID for SM/SA queries
-C/--Ca <ca_name>	Use the specified channel adapter or router
-P/--Port <ca_port>	Use the specified port

Table 8 - ibroute Flags and Options

Flag	Description
-t/--timeout<timeout_ms>	Override the default timeout for the solicited MADs [msec]
<dest dr_path | lid | guid>	Destination's directed path, LID, or GUID
<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 Technolo-
gies')

0x0003 021 : (Switch portguid 0x000b8cffff004016: 'MT47396 Infiniscale-III Mellanox Technolo-
gies')

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 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 Technolo-
gies')

0x0006 007 : (Channel Adapter portguid 0x0002c90300001039: 'sw137 HCA-1')

0x0007 021 : (Channel Adapter portguid 0x0002c9020025874a: 'sw157 HCA-1')

3 valid lids dumped
```

3. 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 Technolo-
gies')

0x0003 000 : (Switch portguid 0x000b8cffff004016: 'MT47396 Infiniscale-III Mellanox Technolo-
gies')

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
```

4. 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 Technologies):

           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
```

7.2.4 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.

7.2.4.1 Applicable Hardware

All InfiniBand devices.

7.2.4.2 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]
```

7.2.4.3 Options

The table below lists the various flags of the command.

Table 9 - smpquery 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/--errors	Show send and receive errors (timeouts and others)
-v/--verbose	Increase verbosity level. May be used several times for additional verbosity (-vvv or -v -v -v)
-D/--Direct	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/--Guid	Use GUID address argument. In most cases, it is the Port GUID. Example: ‘0x08f1040023’
-s/--sm_port <smlid>	Use <smlid> as the target LID for SM/SA queries
-V/--version	Show version info
-L/--Lid	Use Lid address argument
-c/--combined	Use combined route address argument
-u/--usage	Usage message
-C/--Ca <ca_name>	Use the specified channel adapter or router
-P/--Port <ca_port>	Use the specified port
-t/--timeout <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
--node-name-map <file>	Node name map file
-x/--extended	Use extended speeds

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
FilterRawInb:.....0
FilterRawOutb:.....0
MkeyViolations:.....0
PkeyViolations:.....0
QkeyViolations:.....0
GuidCap:.....128
ClientReregister:.....0
SubnetTimeout:.....18
RespTimeVal:.....16
LocalPhysErr:.....8
OverrunErr:.....8
MaxCreditHint:.....0
RoundTrip:.....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
```

7.2.5 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.

7.2.5.1 Applicable Hardware

All InfiniBand devices.

7.2.5.2 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 10 - perfquery Flags and Options

Flag	Description
--help, -h	Print the help menu
--debug, -d	Raise the IB debug level. May be used several times for higher debug levels (-ddd or -d -d -d)
--Guid,-G	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
--all_ports, -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
--loop_ports, -l	Loop ports
--reset_after_read, -r	Reset the counters after reading them
--Ca, -C <ca_name>	Use the specified channel adapter or router
--Port, -P <ca_port>	Use the specified port
--Reset_only, -R	Reset the counters
--timeout, -t <timeout_ms>	Override the default timeout for the solicited MADs [msec]
--version, -V	Show version info
<lid | guid> [[port][reset_mask]]	LID or GUID
--extended, -x	show extended port counters
--extended_speeds, -T	show port extended speeds counters
--oprcvcounters	show Rcv Counters per Op code
--flowctlcounters	show flow control counters
--vloppackets	show packets received per Op code per VL
--vlopdata	show data received per Op code per VL
--vlxmitflowctlerrors	show flow control update errors per VL

Table 10 - perfquery Flags and Options

Flag	Description
--vlxmitcounters	show ticks waiting to transmit counters per VL
--swportvlcong	show sw port VL congestion
--rcvcc	show Rcv congestion control counters
--slrcvfecn	show SL Rcv FECN counters
--slrcvbecn	show SL Rcv BECN counters
--xmitcc	show Xmit congestion control counters
--vlxmittimecc	show VL Xmit Time congestion control counters

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
```

7.2.6 ibping

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

7.2.6.1 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>
```

7.2.6.2 Options

The table below lists the various flags of the command.

Table 11 - ibping Flags and Options

Flag	Description
--count, -c <num>	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
--Server, -S	Starts in server mode (do not return)
--debug, -d/-ddd/ -d -d -d	Raises the IB debugging level
--errors, -e	Shows send and receive errors (timeouts and others)

Table 11 - ibping Flags and Options

Flag	Description
--help, -h	Shows the usage message
--verbose, -v/-vvv/-v -v -v	Increases the application verbosity level
--version, -V	Shows the version info
--Lid, -L	Use LID address argument
--usage, -u	Usage message
--Guid, -G	Uses GUID address argument. In most cases, it is the Port GUID. For example: "0x08f1040023"
--sm_port, -s <smlid>	Uses 'smlid' as the target lid for SM/SA queries
--Ca, -C <ca_name>	Uses the specified ca_name
--Port, -P <ca_port>	Uses the specified ca_port
--timeout, -t <timeout_ms>	Overrides the default timeout for the solicited mads

7.2.7 ibnetdiscover

ibnetdiscover performs IB subnet discovery and outputs a readable topology file. GUIDs, node types, and port numbers are displayed as well as port LIDs and NodeDescriptions. All nodes (and links) are displayed (full topology). Optionally, this utility can be used to list the current connected nodes by node-type. The output is printed to standard output unless a topology file is specified.

7.2.7.1 Synopsys

```
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>]
```

7.2.7.2 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 - 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

Table 12 - ibnetdiscover Flags and Options

Flag	Description
-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 61.
--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
-p, --ports	Obtain a ports report which is a list of connected ports with relevant information (like LID, portnum, GUID, width, speed, and NodeDescription)
-m, --max_hops	Report max hops discovered
--debug, -d/-ddd/ -d -d -d	Raise the IB debugging level
--errors, -e	Show send and receive errors (timeouts and others)
--help, -h	Show the usage message
--verbose,-v/-vv/ -v -v -v	Increase the application verbosity level
--version, -V	Show the version info
--outstanding_smps -o <val>	Specify the number of outstanding SMPs which should be issued during the scan
-usage, -u	Usage message
--Ca, -C <ca_name>	Use the specified ca_name
--Port, -P <ca_port>	Use the specified ca_port
--timeout, -t <timeout_ms>	Override the default timeout for the solicited mads
--full, -f	show full information (ports' speed and width)
--show, -s	show more information

7.2.7.3 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 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
```

Non-Chassis Nodes

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.

```
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 Mellanox Technolo-
gies" lid 4 4xSDR

[10]    "S-0008f10400410015"[1]         # "SW-6IB4 Voltaire" lid 3 4xSDR

[8]     "H-0008f10403960558"[2](8f1040396055a)         # "MT23108 InfiniHost Mellanox Technolo-
gies" lid 14 4xSDR

[6]     "S-0008f10400410015"[3]         # "SW-6IB4 Voltaire" lid 3 4xSDR

[12]    "H-0008f10403960558"[1](8f10403960559)         # "MT23108 InfiniHost Mellanox Technolo-
gies" 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 Mellanox Technolo-
gies" lid 16 4xSDR

[4]     "H-005442b100004900"[1](5442b100004901)        # "MT23108 InfiniHost Mellanox Technolo-
gies" 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 Voltaire" 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 Voltaire" 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 Voltaire" lid 6
4xSDR

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

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"
```

7.2.8 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.

7.2.8.1 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>]]
```

7.2.8.2 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 13 - ibtracert Flags and Options

Flag	Description
--force, -f	Force
-n, --no_info	Simple format; do not show additional information
--mlid, -m <mlid>	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 61.
--debug, -d/-ddd/-d -d -d	Raise the IB debugging level
--Lid, -L	Use LID address argument
--errors, -e	Show send and receive errors
--usage, -u	Usage message
--Guid, -G	Use GUID address argument. In most cases, it is the Port GUID. Example: "0x08f1040023"
--sm_port, -s <smlid>	Use 'smlid' as the target lid for SM/SA queries
--help, -h	Show the usage message
-verbose, -v/-vv/-v -v -v	Increase the application verbosity level
--version, -V	Show the version info
--Ca, -C <ca_name>	Use the specified ca_name
--Port, -P <ca_port>	Use the specified ca_port
--timeout, -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
```

7.2.9 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 result in a malfunction of the target SM.

7.2.9.1 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]
```

7.2.9.2 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 - sminfo Flags and Options

Flag	Description
--state, -s	Set SM state • 0 - not active • 1 - discovering • 2 - standby • 3 - master
--priority, -p	Set priority (0-15)
--activity, -a	Set activity count
--debug, -d/-ddd/-d -d -d	Raise the IB debugging level
--Direct, -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
--errors, -e	Show send and receive errors (timeouts and others)
--Guid, -G	Use GUID address argument. In most cases, it is the Port GUID. Example: "0x08f1040023"
--help, -h	Show the usage message
-verbose, -v/-vv/-v -v -v	Increase the application verbosity level
--version, -V	Show the version info
--Ca, -C <ca_name>	Use the specified ca_name
--Port, -P <ca_port>	Use the specified ca_port
--timeout, -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
```

7.2.10 ibclearerrors

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

7.2.10.1 Synopsys

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

7.2.10.2 Options

The table below lists the various flags of the command.

Table 15 - ibclearerrors Flags and Options

Flag	Description
-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

7.2.11 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.

7.2.11.1 Synopsys

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

7.2.11.2 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 16 - ibstat Flags and Options

Flag	Description
-l, --list_of_cas	List all IB devices
-s, --short	Short output

Table 16 - ibstat Flags and Options

Flag	Description
-p, --port_list	Show port list
ca_name	InfiniBand device name
portnum	Port number of InfiniBand device
--debug, -d/-ddd/-d -d -d	Raise the IB debugging level
--help, -h	Show the usage message
-verbose, -v/-vv/-v -v -v	Increase the application verbosity level
--version, -V	Show the version info
--usage, -u	usage message

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'
```

7.2.12 vstat

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

7.2.12.1 Synopsys

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

7.2.12.2 Options

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

Table 17 - ibstat Flags and Options

Flag	Description
-v -	Verbose mode
-c	HCA error/statistic counters
-m	more verbose mode
-p N	repeat every N sec

7.2.13 part_man

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

7.2.13.1 Synopsys

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

7.2.13.2 Options

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

Table 18 - part_man 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: • part_man add <port guid> <pkey |> <peky> Port guid is in the format of : • xxxx:xxxx:xxxx:xxxx Pkey format: • 0x8xxx or 8xxx
rem	Removes partition key of the specified port. The output format is: part_man.exe rem <port_guid> <pkey1> <pkey2>

7.2.14 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.

7.2.14.1 Synopsys

```
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)]
```

7.2.14.2 Options

The table below lists the various flags of the command.

Table 19 - 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)
-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

Table 19 - osmtest Flags and Options

Flag	Description
-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	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 stdout.
-v	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.
-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.

7.3 InfiniBand Fabric Performance Utilities

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

7.3.1 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).

7.3.1.1 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)]
```

7.3.1.2 Options

The table below lists the various flags of the command.

Table 20 - 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)
-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
-g, --grh	Use GRH with packets (mandatory for RoCE)

7.3.2 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).

7.3.2.1 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]
```

7.3.2.2 Options

The table below lists the various flags of the command.

Table 21 - 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)
-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
-g, --grh	Use GRH with packets (mandatory for RoCE)

7.3.3 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.

7.3.3.1 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)]
```

7.3.3.2 Options

The table below lists the various flags of the command.

Table 22 - 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
-g, --grh	Use GRH with packets (mandatory for RoCE)

7.3.4 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.

7.3.4.1 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]
```

7.3.4.2 Options

The table below lists the various flags of the command.

Table 23 - 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
-g, --grh	Use GRH with packets (mandatory for RoCE)

7.3.5 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.

7.3.5.1 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(idirec-
tional)]                                                          [-a(ll)] [-V(ersion)]
```

7.3.5.2 Options

The table below lists the various flags of the command.

Table 24 - 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
-o, --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)
-g, --grh	Use GRH with packets (mandatory for RoCE)

7.3.6 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.

7.3.6.1 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]
```

7.3.6.2 Options

The table below lists the various flags of the command.

Table 25 - 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
-g, --grh	Use GRH with packets (mandatory for RoCE)

7.3.7 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).

7.3.7.1 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)]
```

7.3.7.2 Options

The table below lists the various flags of the command.

Table 26 - 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
-R, --rdma_cm	Connect QPs with rdma_cm and run test on those QPs
-z, --com_rdma_cm	Communicate with rdma_cm module to exchange data - use regular QPs
-c, --connection=<RC/UC/UD>	Connection type RC/UC/UD (default RC)
-I, --inline_size=<size>	Max size of message to be sent in inline (default 0)
-Q, --cq-mod	Generate Cqe only after <--cq-mod> completion
-N, --no peak-bw	Cancel peak-bw calculation (default with peak)

7.3.8 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).

7.3.8.1 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]
```

7.3.8.2 Options

The table below lists the various flags of the command.

Table 27 - 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
-R, --rdma_cm	Connect QPs with rdma_cm and run test on those QPs
-z, --com_rdma_cm	Communicate with rdma_cm module to exchange data - use regular QPs
-c, --connection=<RC/UC/UD>	Connection type RC/UC/UD (default RC)
-I, --inline_size=<size>	Max size of message to be sent in inline (default 400)

7.3.9 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.

7.3.9.1 Synopsys

```
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(idirectional)] [-a(ll)] [-V(ersion)]
```

7.3.9.2 Options

The table below lists the various flags of the command.

Table 28 - 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)

Table 28 - ibv_send_bw Flags and Options

Flag	Description
-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]
-R, --rdma_cm	Connect QPs with rdma_cm and run test on those QPs
-z, --com_rdma_cm	Communicate with rdma_cm module to exchange data - use regular QPs
-Q, --cq-mod	Generate Cqe only after <--cq-mod> completion

7.3.10 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.

7.3.10.1 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]
```

7.3.10.2 Options

The table below lists the various flags of the command.

Table 29 - 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)
-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

Table 29 - ibv_send_lat Flags and Options

Flag	Description
-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
-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', where X is a vlaue within [0,255]. You must specify a different MGID on both sides to avoid loopback.
-R, --rdma_cm	Connect QPs with rdma_cm and run test on those QPs
-z, --com_rdma_cm	Communicate with rdma_cm module to exchange data - use regular QPs

7.3.11 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.

7.3.11.1 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)]
```

7.3.11.2 Options

The table below lists the various flags of the command.

Table 30 - ibv_write_bw Flags and Options

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

Table 30 - 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)
-R, --rdma_cm	Connect QPs with rdma_cm and run test on those QPs
-z, --com_rdma_cm	Communicate with rdma_cm module to exchange data - use regular QPs
-Q, --cq-mod	Generate Cqe only after <--cq-mod> completion

7.3.12 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.

7.3.12.1 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]
```

7.3.12.2 Options

The table below lists the various flags of the command.

Table 31 - 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)
-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)
-R, --rdma_cm	Connect QPs with rdma_cm and run test on those QPs
-z, --com_rdma_cm	Communicate with rdma_cm module to exchange data - use regular QPs

8 Software Development Kit

Software Development Kit (SDK) a set of development tools that allows the creation of InfiniBand applications for MLNX_VPI software package.

The SDK package contains, header files, libraries, and code examples. To open the SDK package you must run the sdk.exe file and get the complete list of files. SDK package can be found under <installation_directory>\IB\SDK

9 Troubleshooting

9.1 InfiniBand Troubleshooting

Issue # 1: The IB interfaces is not up after the first reboot after the installation process is completed.

Suggestion: To troubleshoot this issue, follow the steps bellow:

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).
2. 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")
   - PORT_ARMED - when the port is connected to some other port ("physical link");
3. Run OpenSM - see OpenSM operation instructions in the OpenSM section above.
4. Verify the status of ports by using vstat: All connected ports should report "PORT_ACTIVE" state.

9.2 Ethernet Troubleshooting

Issue # 1: The installation of MLNX_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 does no 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. 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:

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

   OR

1. Click start-->Run and enter "eventvwr.exe".
2. 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>.

10 Documentation

- Under <installation_directory>\Documentation:
  - License file
  - User Manual (this document)
  - MLNX_VPI_Installation Guide
  - MLNX_VPI_Release Notes