SOFTWARE MANUAL

Mellanox WinOF VPI

User Manual

Rev 5.35

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 PRODUCT(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

Table of Contents

Document Revision History 9

About this Manual 16

Chapter 1 Introduction 19

1.1 Supplied Packages 20
1.2 WinOF Set of Documentation 20
1.3 Windows MPI (MS-MPI) 20
1.4 Windows Nano Server 2016 20

Chapter 2 Installation 22

2.1 Hardware and Software Requirements 22
2.2 Downloading Mellanox WinOF Driver. 22
2.3 Installing Mellanox WinOF Driver 23
2.3.1 Attended Installation 23
2.3.2 Unattended Installation 29
2.4 Installation Results. 30
2.5 Extracting Files Without Running Installation 31
2.6 Uninstalling Mellanox WinOF Driver. 34
2.6.1 Attended Uninstallation 34
2.6.2 Unattended Uninstallation. 34
2.7 Firmware Upgrade 34
2.8 Upgrading Mellanox WinOF Driver 34
2.9 Booting Windows from an iSCSI Target or PXE 35
2.9.1 Configuring the WDS, DHCP and iSCSI Servers 35
2.9.2 Configuring the Client Machine 36
2.9.3 Installing OS 36
2.10 Deploying the Driver on a Nano Server 38
2.10.1 Offline Installation 38
2.10.2 Online Update 39

Chapter 3 Features Overview and Configuration 41

3.1 Ethernet Network. 41
3.1.1 Port Configuration. 41
3.1.2 Assigning Port IP After Installation 44
3.1.3 56GbE Link Speed 46
3.1.4 RDMA over Converged Ethernet (RoCE) 47
3.1.5 Teaming and VLAN 54
3.1.6 Command Line Based Teaming Configuration 65
3.1.7 Header Data Split 69
3.1.8 Configuring Quality of Service (QoS) 69
3.1.9 Configuring the Ethernet Driver 73
3.1.10 Differentiated Services Code Point (DSCP) 75

3.1.11 Lossless TCP 79
3.1.12 Receive Side Scaling (RSS) 82
3.1.13 Ignore Frame Check Sequence (FCS) Errors 82
3.2 InfiniBand Network 83
3.2.1 Port Configuration. 83
3.2.2 OpenSM - Subnet Manager. 83
3.2.3 Modifying IPoIB Configuration 84
3.2.4 Displaying Adapter Related Information 84
3.2.5 Assigning Port IP After Installation 86
3.2.6 Receive Side Scaling (RSS) 86
3.2.7 Multiple Interfaces over non-default PKeys Support 86
3.2.8 Teaming. 88
3.3 Management 91
3.3.1 PowerShell Configuration. 91
3.4 Storage Protocols 92
3.4.1 Deploying Windows Server 2012 and Above with SMB Direct 92
3.5 Virtualization 94
3.5.1 Virtual Ethernet Adapter 94
3.5.2 Hyper-V with VMQ 95
3.5.3 Network Virtualization using Generic Routing Encapsulation (NVGRE) 96
3.5.4 Single Root I/O Virtualization (SR-IOV) 100
3.5.5 Virtual Machine Multiple Queue (VMMQ) 120
3.5.6 Network Direct Kernel Provider Interface 122
3.5.7 PacketDirect Provider Interface 126
3.6 Configuration Using Registry Keys. 129
3.6.1 Finding the Index Value of the HCA 130
3.6.2 Finding the Index Value of the Network Interface 130
3.6.3 Basic Registry Keys 132
3.6.4 Off-load Registry Keys 134
3.6.5 Performance Registry Keys. 136
3.6.6 Ethernet Registry Keys 142
3.6.7 IPolB Registry Keys 146
3.6.8 General Registry Values 148
3.6.9 MLX BUS Registry Keys 149
3.7 Software Development Kit (SDK). 152
3.7.1 Network Direct Interface. 153
3.7.2 Win-Linux nd_rping Test 153
3.8 Performance Tuning and Counters 154
3.8.1 General Performance Optimization and Tuning 155
3.8.2 Application Specific Optimization and Tuning 163
3.8.3 Tunable Performance Parameters 164
3.8.4 Adapter Proprietary Performance Counters 167
3.8.5 Device Proprietary Counters. 175
3.9 System Recovery upon Error Detection 178
3.10 NIC Resiliency. 179

3.11 RDMA Configuration 179
3.11.1 Memory Translation Table (MTT) Optimization 179
Chapter 4 Utilities 180
4.1 Snapshot Tool 180
4.1.1 Snapshot Usage 180
4.2 part_man - Virtual IPoIB Port Creation Utility 180
4.3 vea_man- Virtual Ethernet. 182
4.3.1 Adding a New Virtual Adapter 182
4.3.2 Removing a Virtual Ethernet Adapter. 183
4.3.3 Querying the Virtual Ethernet Database 183
4.3.4 Help Message 183
4.4 InfiniBand Fabric Diagnostic Utilities. 183
4.4.1 Utilities Usage: Common Configuration, Interface and Addressing 183
4.5 Fabric Performance Utilities 187
4.6 mlxtool. 188
4.6.1 mlxtool Help 189
4.6.2 dbg sub-tool. 190
4.6.3 show 195
4.6.4 modify Tool. 196
Chapter 5 Troubleshooting 197
5.1 Installation Related Troubleshooting. 198
5.1.1 Installation Error Codes and Troubleshooting 198
5.2 InfiniBand Related Troubleshooting 200
5.3 Ethernet Related Troubleshooting 200
5.4 Performance Related Troubleshooting 202
5.4.1 General Diagnostic 203
5.5 Virtualization Related Troubleshooting 204
5.6 Reported Driver Events 205
5.7 Extracting WPP Traces 206
5.8 State Dumping 207
Appendix A NVGRE Configuration Scripts Examples 209
A.1 Adding NVGRE Configuration to Host 14 Example 209
A.2 Adding NVGRE Configuration to Host 15 Example 210
Appendix B Windows MPI (MS-MPI) 212
B.1 Overview 212
B.2 System Requirements 212
B.3 Running MPI 212
B.4 Directing MSMPI Traffic. 212
B.5 Running MSMPI on the Desired Priority. 213
B.6 Configuring MPI 213
B.7 PFC Example. 213
B.8 Running MPI Command Examples. 214

List of Tables

Table 1: Document Revision History 9
Table 2: Documentation Conventions 16
Table 3: Abbreviations and Acronyms 17
Table 4: Related Documents 18
Table 5: Hardware and Software Requirements 22
Table 6: Reserved IP Address Options 36
Table 7: DSCP Registry Keys Settings 77
Table 8: DSCP Default Registry Keys Settings. 78
Table 9: Enabling Lossless TCP Using The Registry Key DelayDropTimeout 81
Table 10: Lossless TCP Associated Events. 81
Table 11: Registry Keys Setting 82
Table 12: Recommended Values 111
Table 13: SR-IOV Mode Configuration Parameters 113
Table 14: Default Ethertype Values. 118
Table 15: Ethertype Spoof Protection Registry Keys 119
Table 16: Registry Key Groups 130
Table 17: Basic Registry Keys. 132
Table 18: Off-load Registry Keys 134
Table 19: Performance Registry Keys. 136
Table 20: Ethernet Registry Keys 142
Table 21: Flow Control Options 144
Table 22: VMQ Options 145
Table 23: IPoIB Registry Keys 146
Table 24: General Registry Values 148
Table 25: SR-IOV Registry Keys 149
Table 26: RoCE Options 151
Table 27: NIC Resiliency Registry Keys. 152
Table 28: General Registry Keys. 152
Table 29: Performance Tuning Tool Application Options 159
Table 30: Mellanox Adapter Traffic Counters 167
Table 31: Mellanox Adapter Diagnostics Counters. 169
Table 32: Mellanox Qos Counters 171
Table 33: Mellanox Adapter RSS Counters 173
Table 34: RDMA Activity 175

Table 35: Device Proprietary Counters 176
Table 36: Mellanox Proprietary WinOF Bus Counters 177
Table 37: RDMA Activity 178
Table 38: Virtual IPoIB Port Creation Utility 181
Table 39: Diagnostic Utilities 185
Table 40: Fabric Performance Utilities 187
Table 41: Installation Related Issues 198
Table 42: Setup Return Codes 198
Table 43: Firmware Burning Warning Codes 199
Table 44: Restore Configuration Warnings 199
Table 45: InfiniBand Related Issues 200
Table 46: Ethernet Related Issues 200
Table 47: Performance Related Issues 202
Table 48: Virtualization Related Issues 204
Table 49: Events Causing Automatic State Dumps 207

List of Figures

Figure 1: Installation Results 31
Figure 2: Network Service Boot in ISCSi 37
Figure 3: Network Service Boot in PXE 37
Figure 4: RoCE and RoCE v2 Frame Format Differences 50
Figure 5: RoCE Protocol Stack 51
Figure 6: Lossless TCP 79
Figure 7: NVGRE Packet Structure 97
Figure 8: Operating System Supports SR-IOV 107
Figure 9: SR-IOV Support 107
Figure 10: Hyper-V Manager 108
Figure 11: Connect Virtual Hard Disk 109
Figure 12: System Event Log 114
Figure 13: Virtual Switch with SR-IOV 115
Figure 14: Adding a VMNIC to a Mellanox V-switch 116
Figure 15: Enable SR-IOV on VMNIC 117
Figure 16: Virtual Function in the VM 118

Document Revision History

Table 1 - Document Revision History

Document Revision	Date	Changes
Rev 5.35	January 29, 2017	- Added the following sections: - Section 3.8.4.1.4, “Propriety Mellanox Adapter RSS Counters”, on page 172 (and its subsections) - Section 4.6.2.5, “Res”, on page 193 - Section 4.6.2.6, “Ipoib-ep”, on page 194 - Section 4.6.2.7, “Get-state”, on page 195 - Section 4.6.2.8, “Restart”, on page 195 - Updated the following sections: - Section 3.5.4, “Single Root I/O Virtualization (SR-IOV)”, on page 100 - Section 3.5.4.1.1, “System Requirements”, on page 100 - Section 3.5.4.1.2, “Feature Limitations”, on page 100 - Section 3.6.3, “Basic Registry Keys”, on page 132 - Section 3.6.4, “Off-load Registry Keys”, on page 134 - Section 3.6.5, “Performance Registry Keys”, on page 136 - Section 3.7, “Software Development Kit (SDK)”, on page 152 - Table 19, “Performance Registry Keys,” on page 136 - Table 31, “Mellanox Adapter Diagnostics Counters,” on page 169 - Table 45, “InfiniBand Related Issues,” on page 200
Rev 5.25	August 2016	- Added the following sections: - Section 2.10, “Deploying the Driver on a Nano Server”, on page 38 - Section 2.10.1, “Offline Installation”, on page 38 - Section 2.10.2, “Online Update”, on page 39 - Section 3.5.4.6, “Ethertype Spoof Protection”, on page 118 (and its subsections) - Section 3.5.6, “Network Direct Kernel Provider Interface”, on page 122 (and its subsections) - Section 3.5.7, “PacketDirect Provider Interface”, on page 126 (and its subsections) - Updated the following sections: - Section 3.6.4, “Off-load Registry Keys”, on page 134, added the “RssOnHostVPorts” Registry Key

Table 1 - Document Revision History

Document Revision	Date	Changes
Rev 5.22	June 2016	• Added the following sections: • Section 3.8.5, “Device Proprietary Counters”, on page 175 • Section 3.1.6, “Command Line Based Teaming Configuration”, on page 65 • Section 3.1.12, “Tx Limiter”, on page 88 • Updated the following sections: • Section 4.6, “mlxtool”, on page 188 • Section 5.5, “Virtualization Related Troubleshooting”, on page 204 • Section 2.9, “Booting Windows from an iSCSI Target or PXE”, on page 35 • Section 2.3, “Installing Mellanox WinOF Driver”, on page 23 • Section 4.6.4.1, “Modify Traffic Classes Bandwidth (BW) Limit Configuration Tool”, on page 196
Rev 5.10	September 2015	• Updated the following sections: • Section 4.6, “mlxtool”, on page 188 • Section 5.4.1, “General Diagnostic”, on page 203 • Section 3.6.9, “MLX BUS Registry Keys”, on page 149 • Added the following sections: • Section 3.6.9, “MLX BUS Registry Keys”, on page 149 • Section 5.8, “State Dumping”, on page 207 • Section 3.7.2, “Win-Linux nd_rping Test”, on page 153 • Section 3.1.4.3, “RoCE v2 UDP Port”, on page 51
Rev 4.95.50000	April 30, 2015	• Updated the following sections: • 3.7.1 “Network Direct Interface,” on page 153 • 3.2.8.1 “System Requirements,” on page 88 • Added the following sections: • 3.1.13 “Ignore Frame Check Sequence (FCS) Errors,” on page 82 • 3.8.1.1 “Mellanox Specific Extensions to the ND Interface,” on page 155 • Section 5.7, “Extracting WPP Traces”, on page 206 • Moved IPoIB content under Section 3.2, “InfiniBand Network”, on page 83

Table 1 - Document Revision History

Document Revision	Date	Changes
Rev 4.90.50000	January 15, 2015	• Restructured Section 5, "Troubleshooting", on page 197 • Added the following sections: • Section 3.3.1, "PowerShell Configuration", on page 91 • 3.9 "System Recovery upon Error Detection," on page 178 • Updated the following sections: • Section 2.3.2, "Unattended Installation", on page 32 • Section 2.6.2, "Unattended Uninstallation", on page 34 • Section 5.1, "Installation Related Troubleshooting", on page 198 • Section 5.3, "Ethernet Related Troubleshooting", on page 200 • Section 3.1.5, "Teaming and VLAN", on page 54 • Section 3.1.10, "Differentiated Services Code Point (DSCP)", on page 75 • Section 3.1.4.2, "RoCEv2", on page 49 • Section 4, "Utilities", on page 180 • Section 4.2, "part_man - Virtual IPoIB Port Creation Utility", on page 180
Rev 4.80.50000	August 30, 2014	Added the following sections: • Section 3.8.4.1.4, "Propriety Mellanox Adapter RSS Counters", on page 172 • Section 3.6.9, "MLX BUS Registry Keys", on page 149 • Section 4.1, "Snapshot Tool", on page 180 • Section 3.2.7, "Multiple Interfaces over non-default PKeys Support", on page 86 • Section 5.4.1, "General Diagnostic", on page 203 Updated the following sections: • Section 4.4, "InfiniBand Fabric Diagnostic Utilities", on page 183 • Section 4.5, "Fabric Performance Utilities", on page 187 • Section 4.2, "part_man - Virtual IPoIB Port Creation Utility", on page 180

Table 1 - Document Revision History

Document Revision	Date	Changes
Rev 4.70	May 4, 2014	Updated the following sections: • Section 1.2, "WinOF Set of Documentation", on page 20 • Section 2.7, "Firmware Upgrade", on page 34 • Section 3.5.4.4.2, "Enabling SR-IOV in Mellanox WinOF Package (Ethernet SR-IOV Only)", on page 112 • Section 3.4.1.2.1, "Verifying Network Adapter Configuration", on page 93 • Section 5.3, "Ethernet Related Troubleshooting", on page 200
Rev 4.70	May 4, 2014	Added the following sections: • Section 2.3, "Installing Mellanox WinOF Driver", on page 23 • Section 2.5, "Extracting Files Without Running Installation", on page 31 • Section 3.5.3.5, "Removing NVGRE configuration", on page 99 • Section 3.5.4, "Single Root I/O Virtualization (SR-IOV)", on page 100 • Section 3.5.1, "Virtual Ethernet Adapter", on page 94 • Section 3.5.4.2, "SR-IOV InfiniBand over KVM", on page 101 • Section 3.1.11, "Lossless TCP", on page 79 • Section 2.9, "Booting Windows from an iSCSI Target or PXE", on page 35 Removed the following sections: • Documentation

Table 1 - Document Revision History

Document Revision	Date	Changes
Rev 4.60	February 13, 2014	Updated the following sections: • Section 3.5.2, “Hyper-V with VMQ”, on page 95 • Section 3.5.3.3, “Enabling/Disabling NVGRE Offloading”, on page 97 Added the following sections: • Section 3.5.3.4, “Verifying the Encapsulation of the Traffic”, on page 99 • Section 3.5.1, “Virtual Ethernet Adapter”, on page 94
	December 30, 2013	Updated the following sections: • Section 3.1.4.1.2, “Configuring Windows Host”, on page 48 - Updated the example in Step 5 • Section 3.8.1.5.1, “Performance Tuning Tool Application”, on page 159 - Updated the Options table • Section 3.8.2, “Application Specific Optimization and Tuning”, on page 163 - Removed the “Bus-master DMA Operations” • Section 3.2.2, “OpenSM - Subnet Manager”, on page 83 - Added an option of how to register OpemSM via the PowerShell • Section 3.5.3.3.1, “Configuring the NVGRE using PowerShell”, on page 98
Rev 4.60	December 30, 2013	Added the following sections: • Section 3.1.8, “Configuring Quality of Service (QoS)”, on page 69 • Appendix A: “NVGRE Configuration Scripts Examples,” on page 209
Rev 4.55	December 15, 2013	Updated the following sections: • Section 3.1.5, “Teaming and VLAN”, on page 54 • Section 3.5.3.3.1, “Configuring the NVGRE using PowerShell”, on page 98
	November 07, 2013	Updated the following sections: • Section 3.1.4.1.2, “Configuring Windows Host”, on page 48
	October 03, 2013	Added support for Windows Server 2012 R2

Table 1 - Document Revision History

Document Revision	Date	Changes
Rev 4.40	July 17, 2013	Updated the following sections: • Section 3.1.4, “RDMA over Converged Ethernet (RoCE)”, on page 47 • Section 3.2.2, “OpenSM - Subnet Manager”, on page 83 • Section 5, “Troubleshooting”, on page 197 Added the following sections: Appendix A: “NVGRE Configuration Scripts Examples,” on page 209
	June 10, 2013	Updated the following sections: • Section 5, “Troubleshooting”, on page 197 • Section 1.2, “WinOF Set of Documentation”, on page 20 Added the following sections: • Section 3.8.4, “Adapter Proprietary Performance Counters”, on page 167
Rev 4.2	October 20, 2012	Added the following sections: • Section 3.4.1, “Deploying Windows Server 2012 and Above with SMB Direct”, on page 92, and its subsections • Section 3.1.6, “Command Line Based Teaming Configuration”, on page 65 • Section 4.2, “part_man - Virtual IPoIB Port Creation Utility”, on page 180 Updated Section 3.8, “Performance Tuning and Counters”, on page 154
Rev 3.2.0	July 23, 2012	• No changes
Rev 3.1.0	May 21, 2012	• Added section Tuning the IPoIB Network Adapter • Added section Tuning the Ethernet Network Adapter • Added section Performance tuning tool application • Removed section Tuning the Network Adapter • Removed section part_man • Removed section ibdiagnet

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

About this Manual

Scope

The document describes WinOF Rev 5.35 features, performance, diagnostic tools, content and configuration. Additionally, this document provides information on various performance tools supplied with this version.

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) ConnectX-3 and ConnectX-3 Pro adapter cards. It is also intended for application developers.

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.

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
NVGRE	Network Virtualization using Generic Routing Encapsulation
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
TC	Traffic Class

Related Documents

Table 4 - Related Documents

Document	Description
MFT User Manual	Describes the set of firmware management tools for a single InfiniBand node. MFT can be used for: • Generating a standard or customized Mellanox firmware image Querying for firmware information • Burning a firmware image to a single InfiniBand node • Enabling changing card configuration to support SR-IOV
WinOF Release Notes	For possible software issues, please refer to WinOF Release Notes.
MLNX OFED User Manual	For more information on SR-IOV over KVM, please refer to MLNX_OFED User Manual.
InfiniBand™ Architecture Specification, Volume 1, Release 1.2.1	The InfiniBand Specification by IBTA

1 Introduction

This User Manual describes installation, configuration and operation of Mellanox WinOF driver Rev 5.35 package.

Mellanox WinOF is composed of several software modules that contain InfiniBand and Ethernet drivers for ConnectX-3 and ConnectX-3 Pro adapter cards. The Mellanox WinOF driver supports 10, 40 or 56 Gb/s Ethernet, and 40 or 56 Gb/s InfiniBand network ports. The port type is determined upon boot based on card capabilities and user settings.

The Mellanox VPI WinOF driver release introduces the following capabilities:

- Support for Single and Dual port Adapters
- Up to 16 Rx queues per port
- Rx steering mode (RSS)
- Hardware Tx/Rx checksum calculation
- Large Send off-load (i.e., TCP Segmentation Off-load)
- Hardware multicast filtering
- Adaptive interrupt moderation
- Support for MSI-X interrupts
- Support for Auto-Sensing of Link level protocol
- NDK with SMB-Direct
- NDv1 and v2 API support in user space
- VMQ for Hypervisor
- CIM and PowerShell

Ethernet Only:

- Hardware VLAN filtering
- Header Data Split
- RDMA over Converged Ethernet
  - RoCE MAC Based (v1)
  - RoCE IP Based (v1)
  - RoCE over UDP (v2)
- DSCP over IPv4
- NVGRE hardware off-load in ConnectX®-3 Pro
- Ports TX arbitration/Bandwidth allocation per port
- Ports RX arbitration
- Enhanced Transmission Selection (ETS)

- SR-IOV Ethernet on Windows Server 2012 R2/2016 Hypervisor with Windows Server 2012 and above guests.
- Virtual Machine Multiple Queue (VMMQ)
- Network Direct Kernel Provider Interface
- PacketDirect Provider Interface

InfiniBand Only:

- SR-IOV over KVM Hypervisor
- Diagnostic tools

For the complete list of Ethernet and InfiniBand Known Issues and Limitations, WinOF Release Notes (www.mellanox.com -> Products -> Software -> InfiniBand/VPI Drivers -> Windows SW/ Drivers).

1.1 Supplied Packages

Mellanox WinOF driver Rev 5.35 includes the following package:

- MLNX_VPI_WinOF-<version>_All_<OS>_<arch>.exe:

  In this package, the port default is auto and RoCE v2 is enabled

1.2 WinOF Set of Documentation

Under <installation_directory>\Documentation:

- License file
- User Manual (this document)
- MLNX_VPI_WinOF Release Notes

1.3 Windows MPI (MS-MPI)

Message Passing Interface (MPI) is meant to provide virtual topology, synchronization, and communication functionality between a set of processes. MPI enables running one process on several hosts.

- Windows MPI runs over the following protocols:
  - Sockets (Ethernet)
  - Network Direct (ND)

For further details on MPI, please refer to Appendix B,“Windows MPI (MS-MPI),” on page 212.

1.4 Windows Nano Server 2016

Windows Server 2016 offers a new installation option: Nano Server, a remotely administered server operating system. WinOF currently supports driver-only installation over Nano server, with the following capabilities:

- Support for Single and Dual port Adapters
- Up to 16 Rx queues per port
- Rx steering mode (RSS)
- Hardware Tx/Rx checksum calculation
- Large Send off-load (i.e., TCP Segmentation Off-load)
- Hardware multicast filtering
- Adaptive interrupt moderation
- Support for MSI-X interrupts
- Support for Auto-Sensing of Link level protocol
- NDK with SMB-Direct
- Ethernet Only:
- Hardware VLAN filtering
- Header Data Split
- RDMA over Converged Ethernet
- RoCE MAC Based (v1)
- RoCE IP Based (v1)
- RoCE over UDP (v2)
- DSCP over IPv4
- NVGRE hardware off-load in ConnectX®-3 Pro
- Ports TX arbitration/Bandwidth allocation per port
- Ports RX arbitration
- Enhanced Transmission Selection (ETS)
- SR-IOV Ethernet on Windows Server 2016 Hypervisor with Windows Server 2012 and
- above guests.
- Virtual Machine Multiple Queue (VMMQ)
- Network Direct Kernel Provider Interface
- PacketDirect Provider Interface

2 Installation

2.1 Hardware and Software Requirements

Table 5 - Hardware and Software Requirements

Description[a]	Package
Windows Server 2008 R2[b]	MLNX_VPI_WinOF-5_35_All_win2008R2_x64.exe
Windows Server 2012	MLNX_VPI_WinOF-5_35_All_win2012_x64.exe
Windows Server 2012 R2	MLNX_VPI_WinOF-5_35_All_win2012R2_x64.exe
Windows 7 Client (64 bit only)[c]	MLNX_VPI_WinOF-5_35_All_win2008R2_x64.exe
Windows 8.1 Client (64 bit only)[b]	MLNX_VPI_WinOF-5_35_All_win2012R2_x64.exe
Windows 10 Client (64 bit only)	MLNX_VPI_WinOF-5_35_All_win2012R2_x64.exe
Windows Server 2016	MLNX_VPI_WinOF-5_35_All_win2016_x64.exe

[a] The Operating System listed above must run with administrator privileges.

[b] When installing WinOF v5.22 and above over Windows Server 2008 R2 or Windows 7, Microsoft Security Advisory 3033929 is required. As of January 2016, SHA-1 hashing algorithm is deprecated. Therefore, Microsoft signs certificate driver with the SHA-2 hashing algorithm.
To support the SHA-2 algorithm, you must install Microsoft Security Advisory 3033929. For additional information, see the following:
1. For Windows update from march 2015, go to:
https://technet.microsoft.com/en-us/library/security/3033929?f=255&MSPPError=-2147217396
2. To see the notification of the SHA-1deprecation, go to:
https://support.microsoft.com/en-us/kb/3123479

[c] These servers are not signed by Microsoft yet - to be signed in a short period of time.

Required Disk Space for Installation is 100MB

2.2 Downloading Mellanox WinOF Driver

To download the .exe according to your Operating System, please follow the steps below:

Step 1. Obtain the machine architecture.

For Windows Server 2008 R2

1. Open a CMD console (Click start, then run, and enter CMD).
2. Enter the following command.

```
> echo %PROCESSOR_ARCHITECTURE%
```

On an x64 (64-bit) machine, the output will be “AMD64”.

For Windows Server 2012 / 2012 R2 / 2016

1. To go to the Start menu, position your mouse in the bottom-right corner of the Remote Desktop of your screen.

2. Open a CMD console (Click Task Manager-->File --> Run new task, and enter CMD).
3. Enter the following command.

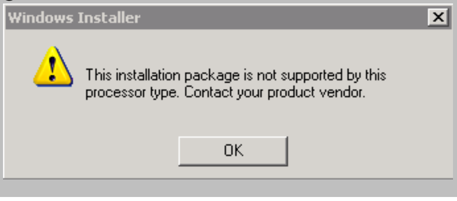
```
> echo %PROCESSOR_ARCHITECTURE%
```

On an x64 (64-bit) machine, the output will be "AMD64".

Step 2. Go to the Mellanox WinOF web page at http://www.mellanox.com > Products > InfiniBand/VPI Drivers => Windows SW/Drivers.

Step 3. Download the .exe image according to the architecture of your machine (see Step 1) and the operating system. The name of the .exe is in the following format MLNX_VPI_WinOF-<version>_All_<OS>_<arch>.exe.

Installing the incorrect .exe file is prohibited. If you do so, an error message will be displayed. For example, if you try to install a 64-bit .exe on a 32-bit machine, the wizard will display the following (or a similar) error message:



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

2.3 Installing Mellanox WinOF Driver

WinOF supports ConnectX-3 and ConnectX-3 Pro adapter cards. In case you have ConnectX-4 adapter card on your server, you will need to install WinOF-2 driver. For details on how to install WinOF-2 driver, please refer to WinOF-2 User Manual.

This section provides instructions for two types of installation procedures:

- "Attended Installation"

  An installation procedure that requires frequent user intervention.
- "Unattended Installation"

  An automated installation procedure that requires no user intervention.

Both Attended and Unattended installations require administrator privileges.

2.3.1 Attended Installation

The following is an example of a MLNX_WinOF_win2016 x64 installation session.

Step 1. Double click the .exe and follow the GUI instructions to install MLNX_WinOF.

> As of MLNX WinOF v4.55, the log option is enabled automatically.
> The default path of the log is: `%LOCALAPPDATA%\MLNX_WinOF.log0`

Step 2. [Optional] Manually configure your setup to contain the logs option.

```
> MLNX_VPI_WinOF-5_35_All_win2016_x64.exe /v"/l*vx [LogFile]"
```

Step 3. [Optional] If you do not want to upgrade your firmware version[1].

```
> MLNX_VPI_WinOF-5_35_All_win2016_x64.exe /v" MT_SKIPFWUPGRD=1"
```

Step 4. [Optional] If you want to control the installation of the WMI/CIM provider[2].

```
> MLNX_VPI_WinOF-5_35_All_win2016_x64.exe /v" MT_WMI=1"
```

Step 5. [Optional] If you want to control whether to restore network configuration or not[3].

```
> MLNX_VPI_WinOF-5_35_All_win2016_x64.exe /v" MT_RESTORECONF=1"
```

For further help, please run:

```
> MLNX_VPI_WinOF-5_35_All_win2016_x64.exe /v" /h"
```

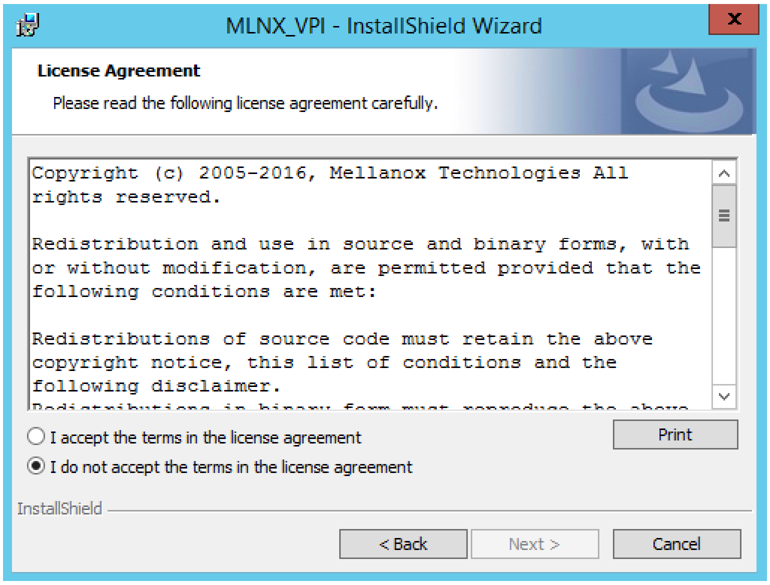
Step 6. [Optional] If you want to control the driver's loading timeout value before burning the firmware, run:

```
MLNX_VPI_WinOF-5_35_All_win2016_x64.exe /v" /MMT_DRIVER_LOAD_TIMEOUT=30"
```

Note: Timeout value is in seconds, and the default value is 120.

Step 7. Click Next in the Welcome screen.

Step 8. Read then accept the license agreement and click Next.



MLNX_VPI - InstallShield Wizard

License Agreement

Please read the following license agreement carefully.

```
Copyright (c) 2005-2016, Mellanox Technologies All
rights reserved.

Redistribution and use in source and binary forms, with
or without modification, are permitted provided that the
following conditions are met:

Redistributions of source code must retain the above
copyright notice, this list of conditions and the
following disclaimer.
```

○ I accept the terms in the license agreement

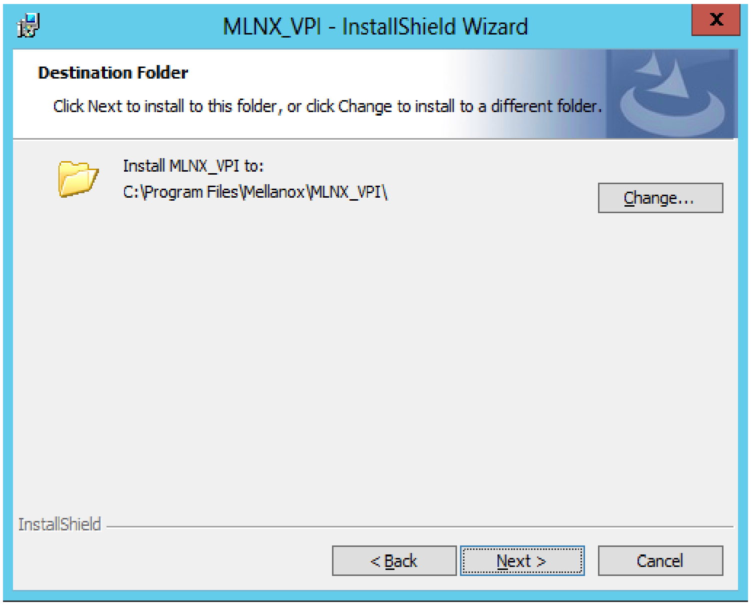
● I do not accept the terms in the license agreement

[1] MT_SKIPFWUPGRD default value is False

[2] MT_WMI default value is True

[3] MT_RESTORECONF default value is True

Step 9. Select the target folder for the installation.



MLNX_VPI - InstallShield Wizard

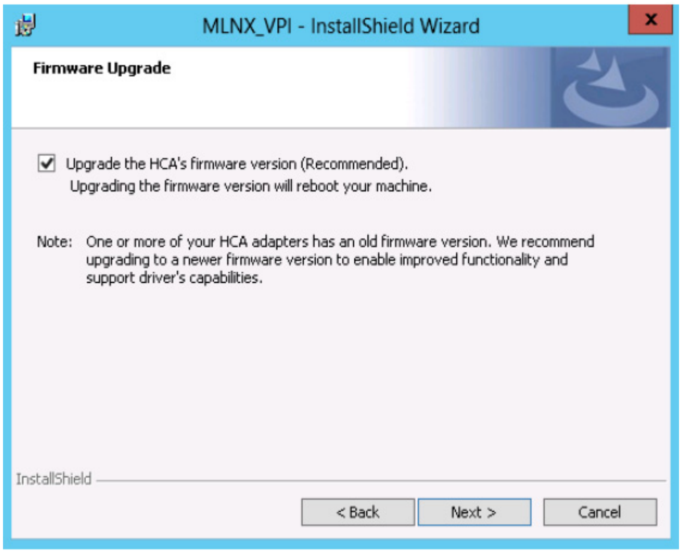
Destination Folder

Click Next to install to this folder, or click Change to install to a different folder.

Install MLNX_VPI to:
C:\Program Files\Mellanox\MLNX_VPI\

Step 10. The firmware upgrade screen will be displayed in the following cases:

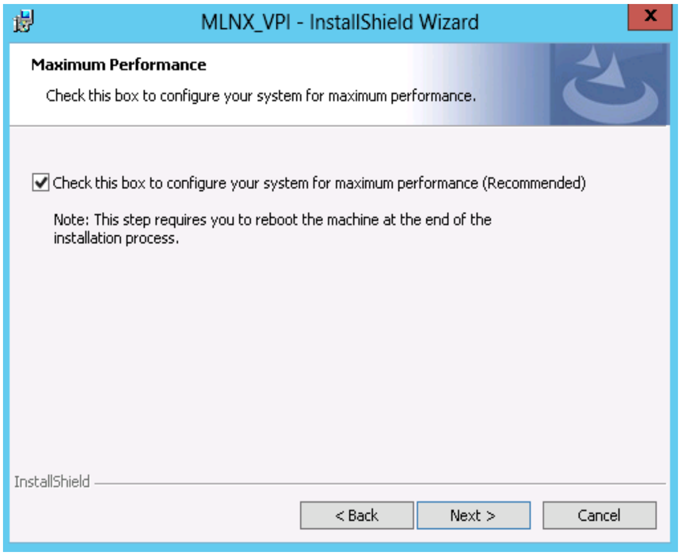
- If the user has an OEM card, in this case the firmware will not be updated.
- If the user has a standard Mellanox card with an older firmware version, the firmware will be updated accordingly. However, if the user has both OEM card and Mellanox card, only Mellanox card will be updated.



☑ Upgrade the HCA's firmware version (Recommended).
Upgrading the firmware version will reboot your machine.

Note: One or more of your HCA adapters has an old firmware version. We recommend upgrading to a newer firmware version to enable improved functionality and support driver's capabilities.

Step 11. Configure your system for maximum performance by checking the maximum performance box.



Maximum Performance

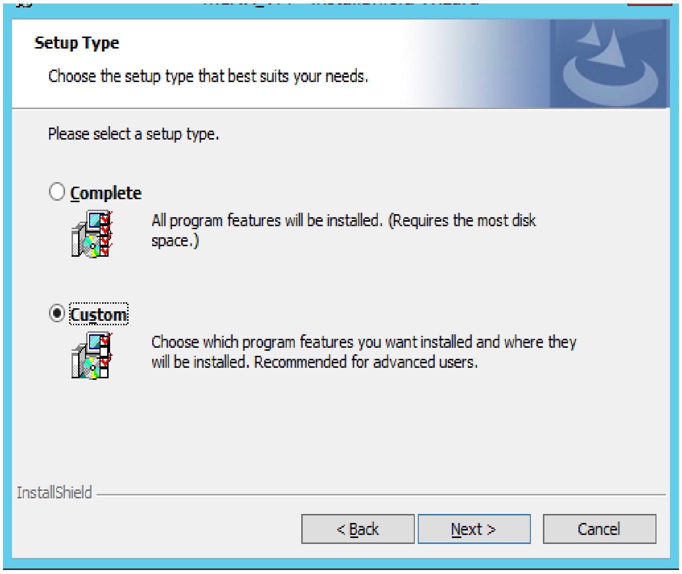
Check this box to configure your system for maximum performance.

☑ Check this box to configure your system for maximum performance (Recommended)

Note: This step requires you to reboot the machine at the end of the installation process.

This step requires rebooting your machine at the end of the installation.

Step 12. In order to complete the installation, select Complete installation.
If you wish to customize the features you want installed, follow Step a and on below.



Setup Type

Choose the setup type that best suits your needs.

Please select a setup type.

○ Complete

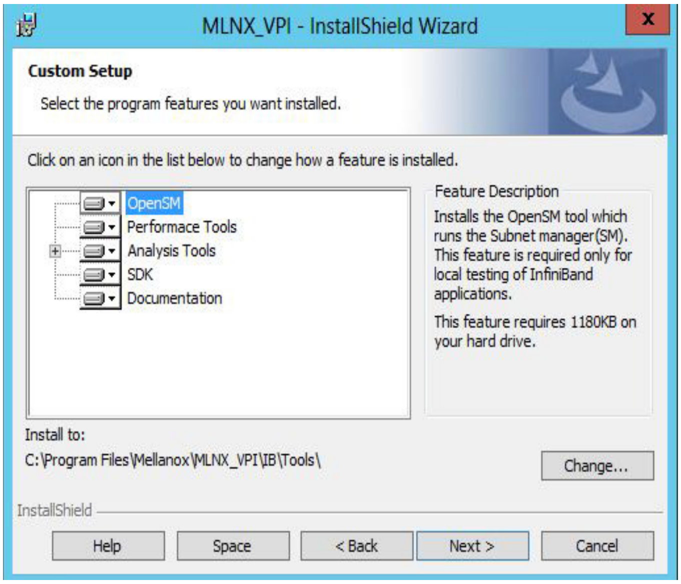
All program features will be installed. (Requires the most disk space.)

● Custom

Choose which program features you want installed and where they will be installed. Recommended for advanced users.

a.Select the desired feature to install:

- OpenSM - installs Windows OpenSM that is required to manage the subnet from a host. OpenSM is part of the driver and installed automatically.
- Performances tools - install the performance tools that are used to measure the InfiniBand performance in user environment.
- Analyze tools - install the tools that can be used either to diagnosed or analyzed the InfiniBand environment.
- SDK - contains the libraries and DLLs for developing InfiniBand application over IBAL.
- Documentation - contains the User Manual and Installation Guide.



MLNX_VPI - InstallShield Wizard

Custom Setup

Select the program features you want installed.

Click on an icon in the list below to change how a feature is installed.

- OpenSM
- Performace Tools
- Analysis Tools
- SDK
- Documentation

Feature Description

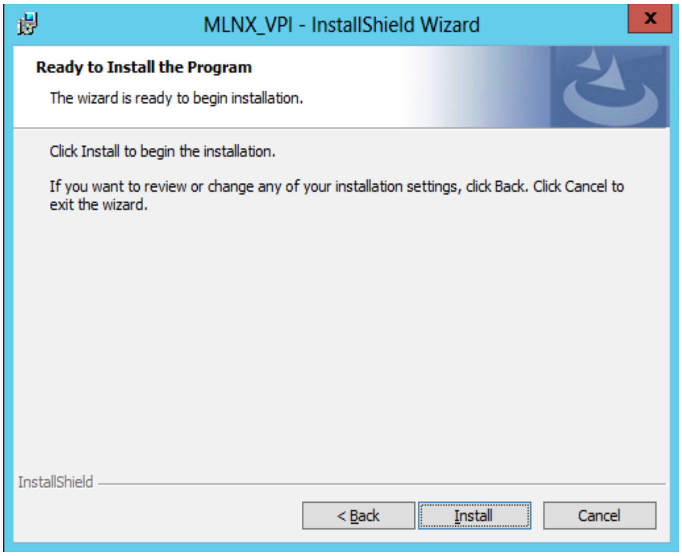
Installs the OpenSM tool which runs the Subnet manager(SM). This feature is required only for local testing of InfiniBand applications.

This feature requires 1180KB on your hard drive.

Install to:

C:\Program Files\Mellanox\MLNX_VPI\IB\Tools\

b. Click Install to start the installation.



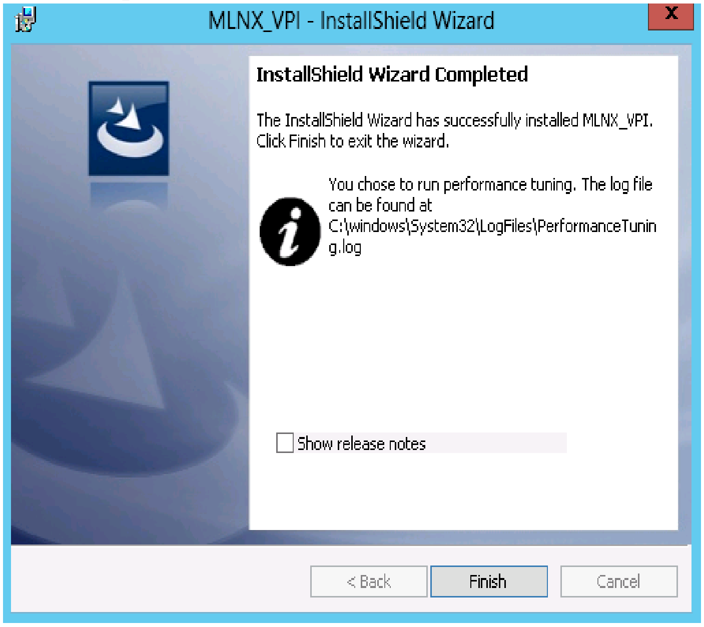
Ready to Install the Program

The wizard is ready to begin installation.

Click Install to begin the installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

Step 13. Click Finish to complete the installation.



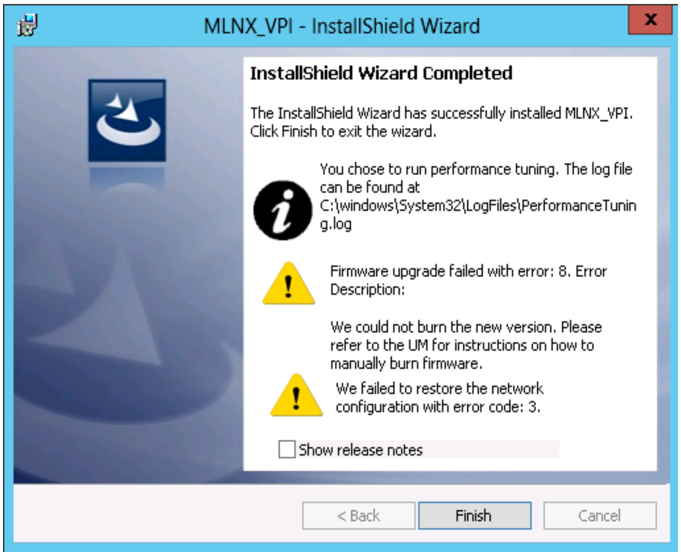
InstallShield Wizard Completed

The InstallShield Wizard has successfully installed MLNX_VPI. Click Finish to exit the wizard.

You chose to run performance tuning. The log file can be found at C:\windows\System32\LogFiles\PerformanceTuning.log

□ Show release notes

- If the firmware upgrade and the restore of the network configuration fails, the following message will be displayed.



MLNX_VPI - InstallShield Wizard

InstallShield Wizard Completed

The InstallShield Wizard has successfully installed MLNX_VPI. Click Finish to exit the wizard.

You chose to run performance tuning. The log file can be found at C:\windows\System32\LogFiles\PerformanceTuning.log

Firmware upgrade failed with error: 8. Error Description:

We could not burn the new version. Please refer to the UM for instructions on how to manually burn firmware.

We failed to restore the network configuration with error code: 3.

2.3.2 Unattended Installation

> If no reboot options are specified, the installer restarts the computer whenever necessary without displaying any prompt or warning to the user.
>
> Use the `/norestart` or `/forcerestart` standard command-line options to control reboots.

The following is an example of a MLNX_WinOF_win2016 x64 unattended installation session.

Step 1. Open a CMD console

[Windows Server 2008 R2] - Click Start-->Run and enter CMD.

[Windows Server 2012 / 2012 R2/ 2016] - Click Start --> Task Manager-->File --> Run new task --> and enter CMD.

Step 2. Install the driver. Run:

```
> MLNX_VPI_WinOF-5_35_All_win2016_x64.exe /S /v"/qn"
```

Step 3. [Optional] Manually configure your setup to contain the logs option:

```
> MLNX_VPI_WinOF-5_35_All_win2016_x64.exe /S /v"/qn" /v"/l*vx [LogFile]"
```

> Starting from MLNX WinOF v4.55, the log option is enabled automatically. The default path of the log is: `%LOCALAPPDATA%\MLNX_WinOF.log0`

Step 4. [Optional] If you do not wish to upgrade your firmware version[1].

```
> MLNX_VPI_WinOF-5_35_All_win2016_x64.exe /v" MT_SKIPFWUPGRD=1"
```

[1] MT_SKIPFWUPGRD default value is False

Step 5. [Optional] If you wish to control the installation of the WMI/CIM provider[1].

```
> MLNX_VPI_WinOF-5_35_All_win2016_x64.exe /v" /MT_WMI=1"
```

Step 6. [Optional] If you wish to control whether to restore network configuration or not[2].

```
> MLNX_VPI_WinOF-5_35_All_win2016_x64.exe /v" MT_RESTORECONF=1"
```

For further help, please run:

```
> MLNX_VPI_WinOF-5_35_All_win2016_x64.exe /v" /h"
```

Step 7. [Optional] If you want to control the driver's loading timeout value before burning the firmware, run:

```
MLNX_VPI_WinOF-5_35_All_win2016_x64.exe /v" /MMT_DRIVER_LOAD_TIMEOUT=30"
```

Timeout value is in seconds, and the default value is 120.

Step 8. [Optional] If you wish to control whether to execute performance tuning or not[3].

```
> MLNX_VPI_WinOF_5_35_All_win2016_x64.exe /vPERFCHECK=0 /vPERFCHECK=0
```

Step 9. [Optional] If you wish to control whether to install ND provider or not[4].

```
> MLNX_VPI_WinOF_5_35_All_win2016_x64.exe /vMT_NDPROPERTY=1
```

Applications that hold the driver files (such as ND applications) will be closed during the unattended installation.

When using the unattended installation option, user should check for the setup return codes and react to them accordingly. See Section 5.1.1.1, "Setup Return Codes", on page 198.

2.4 Installation Results

Upon installation completion, you can verify the successful addition of the network card(s) through the Device Manager.

Upon installation completion, the inf files can be located at:

- %ProgramFiles%\Mellanox\MLNX_VPI\ETH
- %ProgramFiles%\Mellanox\MLNX_VPI\HW\mlx4_bus
- %ProgramFiles%\Mellanox\MLNX_VPI\IB\IPoIB

To see the Mellanox network adapter device, and the Ethernet or IPoIB network device (depending on the used card) for each port, display the Device Manager and expand "System devices" or "Network adapters".

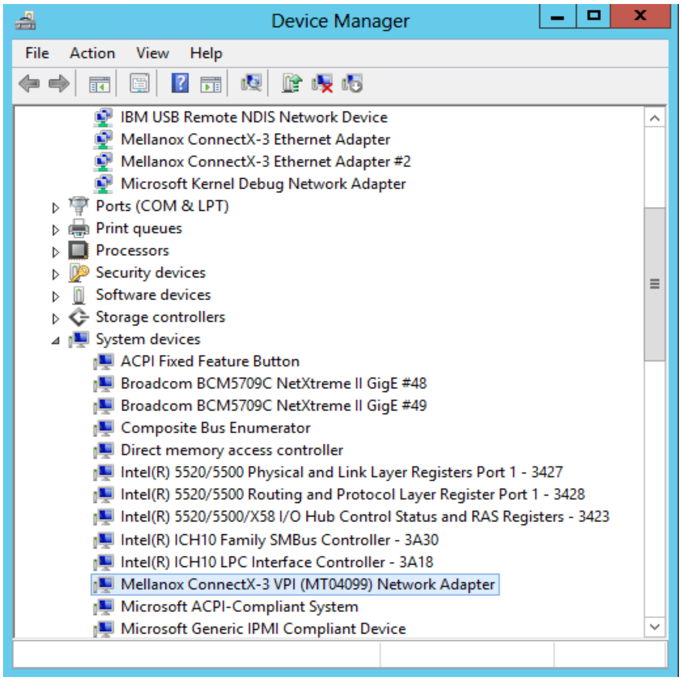
[1] MT_WMI default value is True

[2] MT_RESTORECONF default value is True

[3] PERFCHECK default value is True

[4] MT_NDPROPERTY default value is True

Figure 1: Installation Results



IBM USB Remote NDIS Network Device
Mellanox ConnectX-3 Ethernet Adapter
Mellanox ConnectX-3 Ethernet Adapter #2
Microsoft Kernel Debug Network Adapter
Ports (COM & LPT)
Print queues
Processors
Security devices
Software devices
Storage controllers
System devices
ACPI Fixed Feature Button
Broadcom BCM5709C NetXtreme II GigE #48
Broadcom BCM5709C NetXtreme II GigE #49
Composite Bus Enumerator
Direct memory access controller
Intel(R) 5520/5500 Physical and Link Layer Registers Port 1 - 3427
Intel(R) 5520/5500 Routing and Protocol Layer Register Port 1 - 3428
Intel(R) 5520/5500/X58 I/O Hub Control Status and RAS Registers - 3423
Intel(R) ICH10 Family SMBus Controller - 3A30
Intel(R) ICH10 LPC Interface Controller - 3A18
Mellanox ConnectX-3 VPI (MT04099) Network Adapter
Microsoft ACPI-Compliant System
Microsoft Generic IPMI Compliant Device

2.5 Extracting Files Without Running Installation

To extract the files without running installation, perform the following steps.

Step 1. Open a CMD console

[Windows Server 2008 R2] - Click Start-->Run and enter CMD.

[Windows Server 2012 / 2012 R2/ 2016] - Click Start --> Task Manager-->File --> Run new task --> and enter CMD.

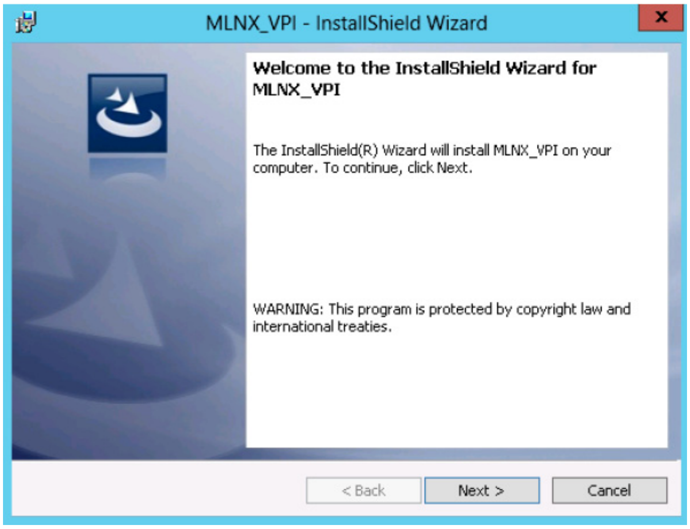
Step 2. Extract the driver and the tools:

```
> MLNX_VPI_WinOF-5_35_All_win2016_x64.exe /a
```

- To extract the driver files only:

```
> MLNX_VPI_WinOF-5_35_All_win2016_x64.exe /a /vMT_DRIVERS_ONLY=1
```

Step 3. Click Next to create a server image.



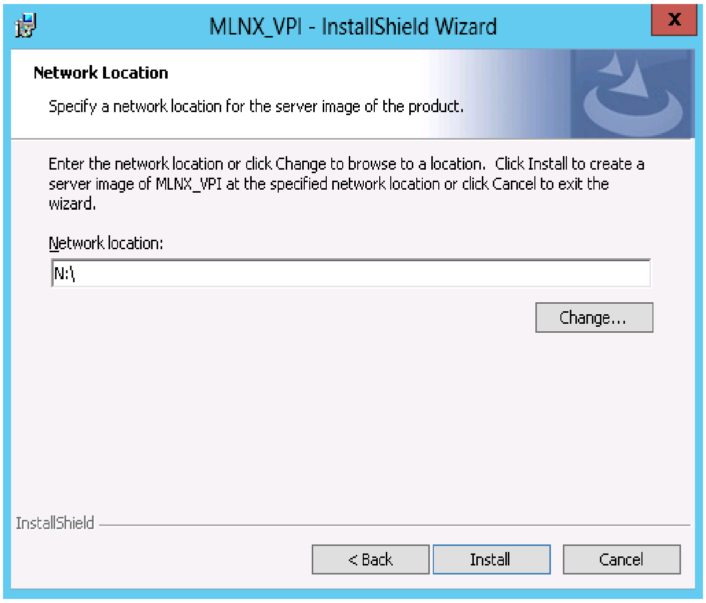
MLNX_VPI - InstallShield Wizard

Welcome to the InstallShield Wizard for MLNX_VPI

The InstallShield(R) Wizard will install MLNX_VPI on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

Step 4. Click Change and specify the location in which the files are extracted to.



MLNX_VPI - InstallShield Wizard

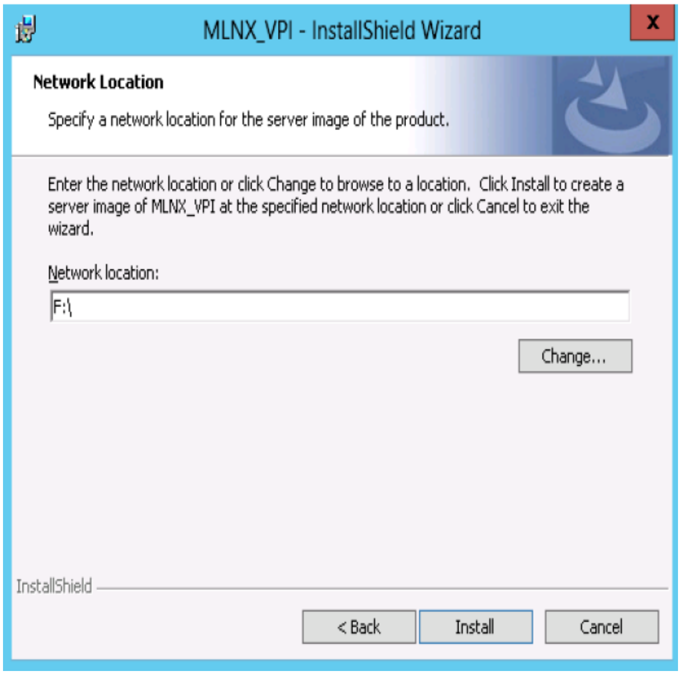
Network Location

Specify a network location for the server image of the product.

Enter the network location or click Change to browse to a location. Click Install to create a server image of MLNX_VPI at the specified network location or click Cancel to exit the wizard.

Network location: N:\

Step 5. Click Install to extract this folder, or click Change to install to a different folder.



Network Location

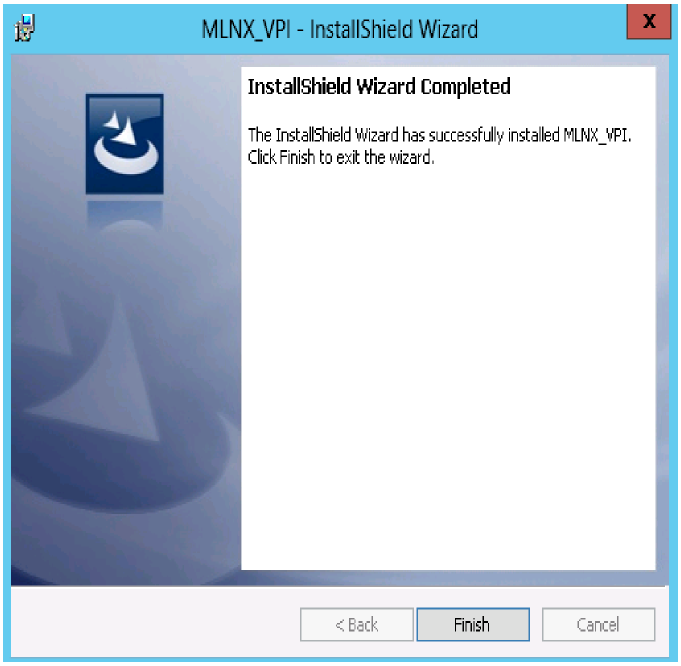
Specify a network location for the server image of the product.

Enter the network location or click Change to browse to a location. Click Install to create a server image of MLNX_VPI at the specified network location or click Cancel to exit the wizard.

Network location:

F:\

Step 6. To complete the extraction, click Finish.



InstallShield Wizard Completed

The InstallShield Wizard has successfully installed MLNX_VPI. Click Finish to exit the wizard.

2.6 Uninstalling Mellanox WinOF Driver

2.6.1 Attended Uninstallation

➢ ***To uninstall MLNX_WinOF on a single node:***

1. Click Start-> Control Panel-> Programs and Features-> MLNX_VPI-> Uninstall. (NOTE: This requires elevated administrator privileges – see Section 1.1, “Supplied Packages”, on page 20 for details.)
2. Double click the .exe and follow the instructions of the install wizard.
3. Click Start -> All Programs -> Mellanox Technologies -> MLNX_WinOF -> Uninstall MLNX_WinOF.

2.6.2 Unattended Uninstallation

> If no reboot options are specified, the installer restarts the computer whenever necessary without displaying any prompt or warning to the user.
>
> Use the `/norestart` or `/forcerestart` standard command-line options to control reboots.

➢ ***To uninstall MLNX_WinOF in unattended mode:***

Step 1. Open a CMD console

[Windows Server 2008 R2] - Click Start-->Run and enter CMD.

[Windows Server 2012 / 2012 R2/ 2016] - Click Start --> Task Manager-->File --> Run new task --> and enter CMD.

Step 2. Uninstall the driver. Run:

```
 > MLNX_VPI_WinOF-4_95_All_win2016_x64.exe /x /v"/qn"
```

2.7 Firmware Upgrade

If the machine has a standard Mellanox card with an older firmware version, the firmware will be updated automatically as part of the installation of the WinOF package.

For information on how to upgrade firmware manually please refer to MFT User Manual: www.mellanox.com ->Products -> InfiniBand/VPI Drivers -> Firmware Tools

2.8 Upgrading Mellanox WinOF Driver

The upgrade process differs between various Operating Systems.

- When upgrading from an Inbox version or any other one, the network configuration is automatically saved upon driver upgrade
- Windows Server 2008 R2:

  When upgrading from WinOF version 3.2.0 to version 4.40 and above, the MLNX_WinOF driver upgrades the driver automatically by uninstalling the previous version and installing the new driver. The existing configuration files are not saved upon driver upgrade.

- Windows Server 2012 and above:
  - When upgrading from WinOF version 4.2 to version 4.40 and above, the MLNX_WinOF driver does not completely uninstall the previous version, but rather upgrades only the components that require upgrade. The network configuration is saved upon driver upgrade.
  - When upgrading from Inbox or any other version, the network configuration is automatically saved upon driver upgrade, except for the RoCE mode that will be changed to V2. For further details, please see section Section 3.1.4.8, “RoCE Default Configuration”, on page 54.

2.9 Booting Windows from an iSCSI Target or PXE

2.9.1 Configuring the WDS, DHCP and iSCSI Servers

2.9.1.1 Configuring the WDS Server

➢ ***To configure the WDS server:***

1. Install the WDS server.
2. Extract the Mellanox drivers to a local directory using the '-a' parameter.

   For boot over Ethernet, when using adapter cards with older firmware version than 2.30.8000, you need to extract the PXE package, otherwise use Mellanox WinOF VPI package.

   Example:

   ```
   Mellanox.msi.exe -a
   ```

3. Add the Mellanox driver to boot.wim[1].

   ```
   dism /Mount-Wim /WimFile:boot.wim /index:2 /MountDir:mnt
   dism /Image:mnt /Add-Driver /Driver:drivers /recurse
   dism /Unmount-Wim /MountDir:mnt /commit
   ```

4. Add the Mellanox driver to install.wim[2].

   ```
   dism /Mount-Wim /WimFile:install.wim /index:4 /MountDir:mnt
   dism /Image:mnt /Add-Driver /Driver:drivers /recurse
   dism /Unmount-Wim /MountDir:mnt /commit
   ```

5. Add the new boot and install images to WDS.

For additional details on WDS, please refer to:

http://technet.microsoft.com/en-us/library/jj648426.aspx

[1] Use `‘index:2’` for Windows setup and `‘index:1’` for WinPE.

[2] When adding the Mellanox driver to install.wim, verify you are using the appropriate index for your OS flavor. To check the OS run `‘imagex /info install.win’`.

2.9.1.2 Configuring iSCSI Target

➢ ***To configure iSCSI Target:***

1. Install iSCSI Target (e.g StartWind).
2. Add to the iSCSI target initiators the IP addresses of the iSCSI clients.

2.9.1.3 Configuring the DHCP Server

➢ ***To configure the DHCP server:***

1. Install a DHCP server.
2. Add to IPv4 a new scope.
3. Add boot client identifier (MAC/GUID) to the DHCP reservation.
4. Add to the reserved IP address the following options if DHCP and WDS are deployed on the same server:

Table 6 - Reserved IP Address Options

Option	Name	Value
017	Root Path	**iscsi:11.4.12.65::::iqn:2011-01:iscsiboot** Assuming the iSCSI target IP is: **11.4.12.65** and the Target Name: **iqn:2011-01:iscsiboot**
060	PXEClient	PXEClient
066	Boot Server Host Name	WDS server IP address
067	Boot File Name	boot\x86\wdsnbp.com

When DHCP and WDS are NOT deployed on the same server, DHCP options (60, 66, 67) should be empty, and the WDS option 60 must be configured.

2.9.2 Configuring the Client Machine

➢ ***To configuring your client:***

1. Verify the Mellanox adapter card is burned with the correct Mellanox FlexBoot version.

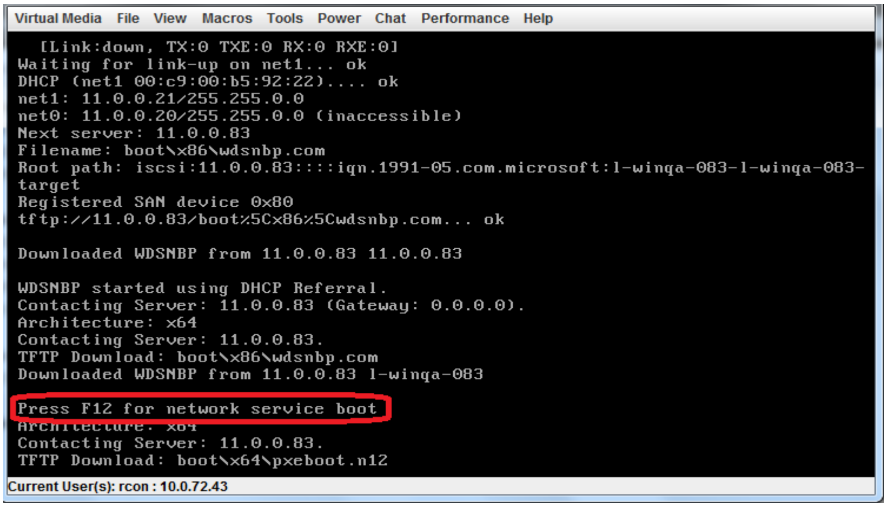
   For boot over Ethernet, when using adapter cards with older firmware version than 2.30.8000, you need to burn the adapter card with Ethernet FlexBoot, otherwise use the VPI FlexBoot.
2. Verify the Mellanox adapter card is burned with the correct firmware version.
3. Set the “Mellanox Adapter Card” as the first boot device in the BIOS settings boot order.

2.9.3 Installing OS

1. Reboot your client.

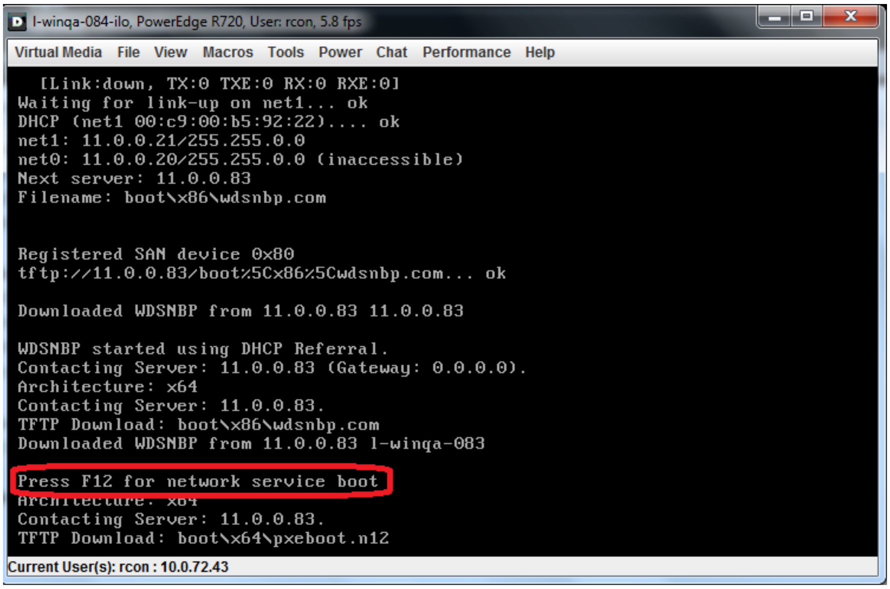
2. Press F12 when asked to proceed to network boot.

Figure 2: Network Service Boot in ISCSi


Virtual Media File View Macros Tools Power Chat Performance Help
Current User(s): rcon : 10.0.72.43

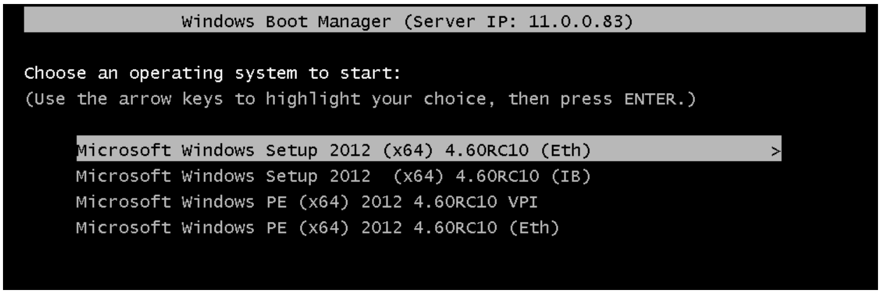
```
  [Link:down, TX:0 TXE:0 RX:0 RXE:0]
Waiting for link-up on net1... ok
DHCP (net1 00:c9:00:b5:92:22).... ok
net1: 11.0.0.21/255.255.0.0
net0: 11.0.0.20/255.255.0.0 (inaccessible)
Next server: 11.0.0.83
Filename: boot\x86\wdsnbp.com
Root path: iscsi:11.0.0.83::::iqn.1991-05.com.microsoft:l-winqa-083-l-winqa-083-
target
Registered SAN device 0x80
tftp://11.0.0.83/boot%5Cx86%5Cwdsnbp.com... ok

Downloaded WDSNBP from 11.0.0.83 11.0.0.83

WDSNBP started using DHCP Referral.
Contacting Server: 11.0.0.83 (Gateway: 0.0.0.0).
Architecture: x64
Contacting Server: 11.0.0.83.
TFTP Download: boot\x86\wdsnbp.com
Downloaded WDSNBP from 11.0.0.83 l-winqa-083

Press F12 for network service boot
Architecture: x64
Contacting Server: 11.0.0.83.
TFTP Download: boot\x64\pxeboot.n12
```

Figure 3: Network Service Boot in PXE


l-winqa-084-ilo, PowerEdge R720, User: rcon, 5.8 fps
Virtual Media File View Macros Tools Power Chat Performance Help
Current User(s): rcon : 10.0.72.43

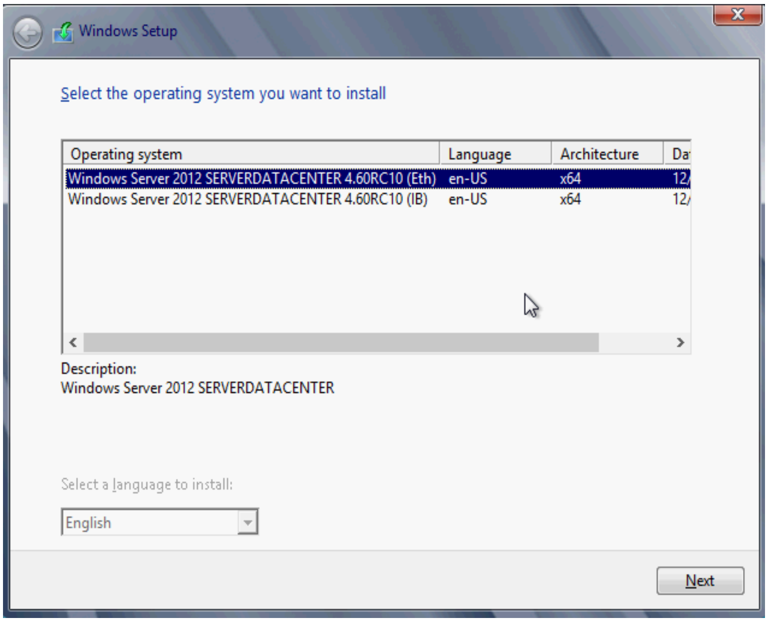
```
  [Link:down, TX:0 TXE:0 RX:0 RXE:0]
Waiting for link-up on net1... ok
DHCP (net1 00:c9:00:b5:92:22).... ok
net1: 11.0.0.21/255.255.0.0
net0: 11.0.0.20/255.255.0.0 (inaccessible)
Next server: 11.0.0.83
Filename: boot\x86\wdsnbp.com


Registered SAN device 0x80
tftp://11.0.0.83/boot%5Cx86%5Cwdsnbp.com... ok

Downloaded WDSNBP from 11.0.0.83 11.0.0.83

WDSNBP started using DHCP Referral.
Contacting Server: 11.0.0.83 (Gateway: 0.0.0.0).
Architecture: x64
Contacting Server: 11.0.0.83.
TFTP Download: boot\x86\wdsnbp.com
Downloaded WDSNBP from 11.0.0.83 l-winqa-083

Press F12 for network service boot
Architecture: x64
Contacting Server: 11.0.0.83.
TFTP Download: boot\x64\pxeboot.n12
```

3. Choose the relevant boot image from the list of all available boot images presented.



```
                 Windows Boot Manager (Server IP: 11.0.0.83)

Choose an operating system to start:
(Use the arrow keys to highlight your choice, then press ENTER.)

     Microsoft Windows Setup 2012 (x64) 4.60RC10 (Eth)                  >
     Microsoft Windows Setup 2012  (x64) 4.60RC10 (IB)
     Microsoft Windows PE (x64) 2012 4.60RC10 VPI
     Microsoft Windows PE (x64) 2012 4.60RC10 (Eth)
```

4. Choose the Operating System you wish to install.



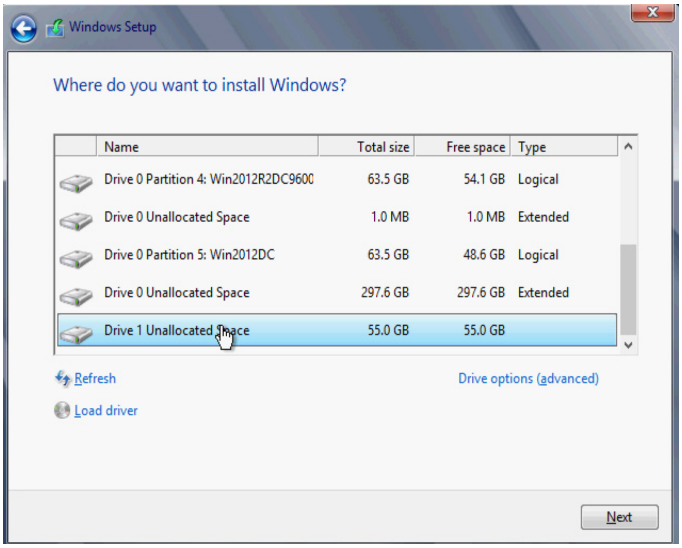
Select the operating system you want to install

Operating system	Language	Architecture	Da
Windows Server 2012 SERVERDATACENTER 4.60RC10 (Eth)	en-US	x64	12/
Windows Server 2012 SERVERDATACENTER 4.60RC10 (IB)	en-US	x64	12/

Description:
Windows Server 2012 SERVERDATACENTER

Select a language to install:
English

5. Run the Windows Setup Wizard.
6. Choose target drive to install Windows and follow the instructions presented by the installation Wizard.



Where do you want to install Windows?

Name	Total size	Free space	Type
Drive 0 Partition 4: Win2012R2DC9600	63.5 GB	54.1 GB	Logical
Drive 0 Unallocated Space	1.0 MB	1.0 MB	Extended
Drive 0 Partition 5: Win2012DC	63.5 GB	48.6 GB	Logical
Drive 0 Unallocated Space	297.6 GB	297.6 GB	Extended
Drive 1 Unallocated Space	55.0 GB	55.0 GB	

Installation process will start once completing all the required steps in the Wizard, the Client will reboot and will boot from the iSCSI target.

2.10 Deploying the Driver on a Nano Server

2.10.1 Offline Installation

➢ ***To deploy the Driver on a Nano Server:***

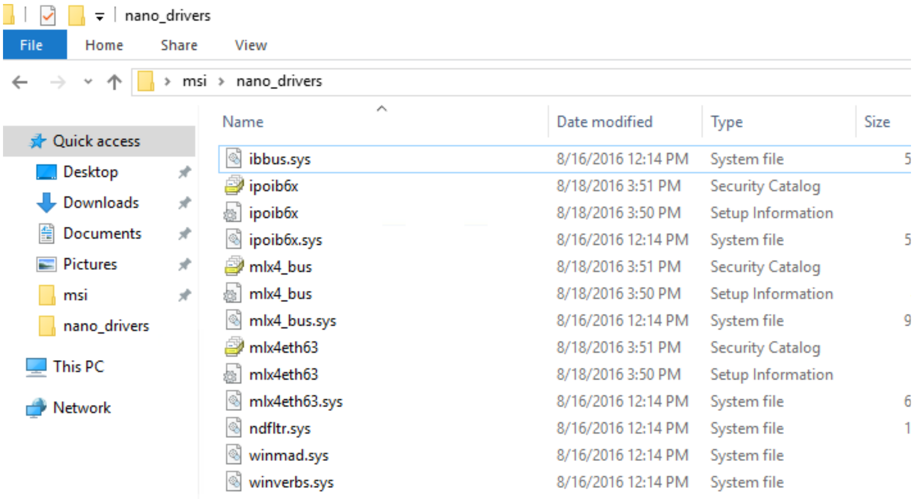
Step 1. Go to the Mellanox WinOF web page at http://www.mellanox.com > Products > InfiniBand/VPI Drivers => Windows SW/Drivers.

Step 2. Download the WinOF driver (MLNX_VPI_WinOF-5_35_All_win2016_x64.exe).

Step 3. Extract the driver to a local directory (see Section 2.5, “Extracting Files Without Running Installation”, on page 31).

Step 4. Copy the contents of this directory to C:\WS2016TP5_Drivers.

The directory contents should appear as follows:



Name	Date modified	Type	Size
ibbus.sys	8/16/2016 12:14 PM	System file	5
ipoib6x	8/18/2016 3:51 PM	Security Catalog	
ipoib6x	8/18/2016 3:50 PM	Setup Information	
ipoib6x.sys	8/16/2016 12:14 PM	System file	5
mlx4_bus	8/18/2016 3:51 PM	Security Catalog	
mlx4_bus	8/18/2016 3:50 PM	Setup Information	
mlx4_bus.sys	8/16/2016 12:14 PM	System file	9
mlx4eth63	8/18/2016 3:51 PM	Security Catalog	
mlx4eth63	8/18/2016 3:50 PM	Setup Information	
mlx4eth63.sys	8/16/2016 12:14 PM	System file	6
ndfltr.sys	8/16/2016 12:14 PM	System file	1
winmad.sys	8/16/2016 12:14 PM	System file	
winverbs.sys	8/16/2016 12:14 PM	System file	

This location should be specified for DriversPath property when injecting driver into the Nano server image:

```
 New-NanoServerImage -MediaPath \\Path\To\Media\en_us -BasePath .\Base -TargetPath
.\InjectingDrivers.vhdx -DriversPath C:\WS2016TP5_Drivers
```

Step 5. Create the Nano server image.

Follow the instructions in the TechNet article for "Getting Started with Nano Server" https://technet.microsoft.com/en-us/library/mt126167.aspx

2.10.2 Online Update

Step 1. Go to the Mellanox WinOF web page at http://www.mellanox.com > Products > InfiniBand/VPI Drivers => Windows SW/Drivers.

Step 2. Download the WinOF driver (MLNX_VPI_WinOF-5_35_All_win2016_x64.exe).

Step 3. Extract the Mellanox drivers to a local directory (see Section 2.5, “Extracting Files Without Running Installation”, on page 31).

Step 4. Open a remote connection to the Nano server.

Step 5. Copy all the driver files to the Nano server

Step 6. Install the driver

- For IPoIB devices, first install the IPoIB driver and then the bus driver
- For Ethernet devices driver, first install the Ethernet driver and then the bus driver
- For VPI devices, first update the Ethernet driver, then the IPoIB driver and last, the bus driver

The following are the installation commands per driver type:

- IPoIB driver installation:

```
pnputil  -I -a  <Driver_Folder>\ipoib6x.inf
```

- Ethernet driver installation:

```
pnputil  -I -a  <Driver_Folder>\mlx4eth63.inf
```

- Bus driver installation:

```
pnputil  -I -a  <Driver_Folder>\mlx4_bus.inf
```

When upgrading the driver on a server where the remote connection was done over Mellanox device, there will be a loss of connectivity between the update of the Ethernet/IPoIB driver and the bus driver. To upgrade, it is recommended to run a script that will execute all the required upgrade commands.

3 Features Overview and 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.

3.1 Ethernet Network

3.1.1 Port Configuration

3.1.1.1 Auto Sensing

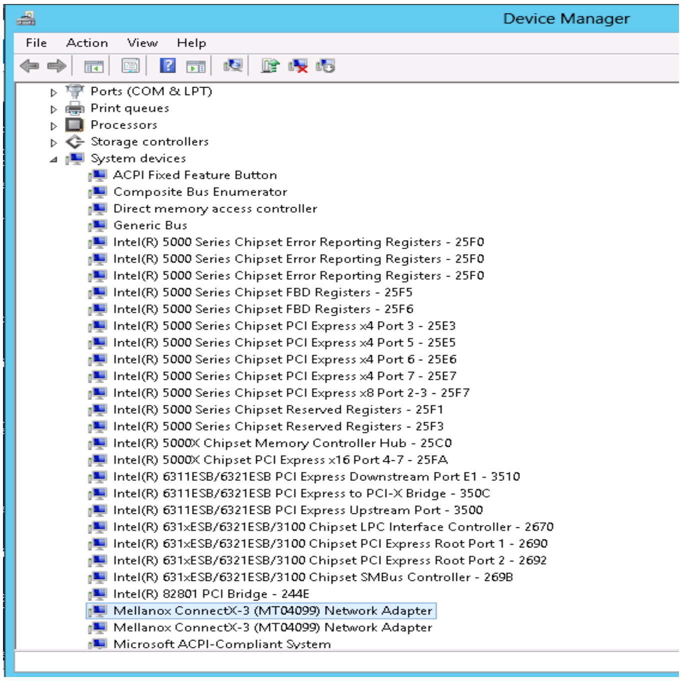
Auto Sensing enables the NIC to automatically sense the link type (InfiniBand or Ethernet) based on the cable connected to the port and load the appropriate driver stack (InfiniBand or Ethernet).

Auto Sensing is performed only when rebooting the machine or after disabling/enabling the adapter cards 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 3.1.1.2, "Port Protocol Configuration", on page 42.

3.1.1.2 Port Protocol Configuration

Step 1. Display the Device Manager and expand “System devices”.

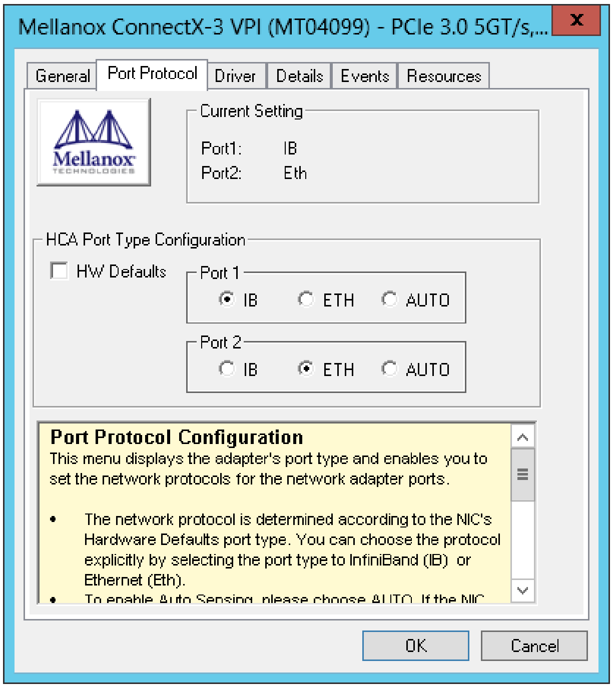


- Ports (COM & LPT)
- Print queues
- Processors
- Storage controllers
- System devices
  - ACPI Fixed Feature Button
  - Composite Bus Enumerator
  - Direct memory access controller
  - Generic Bus
  - Intel(R) 5000 Series Chipset Error Reporting Registers - 25F0
  - Intel(R) 5000 Series Chipset Error Reporting Registers - 25F0
  - Intel(R) 5000 Series Chipset Error Reporting Registers - 25F0
  - Intel(R) 5000 Series Chipset FBD Registers - 25F5
  - Intel(R) 5000 Series Chipset FBD Registers - 25F6
  - Intel(R) 5000 Series Chipset PCI Express x4 Port 3 - 25E3
  - Intel(R) 5000 Series Chipset PCI Express x4 Port 5 - 25E5
  - Intel(R) 5000 Series Chipset PCI Express x4 Port 6 - 25E6
  - Intel(R) 5000 Series Chipset PCI Express x4 Port 7 - 25E7
  - Intel(R) 5000 Series Chipset PCI Express x8 Port 2-3 - 25F7
  - Intel(R) 5000 Series Chipset Reserved Registers - 25F1
  - Intel(R) 5000 Series Chipset Reserved Registers - 25F3
  - Intel(R) 5000X Chipset Memory Controller Hub - 25C0
  - Intel(R) 5000X Chipset PCI Express x16 Port 4-7 - 25FA
  - Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E1 - 3510
  - Intel(R) 6311ESB/6321ESB PCI Express to PCI-X Bridge - 350C
  - Intel(R) 6311ESB/6321ESB PCI Express Upstream Port - 3500
  - Intel(R) 631xESB/6321ESB/3100 Chipset LPC Interface Controller - 2670
  - Intel(R) 631xESB/6321ESB/3100 Chipset PCI Express Root Port 1 - 2690
  - Intel(R) 631xESB/6321ESB/3100 Chipset PCI Express Root Port 2 - 2692
  - Intel(R) 631xESB/6321ESB/3100 Chipset SMBus Controller - 269B
  - Intel(R) 82801 PCI Bridge - 244E
  - Mellanox ConnectX-3 (MT04099) Network Adapter
  - Mellanox ConnectX-3 (MT04099) Network Adapter
  - Microsoft ACPI-Compliant System

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

> 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 window for a dual port VPI adapter card.



Mellanox ConnectX-3 VPI (MT04099) - PCIe 3.0 5GT/s,...

Current Setting

Port1: IB
Port2: Eth

HCA Port Type Configuration

□ HW Defaults

Port 1: ● IB ○ ETH ○ AUTO

Port 2: ○ IB ● ETH ○ AUTO

Port Protocol Configuration

This menu displays the adapter's port type and enables you to set the network protocols for the network adapter ports.

- The network protocol is determined according to the NIC's Hardware Defaults port type. You can choose the protocol explicitly by selecting the port type to InfiniBand (IB) or Ethernet (Eth).
- To enable Auto Sensing, please choose AUTO. If the NIC

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

- If you choose the HW Defaults option, the port protocols will be determined 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).

> Configuring Port 1 as Ethernet with RoCE disabled and Port 2 as IB is not supported by HCA.
>
> If this configuration was created occasionally by auto-sensing, the driver will fail to startup.
>
> If this configuration was intentionally defined by the customer as explained in Step 2. above, the driver will set RoCE mode to v1.25 to create a legal configuration.
>
> In all cases, it will send messages to System Event Log to notify the user about these actions.

- Enable Auto Sensing by checking the AUTO checkbox. If the NIC does not support Auto Sensing, the AUTO option will be grayed out.

> If you choose AUTO, the current setting will indicate the actual port settings: IB or ETH.

For firmware 2.32.5000 and above, there is an option to set port personality using mlxconfig tool. For further details, please refer to MFT User Manual.

3.1.2 Assigning Port IP After Installation

By default, your machine is configured to obtain an automatic IP address via a DHCP server. In some cases, the DHCP server may require the MAC address of the network adapter installed in your machine.

➢ ***To obtain the MAC address:***

Step 1. Open a CMD console

[Windows Server 2008 R2] - Click Start-->Run and enter CMD.

[Windows Server 2012 / 2012 R2/ 2016] - Click Start --> Task Manager-->File --> Run new task --> and enter CMD.

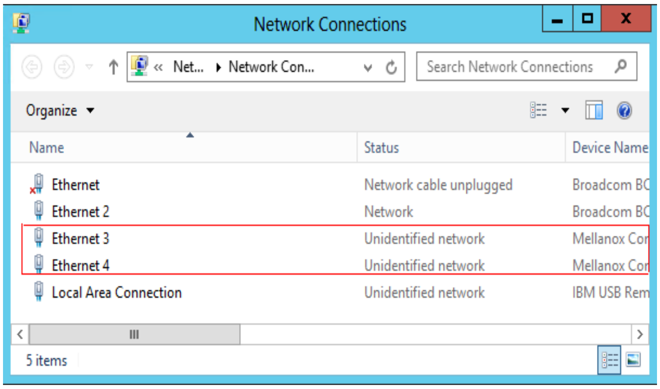
Step 2. Display the MAC address as “Physical Address”

```
> ipconfig /all
```

Configuring a static IP is the same for both IPoIB and Ethernet adapters.

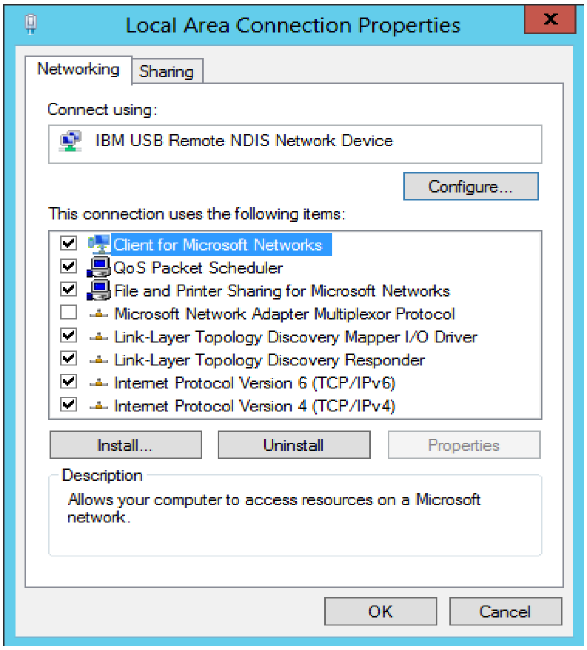
➢ ***To assign a static IP address to a network port after installation:***

Step 1. Open the Network Connections window. Locate Local Area Connections with Mellanox devices.



Name	Status	Device Name
Ethernet	Network cable unplugged	Broadcom BC
Ethernet 2	Network	Broadcom BC
Ethernet 3	Unidentified network	Mellanox Cor
Ethernet 4	Unidentified network	Mellanox Cor
Local Area Connection	Unidentified network	IBM USB Rem

Step 2. Right-click a Mellanox Local Area Connection and left-click Properties.



Local Area Connection Properties

Networking | Sharing

Connect using:

IBM USB Remote NDIS Network Device

Configure...

This connection uses the following items:

- ☑ Client for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ File and Printer Sharing for Microsoft Networks
- ☐ Microsoft Network Adapter Multiplexor Protocol
- ☑ Link-Layer Topology Discovery Mapper I/O Driver
- ☑ Link-Layer Topology Discovery Responder
- ☑ Internet Protocol Version 6 (TCP/IPv6)
- ☑ Internet Protocol Version 4 (TCP/IPv4)

Install... | Uninstall | Properties

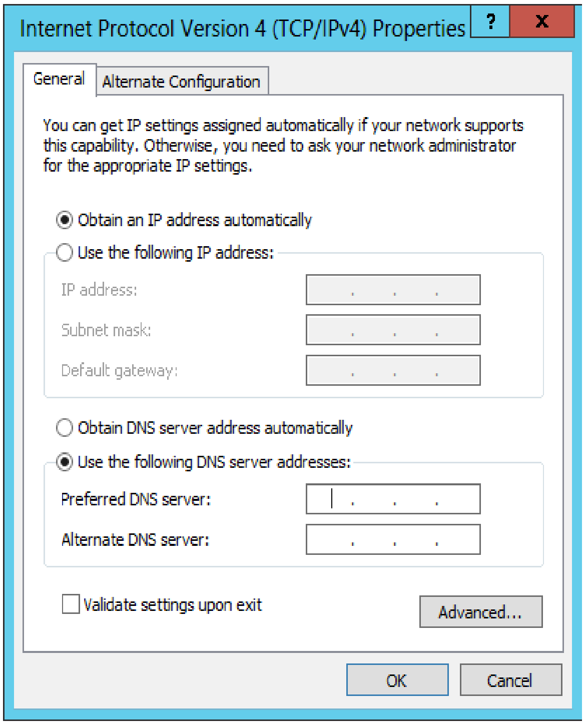
Description

Allows your computer to access resources on a Microsoft network.

OK | Cancel

Step 3. Select Internet Protocol Version 4 (TCP/IPv4) from the scroll list and click Properties.

Step 4. Select the “Use the following IP address:” radio button and enter the desired IP information.



Internet Protocol Version 4 (TCP/IPv4) Properties

General | Alternate Configuration

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

● Obtain an IP address automatically

○ Use the following IP address:

IP address: . . .

Subnet mask: . . .

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

☐ Validate settings upon exit

Advanced...

OK | Cancel

Step 5. Click OK.

Step 6. Close the Local Area Connection dialog.

Step 7. Verify the IP configuration by running ‘ipconfig’ from a CMD console.

```
> ipconfig
...
Ethernet adapter Local Area Connection 4:

   Connection-specific DNS Suffix  . :
   IP Address. . . . . . . . . . . . : 11.4.12.63
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . :
...
```

3.1.3 56GbE Link Speed

3.1.3.1 System Requirements

- Mellanox ConnectX®-3 and ConnectX®-3 Pro cards
- Firmware version: 2.31.5050 and above

3.1.3.2 Configuring 56GbE Link Speed

Mellanox offers proprietary speed of 56GbE link speed over FDR systems. To achieve this, only the switch, supporting this speed, must be configured to enable it. The NIC, on the other hand, auto-detects this configuration automatically.

➢ ***To achieve 56GbE link speed over SwitchX® Based Switch System***

> Make sure your switch supports 56GbE and that you have the relevant switch license installed.

Step 1. Set the system profile to be `eth-single-switch`, and reset the system:

```
switch (config) # system profile eth-single-profile
```

Step 2. Set the speed for the desired interface to 56GbE as follows. For example (for interface 1/1):

```
switch (config) # interface ethernet 1/1
switch (config interface ethernet 1/1) # speed 56000
switch (config interface ethernet 1/1) #
```

Step 3. Verify the speed is 56GbE.

```
switch (config) # show interface ethernet 1/1
Eth1/1
Admin state: Enabled
Operational state: Down
Description: N\A
Mac address: 00:02:c9:5d:e0:26
MTU: 1522 bytes
Flow-control: receive off send off
Actual speed: 56 Gbps
Switchport mode: access
Rx
0 frames
0 unicast frames
0 multicast frames
0 broadcast frames
0 octets
0 error frames
0 discard frames
Tx
0 frames
0 unicast frames
0 multicast frames
0 broadcast frames
0 octets
0 discard frames
switch (config) #
```

3.1.4 RDMA over Converged Ethernet (RoCE)

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 Converged 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® 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, 40GigE and 56GigE link-speed. ConnectX® 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, database, 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 InfiniBand 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.

3.1.4.1 RoCE Configuration

In order to function reliably, RoCE requires a form of flow control. While it is possible to use global flow control, this is normally undesirable, for performance reasons.

The normal and optimal way to use RoCE is to use Priority Flow Control (PFC). To use PFC, it must be enabled on all endpoints and switches in the flow path.

In the following section we present instructions to configure PFC on Mellanox ConnectX™ cards. There are multiple configuration steps required, all of which may be performed via PowerShell. Therefore, although we present each step individually, you may ultimately choose to write a PowerShell script to do them all in one step. Note that administrator privileges are required for these steps.

For further information about RoCE configuration, please refer to:
https://community.mellanox.com

3.1.4.1.1 System Requirements

The following are the driver's prerequisites in order to set or configure RoCE:

- RoCE: ConnectX®-3 and ConnectX®-3 Pro firmware version 2.30.3000 or higher
- RoCEv2: ConnectX®-3 Pro firmware version 2.31.5050 or higher
- All InfiniBand verbs applications which run over InfiniBand verbs should work on RoCE links if they use GRH headers
- Operating Systems: Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows 7 Client, Windows 8.1 Client and Windows Server 2016
- Set HCA to use Ethernet protocol:
  Display the Device Manager and expand "System Devices". Please refer to Section 3.1.1.2, "Port Protocol Configuration", on page 42

3.1.4.1.2 Configuring Windows Host

> Since PFC is responsible for flow controlling at the granularity of traffic priority, it is necessary to assign different priorities to different types of network traffic.
>
> As per RoCE configuration, all ND/NDK traffic is assigned to one or more chosen priorities, where PFC is enabled on those priorities.

Configuring Windows host requires configuring QoS. To configure QoS, please follow the procedure described in Section 3.1.8, "Configuring Quality of Service (QoS)", on page 69

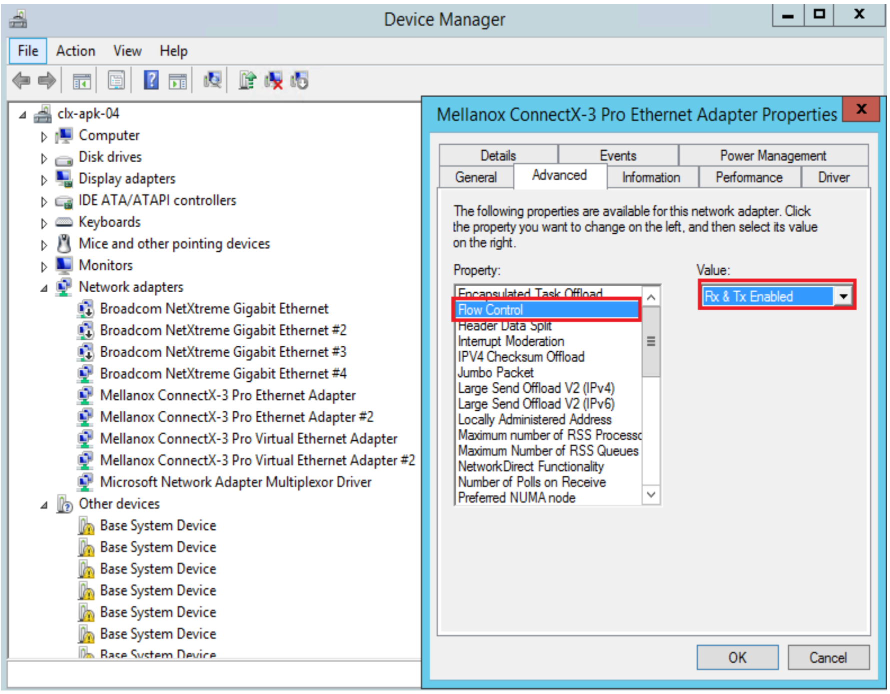
Global Pause (Flow Control)

➢ ***To use Global Pause (Flow Control) mode, disable QoS and Priority:***

```
PS $ Disable-NetQosFlowControl
PS $ Disable-NetAdapterQos <interface name>
```

➢ ***To confirm flow control is enabled in adapter parameters:***

Device manager-> Network adapters-> Mellanox ConnectX-3 Ethernet Adapter-> Properties ->Advanced tab


Device Manager
File
Action
View
Help
clx-apk-04
Computer
Disk drives
Display adapters
IDE ATA/ATAPI controllers
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Broadcom NetXtreme Gigabit Ethernet
Broadcom NetXtreme Gigabit Ethernet #2
Broadcom NetXtreme Gigabit Ethernet #3
Broadcom NetXtreme Gigabit Ethernet #4
Mellanox ConnectX-3 Pro Ethernet Adapter
Mellanox ConnectX-3 Pro Ethernet Adapter #2
Mellanox ConnectX-3 Pro Virtual Ethernet Adapter
Mellanox ConnectX-3 Pro Virtual Ethernet Adapter #2
Microsoft Network Adapter Multiplexor Driver
Other devices
Base System Device
Base System Device
Base System Device
Base System Device
Base System Device
Base System Device
Base System Device
Mellanox ConnectX-3 Pro Ethernet Adapter Properties
Details
Events
Power Management
General
Advanced
Information
Performance
Driver
OK
Cancel

The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property:

Encapsulated Task Offload
Flow Control
Header Data Split
Interrupt Moderation
IPV4 Checksum Offload
Jumbo Packet
Large Send Offload V2 (IPv4)
Large Send Offload V2 (IPv6)
Locally Administered Address
Maximum number of RSS Processo
Maximum Number of RSS Queues
NetworkDirect Functionality
Number of Polls on Receive
Preferred NUMA node

Value: Rx & Tx Enabled

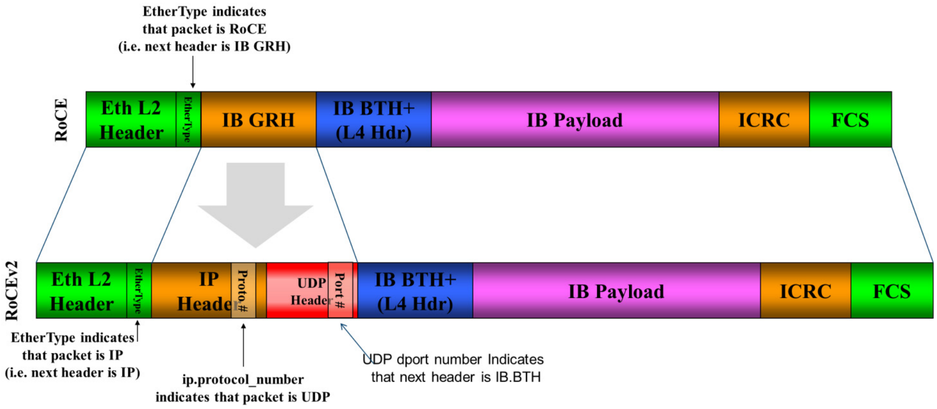
3.1.4.2 RoCEv2

RoCE has two addressing modes: MAC based GIDs, and IP address based GIDs. If the IP address changes while the system is running, the GID for the port will automatically be updated with the new IP address, using either IPv4 or IPv6.

RoCE IP based allows RoCE traffic between Windows and Linux systems, which use IP based GIDs by default.

A straightforward extension of the RoCE protocol enables traffic to operate in layer 3 environments. This capability is obtained via a simple modification of the RoCE packet format. Instead of the GRH used in RoCE, routable RoCE packets carry an IP header which allows traversal of IP L3 Routers and a UDP header that serves as a stateless encapsulation layer for the RDMA Transport Protocol Packets over IP.

Figure 4: RoCE and RoCE v2 Frame Format Differences



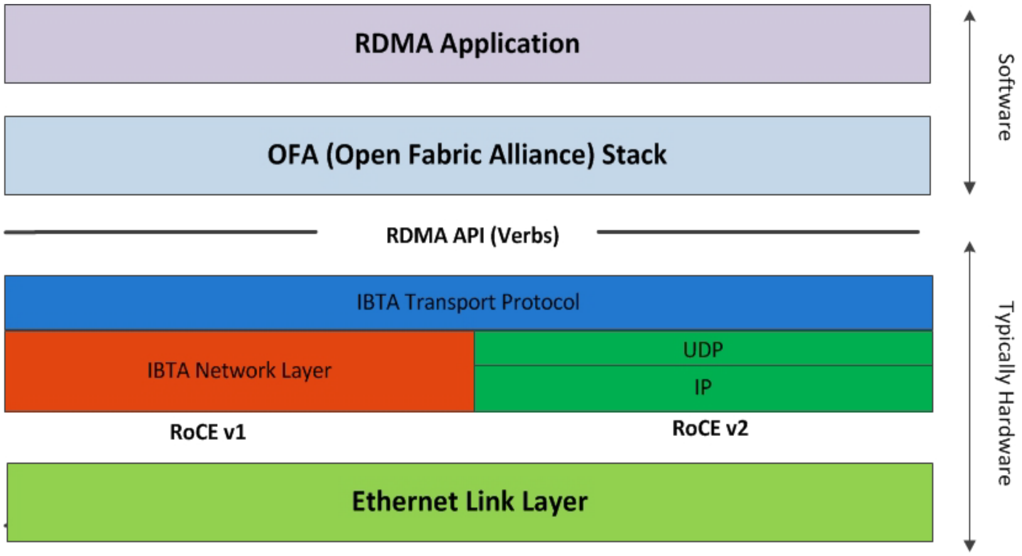
The proposed RoCE packets use a well-known UDP destination port value that unequivocally distinguishes the datagram. Similar to other protocols that use UDP encapsulation, the UDP source port field is used to carry an opaque flow-identifier that allows network devices to implement packet forwarding optimizations (e.g. ECMP) while staying agnostic to the specifics of the protocol header format.

The UDP source port is calculated as follows: `UDP.SrcPort = (SrcPort XOR DstPort) OR 0xC000`, where SrcPort and DstPort are the ports used to establish the connection.
For example, in a Network Direct application, when connecting to a remote peer, the destination IP address and the destination port must be provided as they are used in the calculation above. The source port provision is optional.

Furthermore, since this change exclusively affects the packet format on the wire, and due to the fact that with RDMA semantics packets are generated and consumed below the AP applications can seamlessly operate over any form of RDMA service (including the routable version of RoCE as shown in Figure 4,“RoCE and RoCE v2 Frame Format Differences”), in a completely transparent way[1].

[1] Standard RDMA APIs are IP based already for all existing RDMA technologies

Figure 5: RoCE Protocol Stack



The fabric must use the same protocol stack in order for nodes to communicate.

The default RoCE mode in Windows is MAC based.

The default RoCE mode in Linux is IP based.

In order to communicate between Windows and Linux over RoCE, please change the RoCE mode in Windows to IP based.

3.1.4.3 RoCE v2 UDP Port

In RoCEv2, the RDMA payload is encapsulated as UDP payload with a specific UDP destination port number indicating that the payload is RDMA.

Prior to WinOF v5.02v5.22, the destination port number indicating RoCEv2 traffic was 1021. As of WinOF v5.35, the default destination port number used is 4791. This is to comply with The Internet Assigned Numbers Authority (IANA) guidance.

The UDP destination port is a configurable parameter of the driver. For its registry key, please refer to Table 26 - “RoCE Options,” on page 151.

3.1.4.3.1 Driver Upgrade Considerations

Since the default RoCEv2 port is changed in WinOF 5.10.50000, upgrade from an older version that uses the RoCEv2 with the default port will effectively change the port used for RoCEv2. Therefore, on a system that uses an older version with RoCEv2 and the default port, when upgrading to Rev 5.10.50000 or newer, it is advised that the entire group of computers be upgraded at the same time in order to maintain RoCEv2 connectivity.

To allow gradual upgrade without affecting the RoCEv2 connectivity, it is possible to override the default port before upgrade. This can be done by setting the roce_udp_dport parameter to the desired port in the registry so that this port is used by both older and newer versions.

3.1.4.4 Configuring SwitchX® Based Switch System

➢ ***To enable RoCE, the SwitchX should be configured as follows:***

- Ports facing the host should be configured as access ports, and either use global pause or Port Control Protocol (PCP) for priority flow control
- Ports facing the network should be configured as trunk ports, and use Port Control Protocol (PCP) for priority flow control

For further information on how to configure SwitchX, please refer to SwitchX User Manual.

3.1.4.5 Configuring Arista Switch

Step 1. Set the ports that face the hosts as trunk.

```
(config)# interface et10
(config-if-Et10)# switchport mode trunk
```

Step 2. Set VID allowed on trunk port to match the host VID.

```
(config-if-Et10)# switchport trunk allowed vlan 100
```

Step 3. Set the ports that face the network as trunk.

```
(config)# interface et20
(config-if-Et20)# switchport mode trunk
```

Step 4. Assign the relevant ports to LAG.

```
(config)# interface et10
(config-if-Et10)# dcbx mode ieee
(config-if-Et10)# speed forced 40gfull
(config-if-Et10)# channel-group 11 mode active
```

Step 5. Enable PFC on ports that face the network.

```
(config)# interface et20
(config-if-Et20)# load-interval 5
(config-if-Et20)# speed forced 40gfull
(config-if-Et20)# switchport trunk native vlan tag
(config-if-Et20)# switchport trunk allowed vlan 11
(config-if-Et20)# switchport mode trunk
(config-if-Et20)# dcbx mode ieee
(config-if-Et20)# priority-flow-control mode on
(config-if-Et20)# priority-flow-control priority 3 no-drop
```

3.1.4.5.1 Using Global Pause (Flow Control)

➢ ***To enable Global Pause on ports that face the hosts, perform the following:***

```
(config)# interface et10
(config-if-Et10)# flowcontrol receive on
(config-if-Et10)# flowcontrol send on
```

3.1.4.5.2 Using Priority Flow Control (PFC)

➢ ***To enable Global Pause on ports that face the hosts, perform the following:***

```
(config)# interface et10
(config-if-Et10)# dcbx mode ieee
(config-if-Et10)# priority-flow-control mode on
(config-if-Et10)# priority-flow-control priority 3 no-drop
```

3.1.4.6 Configuring Router (PFC only)

The router uses L3's DSCP value to mark the egress traffic of L2 PCP. The required mapping, maps the three most significant bits of the DSCP into the PCP. This is the default behavior, and no additional configuration is required.

3.1.4.6.1 Copying Port Control Protocol (PCP) between Subnets

The captured PCP option from the Ethernet header of the incoming packet can be used to set the PCP bits on the outgoing Ethernet header.

3.1.4.7 Configuring the RoCE Mode

Configuring the RoCE mode requires the following:

- RoCE mode is configured per-driver and is enforced on all the devices in the system

> The supported RoCE modes depend on the firmware installed. If the firmware does not support the needed mode, the fallback mode would be the maximum supported RoCE mode of the installed NIC.

RoCE mode can be enabled and disabled via PowerShell.

➢ ***To enable RoCEv1 using the PowerShell:***

- Open the PowerShell and run:

```
PS $ Set-MlnxDriverCoreSetting -RoceMode 1
```

➢ ***To enable RoCEv2 using the PowerShell:***

- Open the PowerShell and run:

```
PS $ Set-MlnxDriverCoreSetting -RoceMode 2
```

➢ ***To disable any version of RoCE using the PowerShell:***

- Open the PowerShell and run:

```
PS $ Set-MlnxDriverCoreSetting -RoceMode 0
```

➢ ***To check the current version of RoCE using the PowerShell:***

- Open the PowerShell and run:

```
PS $ Get-MlnxDriverCoreSetting
```

- Example output:

```
Caption                : DriverCoreSettingData 'mlx4_bus'
Description            : Mellanox Driver Option Settings
.
.
.
RoceMode               : 0
```

3.1.4.8 RoCE Default Configuration

Starting from v5.20, the default RoCE mode will be RoCE v2.

The RoCE mode will be set to RoCE v2 only upon the first installation of the package.

In any other case, the RoCE mode will not be changed.

3.1.5 Teaming and VLAN

Windows Server 2012 and above supports Teaming as part of the operating system. Please refer to Microsoft guide "NIC Teaming in Windows Server 2012" following the link below:

https://technet.microsoft.com/en-us/windows-server-docs/networking/technologies/nic-teaming/nic-teaming?f=255&MSPPError=-2147217396#bkmk_over

http://www.microsoft.com/en-us/download/confirmation.aspx?id=40319

Mellanox WinOF drivers provide teaming solutions for Windows Server 2008 R2 operating system and client operating systems namely Windows 7 and Windows 8.1.

3.1.5.1 System Requirements

Ethernet teaming is supported only in Windows 7 Client, Windows 8.1 client and Windows Server 2008 R2.

3.1.5.2 Adapter Teaming

Adapter teaming can group a set of ports inside a network adapter 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.

Teaming Types

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 performs 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.

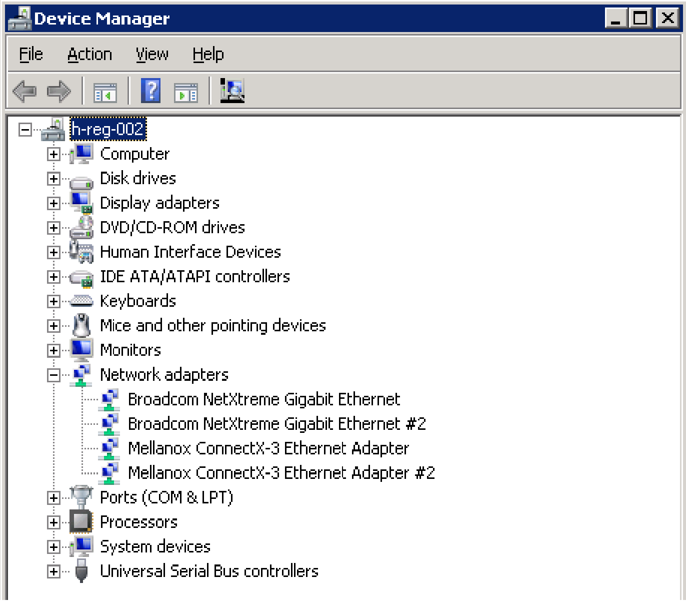
3.1.5.2.1 Creating a Team

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

How to Create a Team

➢ ***The following steps describe the process of creating a team:***

Step 1. Display the Device Manager.



- h-reg-002
  - Computer
  - Disk drives
  - Display adapters
  - DVD/CD-ROM drives
  - Human Interface Devices
  - IDE ATA/ATAPI controllers
  - Keyboards
  - Mice and other pointing devices
  - Monitors
  - Network adapters
    - Broadcom NetXtreme Gigabit Ethernet
    - Broadcom NetXtreme Gigabit Ethernet #2
    - Mellanox ConnectX-3 Ethernet Adapter
    - Mellanox ConnectX-3 Ethernet Adapter #2
  - Ports (COM & LPT)
  - Processors
  - System devices
  - Universal Serial Bus controllers

Step 2. Right-click one of Mellanox ConnectX Ethernet adapters (under "Network adapters" list) and left click Properties. Select the Teaming tab from the Properties window.

> It is not recommended to open the Properties window of more than one adapter at the same time.

Teaming dialog enables creating, modifying or removing a team. Note that only Mellanox Technologies adapters can be part of the team.

➢ ***To create a new team, perform the following***

Step 1. Click Create.

Step 2. Enter a (unique) team name.

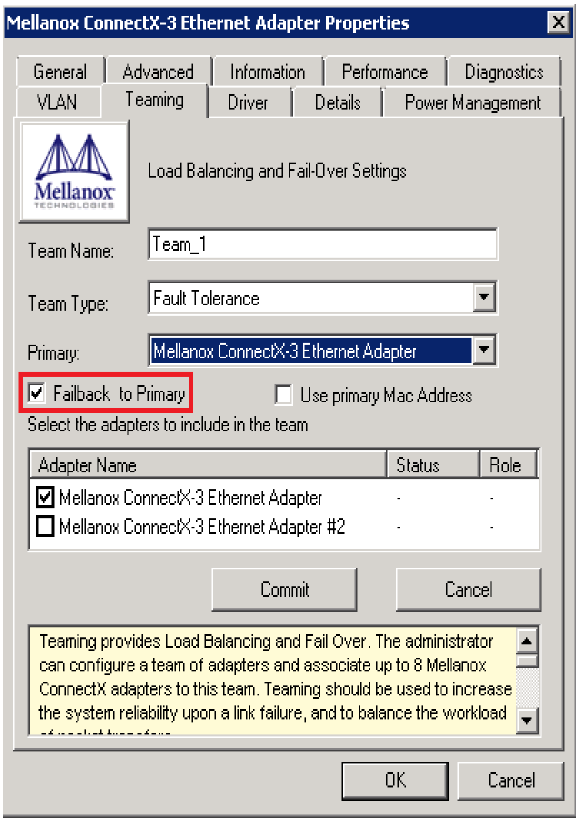
Step 3. Select a team type.

Step 4. Select the adapters to be included in the team (that have not been associated with a VLAN).

Step 5. [Optional] Select Primary Adapter
A failover team type implements an active-passive scenario where only one interface is active at any given time. When the active one is disconnected, one of the other interfaces becomes active. When the primary link comes up, the team interface returns to transfer data using the primary interface. If the primary adapter is not selected, the primary interface is selected randomly.

Step 6. [Optional] Failback to Primary
The Failback to Primary option (checked box) specifies that the team will switch to the primary adapter even though the active adapter can continue functioning as the active one.

When the checkbox is unchecked, the active adapter will remain active even though the primary can function as the active one.



Load Balancing and Fail-Over Settings

Team Name: Team_1

Team Type: Fault Tolerance

Primary: Mellanox ConnectX-3 Ethernet Adapter

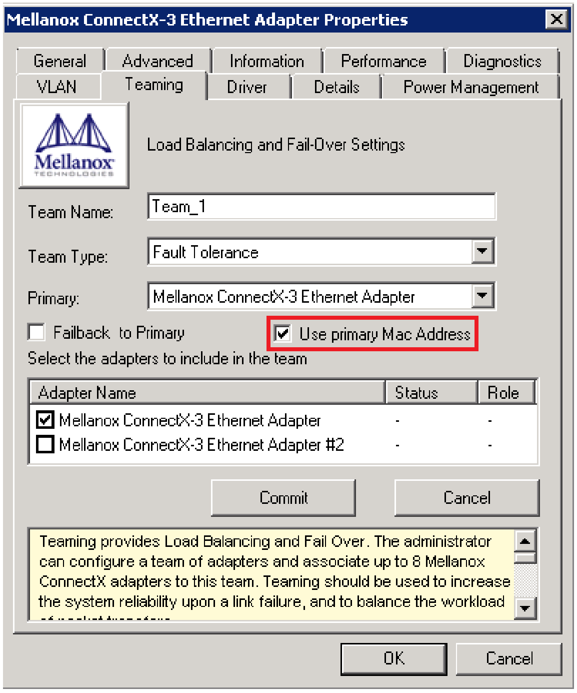
☑ Failback to Primary □ Use primary Mac Address

Select the adapters to include in the team

Adapter Name	Status	Role
☑ Mellanox ConnectX-3 Ethernet Adapter	-	-
□ Mellanox ConnectX-3 Ethernet Adapter #2	-	-

Teaming provides Load Balancing and Fail Over. The administrator can configure a team of adapters and associate up to 8 Mellanox ConnectX adapters to this team. Teaming should be used to increase the system reliability upon a link failure, and to balance the workload of packet transfers

Step 7. [Optional] Primary MAC Address
This option sets the team MAC address to be the same as the primary adapter MAC address.



Mellanox ConnectX-3 Ethernet Adapter Properties

Load Balancing and Fail-Over Settings

Team Name: Team_1

Team Type: Fault Tolerance

Primary: Mellanox ConnectX-3 Ethernet Adapter

□ Failback to Primary ☑ Use primary Mac Address

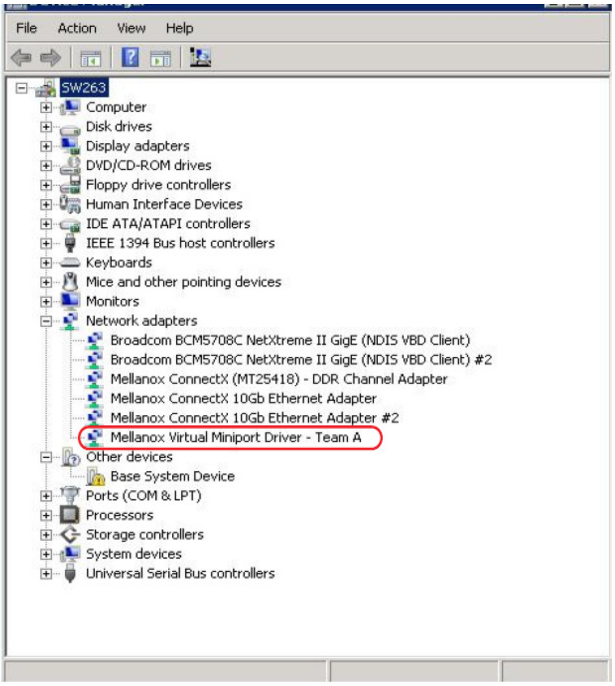
Select the adapters to include in the team

Adapter Name	Status	Role
☑ Mellanox ConnectX-3 Ethernet Adapter	-	-
□ Mellanox ConnectX-3 Ethernet Adapter #2	-	-

Teaming provides Load Balancing and Fail Over. The administrator can configure a team of adapters and associate up to 8 Mellanox ConnectX adapters to this team. Teaming should be used to increase the system reliability upon a link failure, and to balance the workload

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

```
Mellanox Virtual Miniport Driver - Team <team_name>
```



- SW263
  - Computer
  - Disk drives
  - Display adapters
  - DVD/CD-ROM drives
  - Floppy drive controllers
  - Human Interface Devices
  - IDE ATA/ATAPI controllers
  - IEEE 1394 Bus host controllers
  - Keyboards
  - Mice and other pointing devices
  - Monitors
  - Network adapters
    - Broadcom BCM5708C NetXtreme II GigE (NDIS VBD Client)
    - Broadcom BCM5708C NetXtreme II GigE (NDIS VBD Client) #2
    - Mellanox ConnectX (MT25418) - DDR Channel Adapter
    - Mellanox ConnectX 10Gb Ethernet Adapter
    - Mellanox ConnectX 10Gb Ethernet Adapter #2
    - Mellanox Virtual Miniport Driver - Team A
  - Other devices
    - Base System Device
  - Ports (COM & LPT)
  - Processors
  - Storage controllers
  - System devices
  - Universal Serial Bus controllers

➢ ***To modify an existing team, perform the following:***

a. Select the desired team and click Modify

b. Modify the team name, its type, and/or the participating adapters

c. Click the Commit button

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

Notes on this step:

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

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

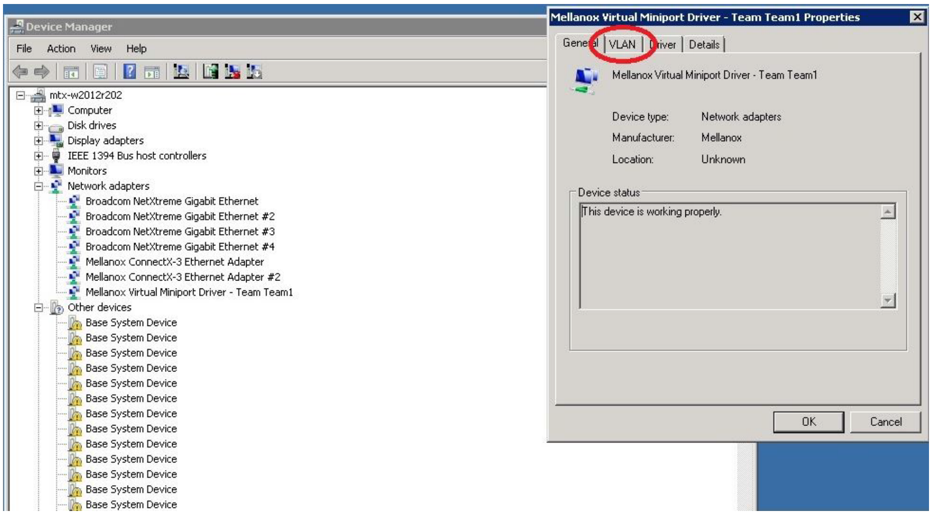
b. Each network adapter that is added or removed from a team gets refreshed (i.e. disabled then enabled). This may cause a temporary loss of connection to the adapter.

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

3.1.5.2.2 VLAN Configuration to a Team

In order to configure a VLAN to a team, follow the steps below:

1. Open the Device Manager.
2. Go to Network Adapters
3. Right click on the Team Adapter that was created and click on the VLAN tab.



mtx-w2012r202
- Computer
- Disk drives
- Display adapters
- IEEE 1394 Bus host controllers
- Monitors
- Network adapters
  - Broadcom NetXtreme Gigabit Ethernet
  - Broadcom NetXtreme Gigabit Ethernet #2
  - Broadcom NetXtreme Gigabit Ethernet #3
  - Broadcom NetXtreme Gigabit Ethernet #4
  - Mellanox ConnectX-3 Ethernet Adapter
  - Mellanox ConnectX-3 Ethernet Adapter #2
  - Mellanox Virtual Miniport Driver - Team Team1
- Other devices
  - Base System Device
  - Base System Device
  - Base System Device
  - Base System Device
  - Base System Device
  - Base System Device
  - Base System Device
  - Base System Device
  - Base System Device
  - Base System Device
  - Base System Device
  - Base System Device
  - Base System Device

Mellanox Virtual Miniport Driver - Team Team1

Device type: Network adapters

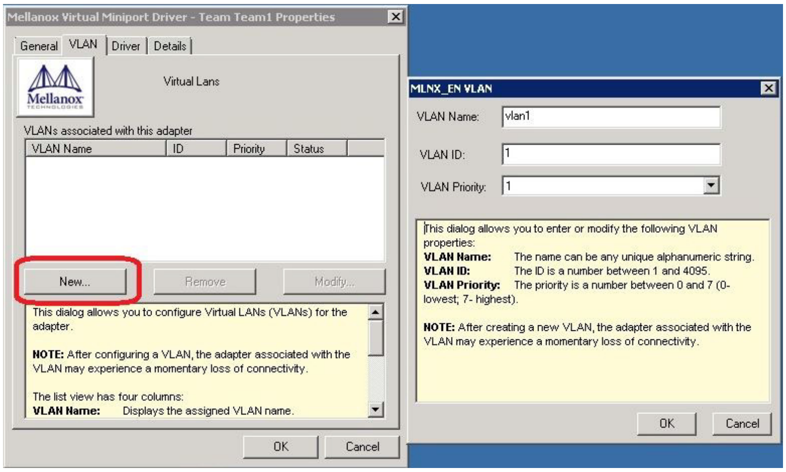
Manufacturer: Mellanox

Location: Unknown

Device status

This device is working properly.

4. In the VLAN tab, click on “New” and fill up the details.



This dialog allows you to configure Virtual LANs (VLANs) for the adapter.

NOTE: After configuring a VLAN, the adapter associated with the VLAN may experience a momentary loss of connectivity.

The list view has four columns:

VLAN Name: Displays the assigned VLAN name.

VLAN Name: vlan1

VLAN ID: 1

VLAN Priority: 1

This dialog allows you to enter or modify the following VLAN properties:

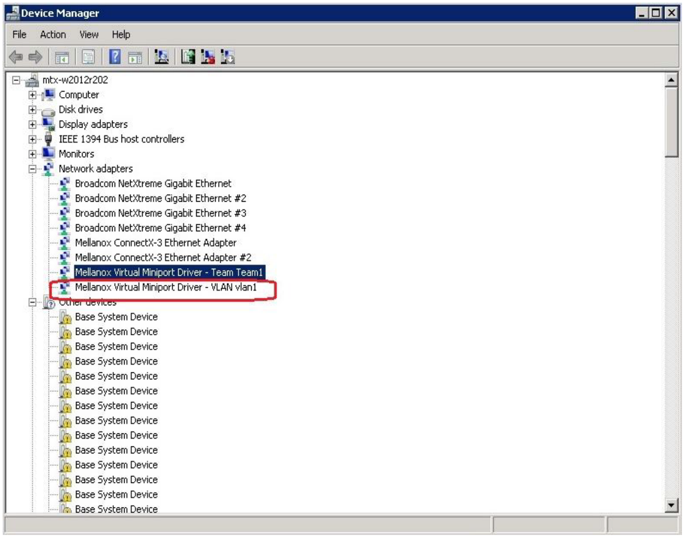
VLAN Name: The name can be any unique alphanumeric string.

VLAN ID: The ID is a number between 1 and 4095.

VLAN Priority: The priority is a number between 0 and 7 (0-lowest; 7- highest).

NOTE: After creating a new VLAN, the adapter associated with the VLAN may experience a momentary loss of connectivity.

5. The newly created VLAN interface will appear as can be seen below:

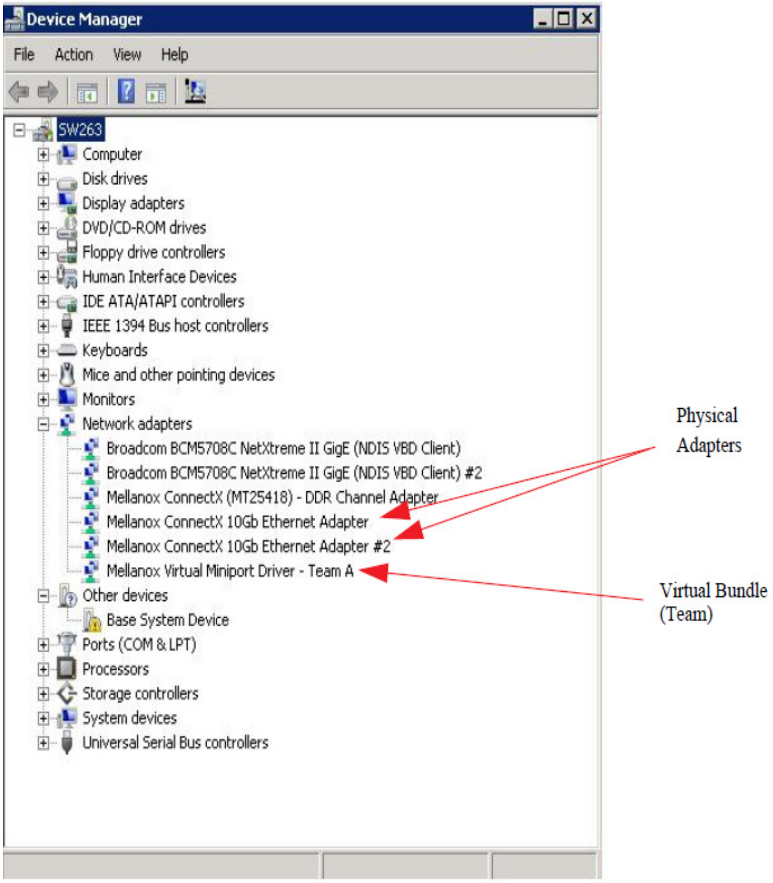


```
mtx-w2012r202
  Computer
  Disk drives
  Display adapters
  IEEE 1394 Bus host controllers
  Monitors
  Network adapters
    Broadcom NetXtreme Gigabit Ethernet
    Broadcom NetXtreme Gigabit Ethernet #2
    Broadcom NetXtreme Gigabit Ethernet #3
    Broadcom NetXtreme Gigabit Ethernet #4
    Mellanox ConnectX-3 Ethernet Adapter
    Mellanox ConnectX-3 Ethernet Adapter #2
    Mellanox Virtual Miniport Driver - Team Team1
    Mellanox Virtual Miniport Driver - VLAN vlan1
  Other devices
    Base System Device
    Base System Device
    Base System Device
    Base System Device
    Base System Device
    Base System Device
    Base System Device
    Base System Device
    Base System Device
    Base System Device
    Base System Device
    Base System Device
    Base System Device
    Base System Device
```

3.1.5.2.3 Creating a Port VLAN in Windows Server 2008 R2

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

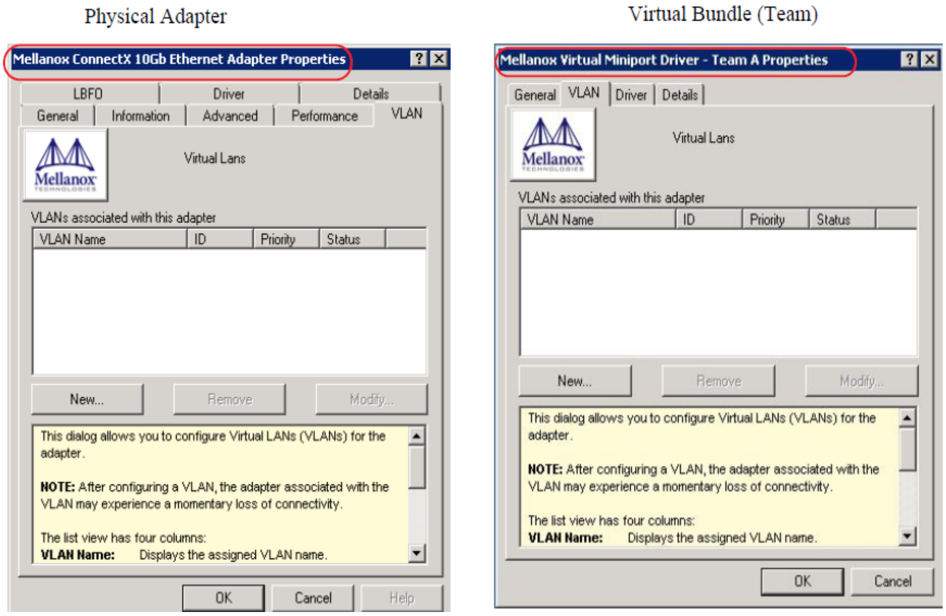
Step 1. Display the Device Manager.


Device Manager
File Action View Help
SW263
Computer
Disk drives
Display adapters
DVD/CD-ROM drives
Floppy drive controllers
Human Interface Devices
IDE ATA/ATAPI controllers
IEEE 1394 Bus host controllers
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Broadcom BCM5708C NetXtreme II GigE (NDIS VBD Client)
Broadcom BCM5708C NetXtreme II GigE (NDIS VBD Client) #2
Mellanox ConnectX (MT25418) - DDR Channel Adapter
Mellanox ConnectX 10Gb Ethernet Adapter
Mellanox ConnectX 10Gb Ethernet Adapter #2
Mellanox Virtual Miniport Driver - Team A
Other devices
Base System Device
Ports (COM & LPT)
Processors
Storage controllers
System devices
Universal Serial Bus controllers

Physical Adapters

Virtual Bundle (Team)

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)

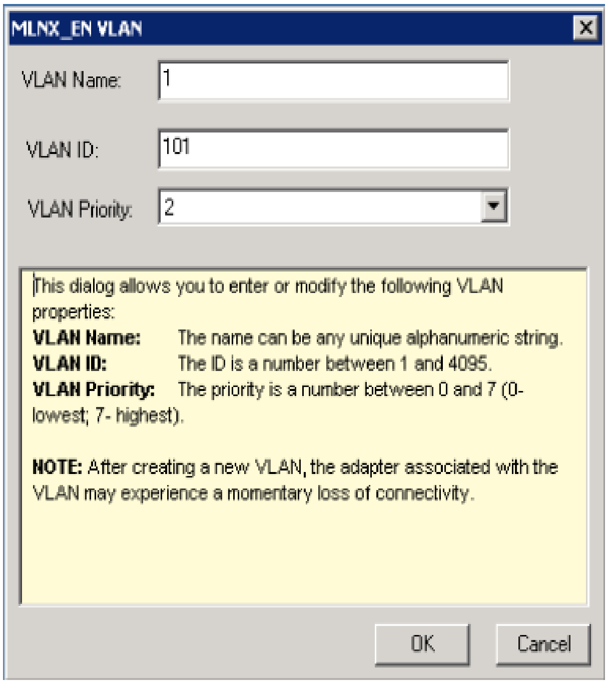
This dialog allows you to configure Virtual LANs (VLANs) for the adapter.

NOTE: After configuring a VLAN, the adapter associated with the VLAN may experience a momentary loss of connectivity.

The list view has four columns:
VLAN Name: Displays the assigned VLAN name.

If a physical adapter has been added to a team, 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.



VLAN Name: 1

VLAN ID: 101

VLAN Priority: 2

This dialog allows you to enter or modify the following VLAN properties:
VLAN Name: The name can be any unique alphanumeric string.
VLAN ID: The ID is a number between 1 and 4095.
VLAN Priority: The priority is a number between 0 and 7 (0-lowest; 7- highest).

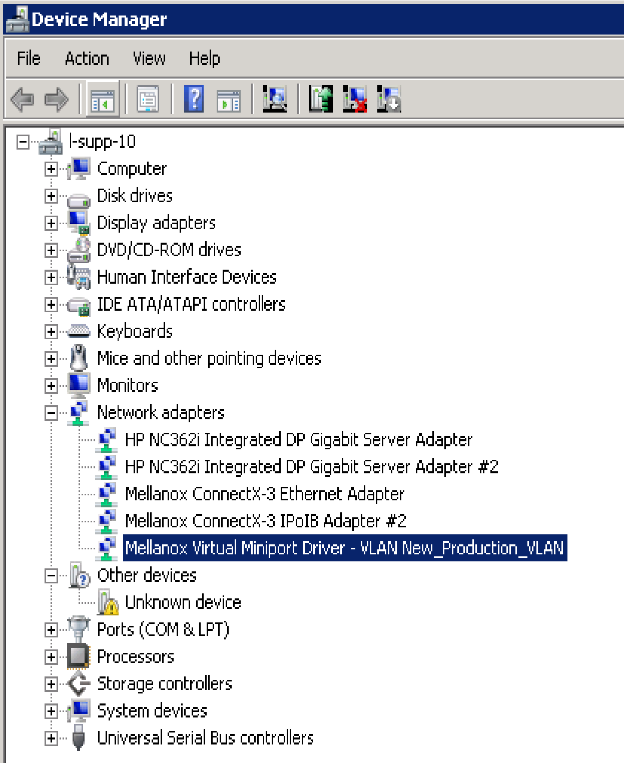
NOTE: After creating a new VLAN, the adapter associated with the VLAN may experience a momentary loss of connectivity.

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>
```



l-supp-10
- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Human Interface Devices
- IDE ATA/ATAPI controllers
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
  - HP NC362i Integrated DP Gigabit Server Adapter
  - HP NC362i Integrated DP Gigabit Server Adapter #2
  - Mellanox ConnectX-3 Ethernet Adapter
  - Mellanox ConnectX-3 IPoIB Adapter #2
  - Mellanox Virtual Miniport Driver - VLAN New_Production_VLAN
- Other devices
  - Unknown device
- Ports (COM & LPT)
- Processors
- Storage controllers
- System devices
- Universal Serial Bus controllers

3.1.5.2.4 Removing a Port VLAN in Windows Server 2008 R2

➢ ***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.

3.1.5.2.5 Configuring a Port to Work with VLAN in Windows Server 2012 and Above

> In this procedure you DO NOT create a VLAN, rather use an existing VLAN ID.

➢ ***To configure a port to work with VLAN using the Device Manager.***

Step 1. Open the Device Manager.

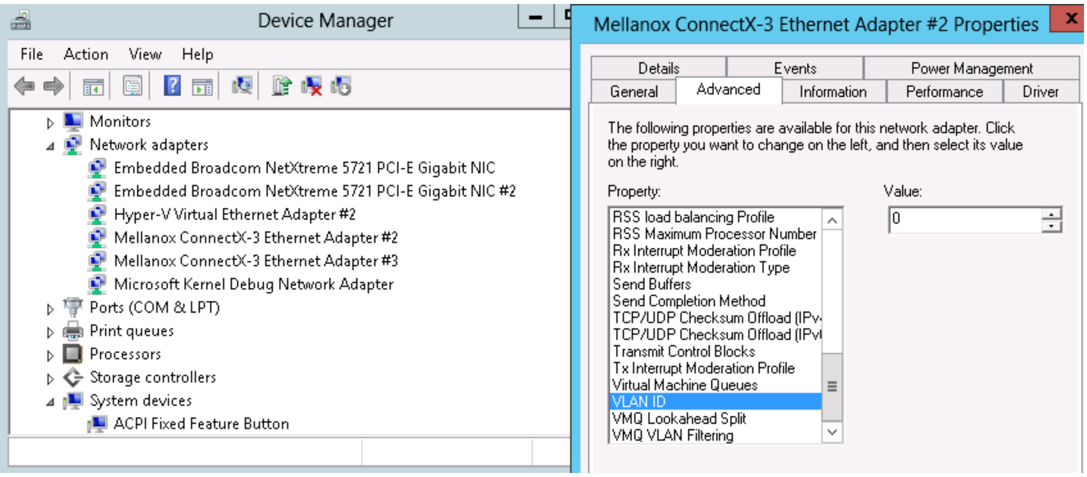
Step 2. Go to the Network adapters.

Step 3. Right click Properties on Mellanox ConnectX®-3 Ethernet Adapter card.

Step 4. Go to Advanced tab.

Step 5. Choose the VLAN ID in the Property window.

Step 6. Set its value in the Value window.


Device Manager
File
Action
View
Help
Monitors
Network adapters
Embedded Broadcom NetXtreme 5721 PCI-E Gigabit NIC
Embedded Broadcom NetXtreme 5721 PCI-E Gigabit NIC #2
Hyper-V Virtual Ethernet Adapter #2
Mellanox ConnectX-3 Ethernet Adapter #2
Mellanox ConnectX-3 Ethernet Adapter #3
Microsoft Kernel Debug Network Adapter
Ports (COM & LPT)
Print queues
Processors
Storage controllers
System devices
ACPI Fixed Feature Button
Mellanox ConnectX-3 Ethernet Adapter #2 Properties
Details
Events
Power Management
General
Advanced
Information
Performance
Driver

The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property:

RSS load balancing Profile
RSS Maximum Processor Number
Rx Interrupt Moderation Profile
Rx Interrupt Moderation Type
Send Buffers
Send Completion Method
TCP/UDP Checksum Offload (IPv
TCP/UDP Checksum Offload (IPv
Transmit Control Blocks
Tx Interrupt Moderation Profile
Virtual Machine Queues
VLAN ID
VMQ Lookahead Split
VMQ VLAN Filtering

Value: 0

3.1.6 Command Line Based Teaming Configuration

3.1.6.1 Show help

The following command prints out all supported modes and functionalities:

```
  C:\Users\Administrator\Desktop>vlan_config --help
  To list all adapters including teams, use:
         vlan_config showlist [IPoIB]
  To create a team use:
      vlan_config create team <Type> <Name> [NoFailBackToPrimary] [IPoIB]
          Type is one of the following: AFT | SFT | SLB | RLB | ALB | 3AD | SLA
          For IPoIB team, only 'AFT' type is supported
  To add adapter to the team use:
      vlan_config attach team <TeamName> {<Adapter-GUID>} [primary] [SetTeamMacAddress]
  To remove an adapter from the team use:
      vlan_config detach team <TeamName> {<Adapter-GUID>}
  To delete an empty team use:
      vlan_config removeteam <TeamName>
  To query an existing team, use:
         vlan_config queryteam <TeamName>
  To modify an existing team, use:
         vlan_config modifyteam <TeamName> <NewTeamName> <TeamType>
  To add a vlan use:
      vlan_config addvlan <TeamName> <VlanName> <VlanId> <Priority>
  To remove a vlan use:
      vlan_config removevlan {TeamName} {VlanName}
  To query vlan, use:
         vlan_config queryvlan <TeamName> <VlanName>
  To modify vlan, use:
        vlan_config modifyvlan <team-name> <current-vlan-name> <newvlanname> <newvlanid>
<newpriority>

  Example:
      vlan_config create team AFT MyTeam
     vlan_config attach team MyTeam {2E9C1992-98B5-43C3-97A0-9993AEAC7F80}
     vlan_config attach team MyTeam {8D05C52B-BCD6-4FCE-8235-1E90BD334519}
```

3.1.6.2 Show all Adapters (including already created teams)

```
 C:\Users\Administrator\Desktop>vlan_config showlist
 {30C354DB-62E9-41CB-B709-11063FAF4E95}          Mellanox ConnectX-3 Ethernet Adapter
 {53AE4E6B-A308-4C74-9791-153BB3104509}         Mellanox ConnectX-3 Ethernet Adapter #2
```

3.1.6.3 Create an Empty Team

```
 C:\Users\Administrator\Desktop>vlan_config create team AFT MyTeam
 Adding team MyTeam
 Team created with Guid = C71781EC-F459-4A8E-ABAC-74CA05F13AE8
```

3.1.6.4 Attach Members to Team

```
 C:\Users\Administrator\Desktop>vlan_config attach team MyTeam {30C354DB-62E9-41CB-
B709-11063FAF4E95} primary setteammacaddress
 Adding adapter {30C354DB-62E9-41CB-B709-11063FAF4E95} to team MyTeam

 C:\Users\Administrator\Desktop>vlan_config attach team MyTeam {53AE4E6B-A308-4C74-
9791-153BB3104509}
 Adding adapter {53AE4E6B-A308-4C74-9791-153BB3104509} to team MyTeam
```

3.1.6.5 Create a VLAN

```
 C:\Users\Administrator\Desktop>vlan_config addvlan MyTeam MyVlan 1 1
 Adding vlan to adapter with Guid ={C71781EC-F459-4A8E-ABAC-74CA05F13AE8}
 Vlan Name=MyVlan
 Vlan Id=1
 Vlan Priority=1
 Miniport created with Guid = 3D7CFE4E-0EE9-4CF6-9843-F44CFBE2E223
```

3.1.6.6 Modify a Team

```
 C:\Users\Administrator\Desktop>vlan_config modifyteam MyTeam NewTeam SFT2
 New Attributes applied
```

3.1.6.7 Modify a VLAN

```
 C:\Users\Administrator\Desktop>vlan_config modifyvlan NewTeam MyVlan NewVlan 3 4
 New Attributes applied
```

3.1.6.8 ShowList (including the team created)

```
 C:\Users\Administrator\Desktop>vlan_config showlist
 {30C354DB-62E9-41CB-B709-11063FAF4E95}          Mellanox ConnectX-3 Ethernet Adapter
 {53AE4E6B-A308-4C74-9791-153BB3104509}         Mellanox ConnectX-3 Ethernet Adapter #2
```

```
Found 1 team(s)

 Name                 : NewTeam
 GUID                 : {C71781EC-F459-4A8E-ABAC-74CA05F13AE8}
 PrimaryAdapterGuid : {30C354DB-62E9-41CB-B709-11063FAF4E95}
 TeamType             : 2
 L2Protocol           : 0
 FallBackToPrimary  : 1
 MemberCount          : 2
 Member[0]            : {30C354DB-62E9-41CB-B709-11063FAF4E95}
 Member[1]            : {53AE4E6B-A308-4C74-9791-153BB3104509}
 VlanCount            : 1
 Vlan[0] Name         : NewVlan
```

3.1.6.9 QueryTeam

```
C:\Users\Administrator\Desktop>vlan_config queryteam NewTeam

Found 1 team(s)

 Name                : NewTeam
 GUID                : {C71781EC-F459-4A8E-ABAC-74CA05F13AE8}
 PrimaryAdapterGuid : {30C354DB-62E9-41CB-B709-11063FAF4E95}
 TeamType            : 2
 L2Protocol          : 0
 FallBackToPrimary  : 1
 MemberCount         : 2
 Member[0]           : {30C354DB-62E9-41CB-B709-11063FAF4E95}
 Member[1]           : {53AE4E6B-A308-4C74-9791-153BB3104509}
 VlanCount           : 1
 Vlan[0] Name        : NewVlan
```

3.1.6.10 QueryVlan

```
C:\Users\Administrator\Desktop>vlan_config queryvlan NewTeam NewVlan
 Vlan Guid       : {3D7CFE4E-0EE9-4CF6-9843-F44CFBE2E223}
 Vlan Name       : NewVlan
 Vlan Id         : 3
 Vlan Priority   : 4
```

3.1.6.11 Remove Vlan

```
C:\Users\Administrator\Desktop>vlan_config removevlan NewTeam NewVlan
removing vlan to adapter with physical Guid ={C71781EC-F459-4A8E-ABAC-74CA05F13AE8}
Vlan Guid ={3D7CFE4E-0EE9-4CF6-9843-F44CFBE2E223}
pMuxPhysicalAdapter->RemoveMiniport succeeded
```

3.1.6.12 Remove an Entire Team

```
C:\Users\Administrator\Desktop>vlan_config removeteam NewTeam
Delete team NewTeam
Deleting member {30C354DB-62E9-41CB-B709-11063FAF4E95}
Deleting member {53AE4E6B-A308-4C74-9791-153BB3104509}
```

3.1.6.13 Show List Again (Back to the Beginning)

```
C:\Users\Administrator\Desktop>vlan_config showlist
{30C354DB-62E9-41CB-B709-11063FAF4E95}          Mellanox ConnectX-3 Ethernet Adapter
{53AE4E6B-A308-4C74-9791-153BB3104509}          Mellanox ConnectX-3 Ethernet Adapter #2
```

3.1.7 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 window.

For further information, please refer to the MSDN library:

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

3.1.8 Configuring Quality of Service (QoS)

3.1.8.1 System Requirements

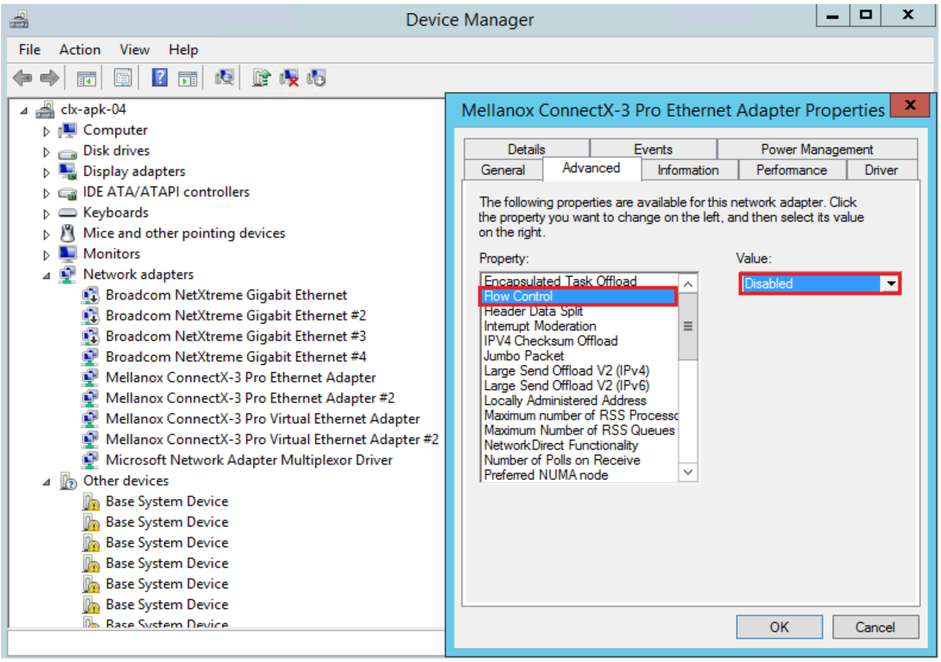
Operating Systems: Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2 and Windows Server 2016.

3.1.8.2 QoS Configuration

Prior to configuring Quality of Service, you must install Data Center Bridging using one of the following methods:

➢ ***To Disable Flow Control Configuration***

Device manager->Network adapters->Mellanox ConnectX-3 Ethernet Adapter->Properties->Advanced tab



Mellanox ConnectX-3 Pro Ethernet Adapter Properties

The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property: Flow Control

Value: Disabled

Device Manager
File
Action
View
Help
clx-apk-04
Computer
Disk drives
Display adapters
IDE ATA/ATAPI controllers
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Broadcom NetXtreme Gigabit Ethernet
Broadcom NetXtreme Gigabit Ethernet #2
Broadcom NetXtreme Gigabit Ethernet #3
Broadcom NetXtreme Gigabit Ethernet #4
Mellanox ConnectX-3 Pro Ethernet Adapter
Mellanox ConnectX-3 Pro Ethernet Adapter #2
Mellanox ConnectX-3 Pro Virtual Ethernet Adapter
Mellanox ConnectX-3 Pro Virtual Ethernet Adapter #2
Microsoft Network Adapter Multiplexor Driver
Other devices
Base System Device
Base System Device
Base System Device
Base System Device
Base System Device
Base System Device
Base System Device
Details
Events
Power Management
General
Advanced
Information
Performance
Driver
Encapsulated Task Offload
Flow Control
Header Data Split
Interrupt Moderation
IPV4 Checksum Offload
Jumbo Packet
Large Send Offload V2 (IPv4)
Large Send Offload V2 (IPv6)
Locally Administered Address
Maximum number of RSS Processo
Maximum Number of RSS Queues
NetworkDirect Functionality
Number of Polls on Receive
Preferred NUMA node
OK
Cancel

➢ *To install the Data Center Bridging using the Server Manager:*

Step 1. Open the 'Server Manager'.

Step 2. Select 'Add Roles and Features'.

Step 3. Click Next.

Step 4. Select 'Features' on the left panel.

Step 5. Check the 'Data Center Bridging' checkbox.

Step 6. Click 'Install'.

➢ *To install the Data Center Bridging using PowerShell:*

Step 1. Enable Data Center Bridging (DCB).

```
PS $ Install-WindowsFeature Data-Center-Bridging
```

➢ *To configure QoS on the host:*

> The procedure below is not saved after you reboot your system. Hence, we recommend you create a script using the steps below and run it on the startup of the local machine.
> Please see the procedure below on how to add the script to the local machine startup scripts.

Step 1. Change the Windows PowerShell execution policy. To change the execution policy, please refer to Step 1 in Section 3.3.1, “PowerShell Configuration”, on page 91

Step 2. Remove the entire previous QoS configuration.

```
PS $ Remove-NetQosTrafficClass
PS $ Remove-NetQosPolicy -Confirm:$False
```

Step 3. Set the DCBX Willing parameter to false as Mellanox drivers do not support this feature.

```
PS $ set-NetQosDcbxSetting -Willing 0
```

Step 4. Create a Quality of Service (QoS) policy and tag each type of traffic with the relevant priority.

In this example, TCP/UDP use priority 1, SMB over TCP use priority 3.

```
PS $ New-NetQosPolicy "DEFAULT" -store Activestore -Default -PriorityValue8021Action 3
PS $ New-NetQosPolicy "TCP" -store Activestore -IPProtocolMatchCondition TCP -Priority-
Value8021Action 1
PS $ New-NetQosPolicy "UDP" -store Activestore -IPProtocolMatchCondition UDP -Priority-
Value8021Action 1
New-NetQosPolicy “SMB” -SMB -PriorityValue8021Action 3
```

Step 5. Create a QoS policy for SMB over SMB Direct traffic on Network Direct port 445.

```
PS $ New-NetQosPolicy "SMBDirect" -store Activestore -NetDirectPortMatchCondition 445 -
PriorityValue8021Action 3
```

Step 6. [Optional] If VLANs are used, mark the egress traffic with the relevant VlanID.
The NIC is referred as "Ethernet 4” in the examples below.

```
PS $ Set-NetAdapterAdvancedProperty -Name "Ethernet 4" -RegistryKeyword "VlanID" -Reg-
istryValue "55"
```

Step 7. [Optional] Configure the IP address for the NIC.

If DHCP is used, the IP address will be assigned automatically.

```
PS $ Set-NetIPInterface -InterfaceAlias “Ethernet 4” -DHCP Disabled
PS $ Remove-NetIPAddress -InterfaceAlias “Ethernet 4” -AddressFamily IPv4 -Con-
firm:$false
PS $ New-NetIPAddress -InterfaceAlias “Ethernet 4” -IPAddress 192.168.1.10 -Prefix-
Length 24 -Type Unicast
```

Step 8. [Optional] Set the DNS server (assuming its IP address is 192.168.1.2).

```
PS $ Set-DnsClientServerAddress -InterfaceAlias “Ethernet 4” -ServerAddresses
192.168.1.2
```

> After establishing the priorities of ND/NDK traffic, the priorities must have PFC enabled on them.

Step 9. Disable Priority Flow Control (PFC) for all other priorities except for 3.

```
PS $ Disable-NetQosFlowControl 0,1,2,4,5,6,7
```

Step 10. Enable QoS on the relevant interface.

```
PS $ Enable-NetAdapterQos -InterfaceAlias "Ethernet 4"
```

Step 11. Enable PFC on priority 3.

```
PS $ Enable-NetQosFlowControl -Priority 3
```

Step 12. Configure Priority 3 to use ETS.

```
PS $ New-NetQosTrafficClass -name "SMB class" -priority 3 -bandwidthPercentage 50 -
Algorithm ETS
```

➢ ***To add the script to the local machine startup scripts:***

Step 1. From the PowerShell invoke.

```
gpedit.msc
```

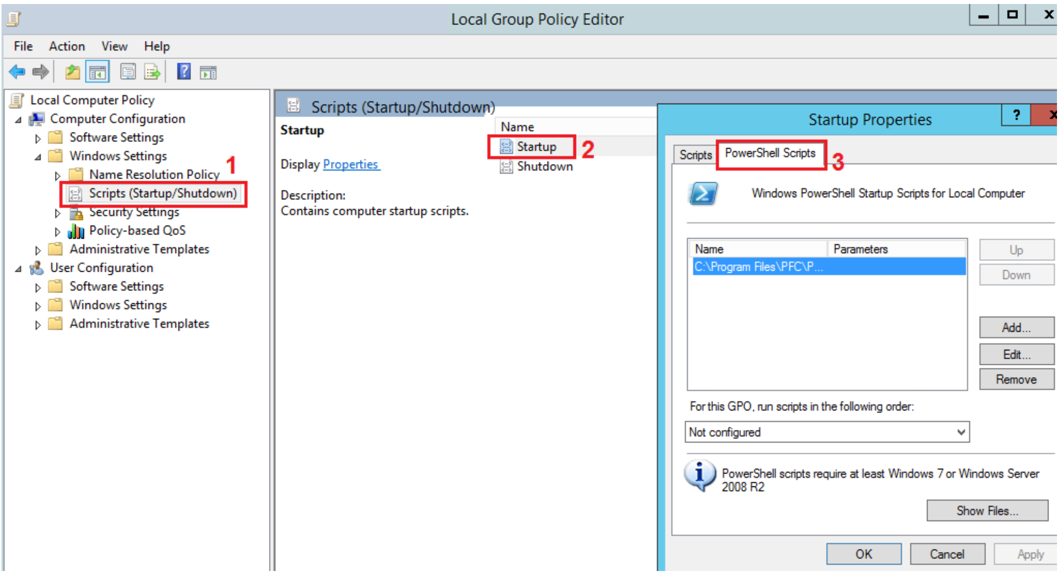
Step 2. In the pop-up window, under the 'Computer Configuration' section, perform the following:

1. Select Windows Settings

2. Select Scripts (Startup/Shutdown)

3. Double click Startup to open the Startup Properties

4. Move to “PowerShell Scripts” tab



5. Click Add

The script should include only the following commands:

```
PS $ Remove-NetQosTrafficClass
PS $ Remove-NetQosPolicy -Confirm:$False
PS $ set-NetQosDcbxSetting -Willing 0
PS $ New-NetQosPolicy "SMB" -Policystore Activestore -NetDirectPortMatchCondition 445 -
PriorityValue8021Action 3
PS $ New-NetQosPolicy "DEFAULT" -Policystore Activestore -Default -PriorityValue8021Ac-
tion 3
PS $ New-NetQosPolicy "TCP" -Policystore Activestore -IPProtocolMatchCondition TCP -
PriorityValue8021Action 1
PS $ New-NetQosPolicy "UDP" -Policystore Activestore -IPProtocolMatchCondition UDP -
PriorityValue8021Action 1
PS $ Disable-NetQosFlowControl 0,1,2,4,5,6,7
PS $ Enable-NetAdapterQos -InterfaceAlias "port1"
PS $ Enable-NetAdapterQos -InterfaceAlias "port2"
PS $ Enable-NetQosFlowControl -Priority 3
PS $ New-NetQosTrafficClass -name "SMB class" -priority 3 -bandwidthPercentage 50 -
Algorithm ETS
```

6. Browse for the script's location.

7. Click OK

8. To confirm the settings applied after boot run:

```
PS $ get-netqospolicy -policystore activestore
```

3.1.8.3 Enhanced Transmission Selection

Enhanced Transmission Selection (ETS) provides a common management framework for assignment of bandwidth to frame priorities as described in the IEEE 802.1Qaz specification:

http://www.ieee802.org/1/files/public/docs2008/az-wadekar-ets-proposal-0608-v1.01.pdf

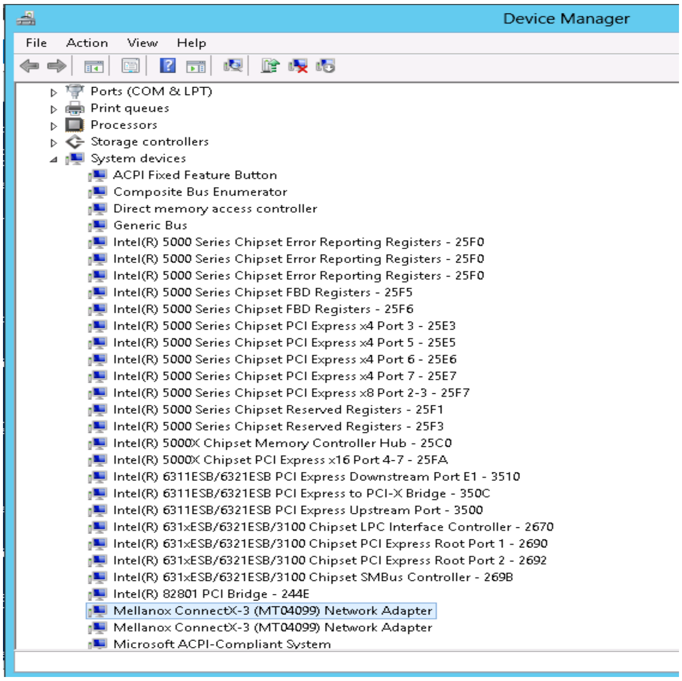
For further details on configuring ETS on Windows™ Server, please refer to:

http://technet.microsoft.com/en-us/library/hh967440.aspx

3.1.9 Configuring the Ethernet Driver

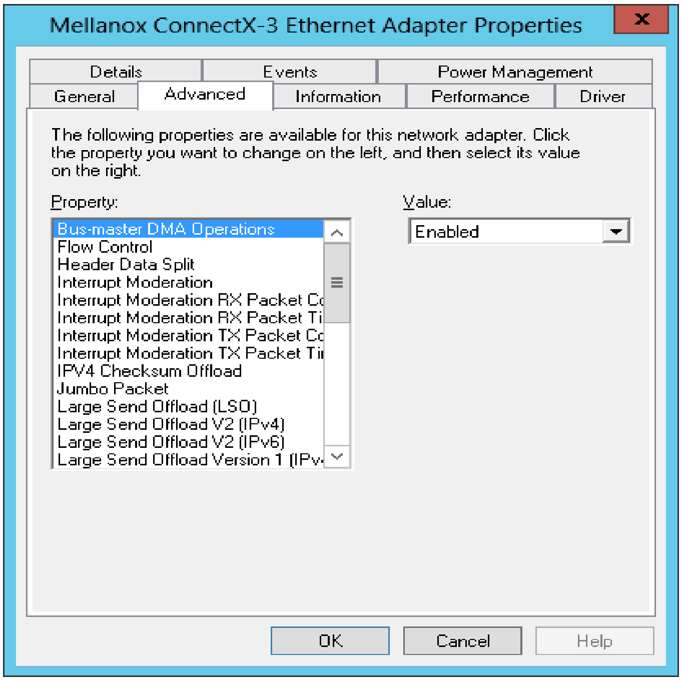
The following steps describe how to configure advanced features.

Step 1. Display the Device Manager.



```
▷ Ports (COM & LPT)
▷ Print queues
▷ Processors
▷ Storage controllers
◢ System devices
    ACPI Fixed Feature Button
    Composite Bus Enumerator
    Direct memory access controller
    Generic Bus
    Intel(R) 5000 Series Chipset Error Reporting Registers - 25F0
    Intel(R) 5000 Series Chipset Error Reporting Registers - 25F0
    Intel(R) 5000 Series Chipset Error Reporting Registers - 25F0
    Intel(R) 5000 Series Chipset FBD Registers - 25F5
    Intel(R) 5000 Series Chipset FBD Registers - 25F6
    Intel(R) 5000 Series Chipset PCI Express x4 Port 3 - 25E3
    Intel(R) 5000 Series Chipset PCI Express x4 Port 5 - 25E5
    Intel(R) 5000 Series Chipset PCI Express x4 Port 6 - 25E6
    Intel(R) 5000 Series Chipset PCI Express x4 Port 7 - 25E7
    Intel(R) 5000 Series Chipset PCI Express x8 Port 2-3 - 25F7
    Intel(R) 5000 Series Chipset Reserved Registers - 25F1
    Intel(R) 5000 Series Chipset Reserved Registers - 25F3
    Intel(R) 5000X Chipset Memory Controller Hub - 25C0
    Intel(R) 5000X Chipset PCI Express x16 Port 4-7 - 25FA
    Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E1 - 3510
    Intel(R) 6311ESB/6321ESB PCI Express to PCI-X Bridge - 350C
    Intel(R) 6311ESB/6321ESB PCI Express Upstream Port - 3500
    Intel(R) 631xESB/6321ESB/3100 Chipset LPC Interface Controller - 2670
    Intel(R) 631xESB/6321ESB/3100 Chipset PCI Express Root Port 1 - 2690
    Intel(R) 631xESB/6321ESB/3100 Chipset PCI Express Root Port 2 - 2692
    Intel(R) 631xESB/6321ESB/3100 Chipset SMBus Controller - 269B
    Intel(R) 82801 PCI Bridge - 244E
    Mellanox ConnectX-3 (MT04099) Network Adapter
    Mellanox ConnectX-3 (MT04099) Network Adapter
    Microsoft ACPI-Compliant System
```

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



The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property:

- Bus-master DMA Operations
- Flow Control
- Header Data Split
- Interrupt Moderation
- Interrupt Moderation RX Packet Co
- Interrupt Moderation RX Packet Ti
- Interrupt Moderation TX Packet Co
- Interrupt Moderation TX Packet Ti
- IPV4 Checksum Offload
- Jumbo Packet
- Large Send Offload (LSO)
- Large Send Offload V2 (IPv4)
- Large Send Offload V2 (IPv6)
- Large Send Offload Version 1 (IPv

Value: Enabled

Step 3. Modify configuration parameters to suit your system.

Please note the following:

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

b. If you select one of the entries Off-load Options, Performance Options, or Flow Control Options, you'll need to click the Properties button to modify parameters via a pop-up dialog.

3.1.10 Differentiated Services Code Point (DSCP)

DSCP is a mechanism used for classifying network traffic on IP networks. It uses the 6-bit Differentiated Services Field (DS or DSCP field) in the IP header for packet classification purposes. Using Layer 3 classification enables you to maintain the same classification semantics beyond local network, across routers.

Every transmitted packet holds the information allowing network devices to map the packet to the appropriate 802.1Qbb CoS. For DSCP based PFC or ETS the packet is marked with a DSCP value in the Differentiated Services (DS) field of the IP header.

3.1.10.1 System Requirements

- Operating Systems: Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2 and Windows Server 2016
- Firmware version: 2.30.8000 or higher

3.1.10.2 Setting the DSCP in the IP Header

Marking DSCP value in the IP header is done differently for IP packets constructed by the NIC (e.g. RDMA traffic) and for packets constructed by the IP stack (e.g. TCP traffic).

- For IP packets generated by the IP stack, the DSCP value is provided by the IP stack. The NIC does not validate the match between DSCP and Class of Service (CoS) values. CoS and DSCP values are expected to be set through standard tools, such as PowerShell command New-NetQosPolicy using PriorityValue8021Action and DSCPAction flags respectively.
- For IP packets generated by the NIC (RDMA), the DSCP value is generated according to the CoS value programmed for the interface. CoS value is set through standard tools, such as PowerShell command New-NetQosPolicy using PriorityValue8021Action flag. The NIC uses a mapping table between the CoS value and the DSCP value configured through the RroceDscpMarkPriorityFlow- Control[0-7] Registry keys

3.1.10.3 Configuring Quality of Service for TCP and RDMA Traffic

Step 1. Verify that DCB is installed and enabled (is not installed by default).

```
PS $ Install-WindowsFeature Data-Center-Bridging
```

Step 2. Import the PowerShell modules that are required to configure DCB.

```
PS $ import-module NetQos
PS $ import-module DcbQos
PS $ import-module NetAdapter
```

Step 3. Configure DCB.

```
PS $ Set-NetQosDcbxSetting -Willing 0
```

Step 4. Enable Network Adapter QoS.

```
PS $ Set-NetAdapterQos -Name "Cx3Pro_ETH_P1" -Enabled 1
```

Step 5. Enable Priority Flow Control (PFC) on the specific priority 3,5.

```
PS $ Enable-NetQosFlowControl 3,5
```

3.1.10.4 Configuring DSCP to Control PFC for TCP Traffic

- Create a QoS policy to tag All TCP/UDP traffic with CoS value 3 and DSCP value 9.

```
PS $ New-NetQosPolicy "DEFAULT" -Default -PriorityValue8021Action 3 -DSCPAction 9
```

DSCP can also be configured per protocol.

```
PS $ New-NetQosPolicy "TCP"  -IPProtocolMatchCondition TCP -PriorityValue8021Action 3 -
DSCPAction 16
PS $ New-NetQosPolicy "UDP"  -IPProtocolMatchCondition UDP -PriorityValue8021Action 3 -
DSCPAction 32
```

3.1.10.5 Configuring DSCP to Control ETS for TCP Traffic

- Create a QoS policy to tag All TCP/UDP traffic with CoS value 0 and DSCP value 8.

```
PS $ New-NetQosPolicy "DEFAULT" -Default -PriorityValue8021Action 0 -DSCPAction 8 -Pol-
icyStore activestore
```

- Configure DSCP with value 16 for TCP/IP connections with a range of ports.

```
PS $ New-NetQosPolicy "TCP1" -DSCPAction 16 -IPDstPortStartMatchCondition 31000 -IPDst-
PortEndMatchCondition 31999 -IPProtocol TCP -PriorityValue8021Action 0 -PolicyStore
activestore
```

- Configure DSCP with value 24 for TCP/IP connections with another range of ports.

```
PS $ New-NetQosPolicy "TCP2" -DSCPAction 24 -IPDstPortStartMatchCondition 21000 -IPDst-
PortEndMatchCondition 31999 -IPProtocol TCP -PriorityValue8021Action 0 -PolicyStore
activestore
```

- Configure two Traffic Classes with bandwidths of 16% and 80%.

```
PS $ New-NetQosTrafficClass -name "TCP1" -priority 3 -bandwidthPercentage 16 -Algorithm
ETS
PS $ New-NetQosTrafficClass -name "TCP2" -priority 5 -bandwidthPercentage 80 -Algorithm
ETS
```

3.1.10.6 Configuring DSCP to Control PFC for RDMA Traffic

- Create a QoS policy to tag the ND traffic for port 10000 with CoS value 3.

```
PS $ New-NetQosPolicy "ND10000" -NetDirectPortMatchCondition 10000 - PriorityVal-
ue8021Action 3
```

Related Commands:

- Get-NetAdapterQos - Gets the QoS properties of the network adapter
- Get-NetQosPolicy - Retrieves network QoS policies
- Get-NetQosFlowControl - Gets QoS status per priority

3.1.10.7 Registry Settings

The following attributes must be set manually and will be added to the miniport registry.

Table 7 - DSCP Registry Keys Settings

Registry Key	Description
TxUntagPriorityTag	If 0x1, do not add 802.1Q tag to transmitted packets which are assigned 802.1p priority, but are not assigned a non-zero VLAN ID (i.e. priority-tagged). Default 0x0, for DSCP based PFC set to 0x1.
RxUntaggedMapToLossless	If 0x1, all untagged traffic is mapped to the lossless receive queue. Default 0x0, for DSCP based PFC set to 0x1.
RroceDscpMarkPriorityFlowControl_<ID>	A value to mark DSCP for RoCE packets assigned to CoS=ID, when priority flow control is enabled. The valid values range is from 0 to 63, Default is ID value, e.g. PriorityToDscpMappingTable_3 is 3. ID values range from 0 to 7.
DscpBasedEtsEnabled	If 0x1 - all Dscp based ETS feature is enabled, if 0x0 - disabled. Default 0x0.
DscpForGlobalFlowControl	Default DSCP value for flow control. Default 0x1a.

For changes to take affect, please restart the network adapter after changing this registry key.

3.1.10.7.1Default Settings

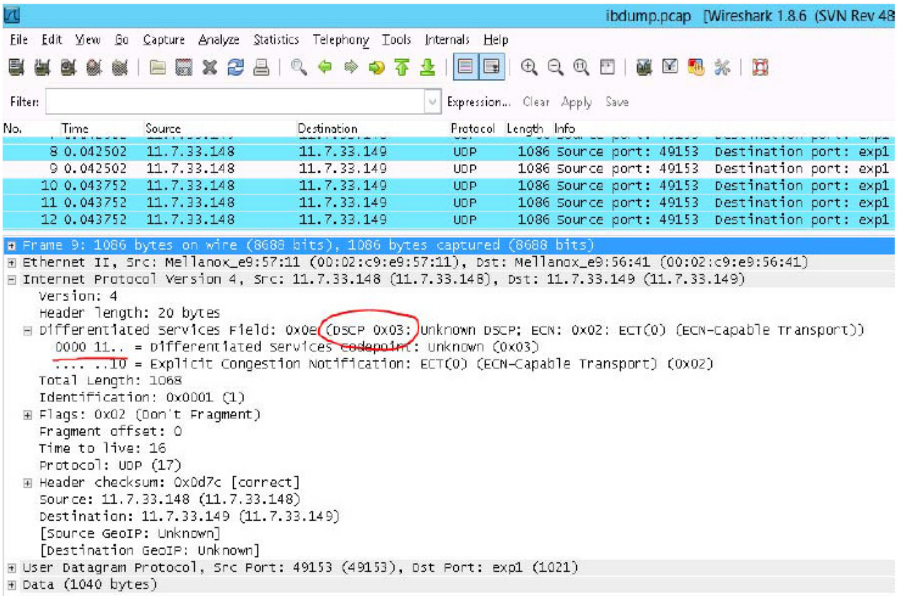
When DSCP configuration registry keys are missing in the miniport registry, the following defaults are assigned:

Table 8 - DSCP Default Registry Keys Settings

Registry Key	Default Value
TxUntagPriorityTag	0
RxUntaggedMapToLossles	0
PriorityToDscpMappingTable_0	0
PriorityToDscpMappingTable_1	1
PriorityToDscpMappingTable_2	2
PriorityToDscpMappingTable_3	3
PriorityToDscpMappingTable_4	4
PriorityToDscpMappingTable_5	5
PriorityToDscpMappingTable_6	6
PriorityToDscpMappingTable_7	7
DscpBasedEtsEnabled	eth:0
DscpForGlobalFlowControl	26

3.1.10.8 DSCP Sanity Testing

To verify that all QoS and DSCP settings were correct, you can capture incoming and outgoing traffic by using the ibdump tool and see the DSCP value in the captured packets as displayed in the figure below.


ibdump.pcap [Wireshark 1.8.6 (SVN Rev 48
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.   Time       Source        Destination   Protocol Length Info
    8 0.042502   11.7.33.148   11.7.33.149   UDP      1086 Source port: 49153  Destination port: expl
    9 0.042502   11.7.33.148   11.7.33.149   UDP      1086 Source port: 49153  Destination port: expl
   10 0.043752   11.7.33.148   11.7.33.149   UDP      1086 Source port: 49153  Destination port: expl
   11 0.043752   11.7.33.148   11.7.33.149   UDP      1086 Source port: 49153  Destination port: expl
   12 0.043752   11.7.33.148   11.7.33.149   UDP      1086 Source port: 49153  Destination port: expl
```

```
⊞ Frame 9: 1086 bytes on wire (8688 bits), 1086 bytes captured (8688 bits)
⊞ Ethernet II, Src: Mellanox_e9:57:11 (00:02:c9:e9:57:11), Dst: Mellanox_e9:56:41 (00:02:c9:e9:56:41)
⊟ Internet Protocol Version 4, Src: 11.7.33.148 (11.7.33.148), Dst: 11.7.33.149 (11.7.33.149)
    Version: 4
    Header length: 20 bytes
  ⊟ Differentiated Services Field: 0x0e (DSCP 0x03: Unknown DSCP; ECN: 0x02: ECT(0) (ECN-Capable Transport))
      0000 11.. = Differentiated Services Codepoint: Unknown (0x03)
      .... ..10 = Explicit Congestion Notification: ECT(0) (ECN-Capable Transport) (0x02)
    Total Length: 1068
    Identification: 0x0001 (1)
  ⊞ Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 16
    Protocol: UDP (17)
  ⊞ Header checksum: 0x0d7c [correct]
    Source: 11.7.33.148 (11.7.33.148)
    Destination: 11.7.33.149 (11.7.33.149)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
⊞ User Datagram Protocol, Src Port: 49153 (49153), Dst Port: expl (1021)
⊞ Data (1040 bytes)
```

3.1.11 Lossless TCP

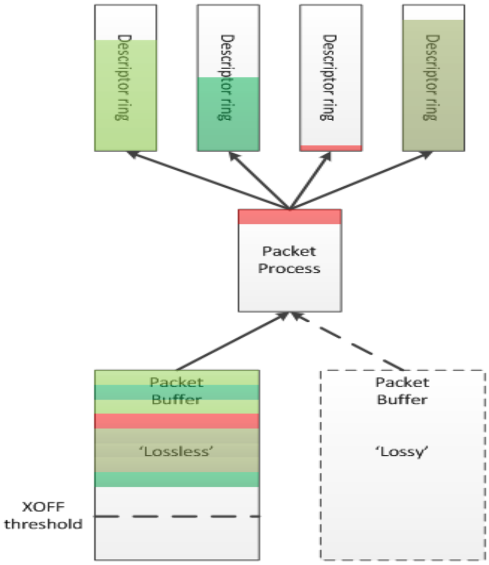
3.1.11.1 System Requirements

Operating Systems: Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows 7 Client, Windows 8.1 Client and Windows Server 2016.

3.1.11.2 Using Lossless TCP

Inbound packets are stored in the data buffers. They are split into 'Lossy' and 'Lossless' according to the priority field in the 802.1Q VLAN tag. In DSCP based PFC, all traffic is directed to the 'Lossless' buffer. Packets are taken out of the packet buffer in the same order they were stored, and moved into processing, where a destination descriptor ring is selected. The packet is then scattered into the appropriate memory buffer, pointed by the first free descriptor.

Figure 6: Lossless TCP



When the 'Lossless' packet buffer crosses the XOFF threshold, the adapter sends 802.3x pause frames according to the port configuration: Global pause, or per-priority 802.1Qbb pause (PFC), where only the priorities configured as 'Lossless' will be noted in the pause frame. Packets arriving while the buffer is full are dropped immediately.

During packet processing, if the selected descriptor ring has no free descriptors, two modes for handling are available:

3.1.11.3 Drop Mode

In this mode, a packet arriving to a descriptor ring with no free descriptors is dropped, after verifying that there are really no free descriptors. This allows isolation of the host driver execution delays from the network, as well as isolation between different SW entities sharing the adapter (e.g. SR-IOV VMs).

3.1.11.4 Poll Mode

In this mode, a packet arriving to a descriptor ring with no free descriptors will patiently wait until a free descriptor is posted. All processing for this packet and the following packets is halted, while free descriptor status is polled. This behavior will propagate the backpressure into the Rx buffer which will accumulate incoming packets. When XOFF threshold is crossed, Flow Control mechanisms mentioned earlier will stop the remote transmitters, thus avoiding packets from being dropped.

Since this mode breaks the aforementioned isolation, the adapter offers a mitigation mechanism that limits the amount of time a packet may wait for a free descriptor, while halting all packet processing. When the allowed time expires the adapter reverts to the 'Drop Mode' behavior.

3.1.11.5 Default behavior

By default the adapter works in 'Drop Mode'. The adapter reverts to this mode upon initialization/ restart.

3.1.11.6 Known Limitations

- The feature is not available for SR-IOV Virtual Functions
- It is recommended that the feature be used only when the port is configured to maintain flow control.
- It is recommended not to exceed typical timeout values of management protocols, usually in the order of several seconds.
- In order for the feature to effectively prevent packet drops, the DPC load duration needs to be lower than the TCP retransmission timeout.
- The feature is only activated if neither of the ports is IB.

3.1.11.7 System Requirements

- Operating Systems: Windows Server 2012 or Windows Server 2012 R2 and Windows Server 2016
- Firmware: 2.31.5050

3.1.11.8 Enabling/Disabling Lossless TCP

This feature is controlled using the registry key `DelayDropTimeout` that enables Lossless TCP capability in hardware and by Set OID `OID_MLX_DROPLESS_MODE` which triggers transition to/ from Lossless (poll) mode.

3.1.11.8.1Enabling Lossless TCP Using The Registry Key `DelayDropTimeout`:

Registry Key location:

```
HKLM\SYSTEM\CurrentControlSet\Control\Class\Class\{4d36e972-e325-11ce-
bfc1-08002be10318}\<nn>\DelayDropTimeout
```

For instructions on how to find interface index in registry <nn>, Please refer to Section 3.6.2, “Finding the Index Value of the Network Interface”, on page 130

Table 9 - Enabling Lossless TCP Using The Registry Key DelayDropTimeout

Key Name	Key Type	Values	Description
Delay-DropTimeout	REG_DWORD	• 0= disabled (default) • 1-65535=enabled 0	Choosing values between 1-65534 enables the feature, but the chosen value limits the amount of time a packet may wait for a free descriptor. The value is in units of 100 microseconds with inaccuracy of up to 2 units. The chosen time ranges between 100 microseconds and ~6.5 seconds. For example, DelayDropTimeout=3000 limits the wait time to 300 miliseconds (+/- 200 microseconds) Choosing the value of 65535 enables the feature but the amount of time a packet may wait for a free descriptor is infinite. **Note**: Changing the value of the DelayDropTimeout registry key requires restart of the network interface

3.1.11.8.2Entering/Exiting Lossless Mode Using Set OID OID_MLX_DROPLESS_MODE:

In order to enter poll mode, registry value of DelayDropTimeout should be non-zero and OID_MLX_DROPLESS_MODE Set OID should be called with Information Buffer containing 1.

- OID_MLX_DROPLESS_MODE value: 0xFFA0C932
- OID Information Buffer Size: 1 byte
- OID Information Buffer Contents: 0 - exit poll mode; 1 - enter poll mode

3.1.11.9 Monitoring Lossless TCP State

In order to allow state transition monitoring, events are written to event log with mlx4_bus as the source. The associated events are listed in Table 9.

Table 10 - Lossless TCP Associated Events

Event ID	Event Description
0x0057 <Device Name>	Dropless mode entered on port <X>. Packets will not be dropped.
0x0058 <Device Name>	Dropless mode exited on port <X>. Drop mode entered; packets may now be dropped.
0x0059 <Device Name>	Delay drop timeout occurred on port <X>. Drop mode entered; packets may now be dropped.

3.1.12 Receive Side Scaling (RSS)

3.1.12.1 System Requirements

Operating Systems: Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows 7 Client, Windows 8.1 Client and Windows Server 2016.

3.1.12.2 Using RSS

Mellanox WinOF Rev 5.35 IPoIB and Ethernet drivers use NDIS 6.30 and above new RSS capabilities. The main changes are:

- Removed the previous limitation of 64 CPU cores
- Individual network adapter RSS configuration usage

RSS capabilities can be set per individual adapters as well as globally.

➢ ***To do so, set the registry keys listed below:***

For instructions on how to find interface index in registry <nn>, please refer to Section 3.6.2, “Finding the Index Value of the Network Interface”, on page 130.

Table 11 - Registry Keys Setting

Sub-key	Description
HKLM\SYSTEM\CurrentControlSet\Control\Class\{4d36e972-e325-11ce-bfc1-08002be10318}\<nn>*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 after you change this registry key.
HKLM\SYSTEM\CurrentControlSet\Control\Class\{4d36e972-e325-11ce-bfc1-08002be10318}\<nn>*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\{4d36e972-e325-11ce-bfc1-08002be10318}\<nn>*NumaNodeID	**NUMA node affinitization**
HKLM\SYSTEM\CurrentControlSet\Control\Class\{4d36e972-e325-11ce-bfc1-08002be10318}\<nn>*RssBaseProcGroup	Sets the RSS base processor group for systems with more than 64 processors.

3.1.13 Ignore Frame Check Sequence (FCS) Errors

Upon receiving packets, these packets go through a checksum validation process for the FCS field. If the validation fails, the received packets are dropped.

When the FCS feature is enabled (disabled by default), the device does not validate the FCS field even if the field is invalid. The registry key for enable/disable is IgnoreFCS.

It is not recommended to ignore FCS, as the field guarantees integrity of received Ethernet frames.

3.2 InfiniBand Network

3.2.1 Port Configuration

For more information on port configuration, please refer to 3.1.1 “Port Configuration,” on page 41.

3.2.2 OpenSM - Subnet Manager

OpenSM v3.3.11 is an InfiniBand Subnet Manager. In order to operate one host machine or more in the InfiniBand cluster., at least one Subnet Manger is required in the fabric.

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

OpenSM can run as a Windows service and can be started manually from the following directory: <installation_directory>\tools. OpenSM as a service will use the first active port, unless it receives a specific GUID.

OpenSM can be registered as a service from either the Command Line Interface (CLI) or the PowerShell.

The following are commands used from the CLI:

➢ ***To register it as a service execute the OpenSM service:***

```
> sc create OpenSM binPath= "c:\Program Files\Mellanox\MLNX-
_VPI\IB\Tools\opensm.exe
-service" start= auto
```

➢ ***To start OpenSM as a service:***

```
> sc start OpenSM
```

➢ ***To run OpenSM manually:***

```
> opensm.exe
```

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

The following are commands used from the PowerShell:

➢ ***To register it as a service execute the OpenSM service:***

```
> New-Service -Name "OpenSM" -BinaryPathName "`"C:\Program Files\Mel-
lanox\MLNX_VPI\IB\Tools\opensm.exe`" --service -L 128" -DisplayName
"OpenSM" -Description "OpenSM for IB subnet" -StartupType Automatic
```

➢ ***To start OpenSM as a service run:***

```
> Start-Service OpenSM1
```

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 two instances of OpenSM in the subnet.

3.2.3 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.

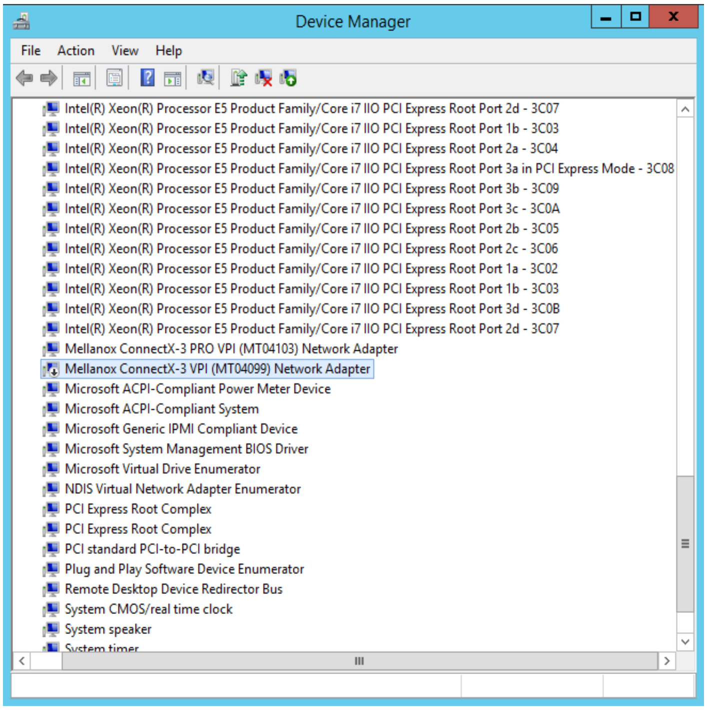
Step 3. Click the Advanced tab and modify the desired properties.

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

3.2.4 Displaying Adapter Related Information

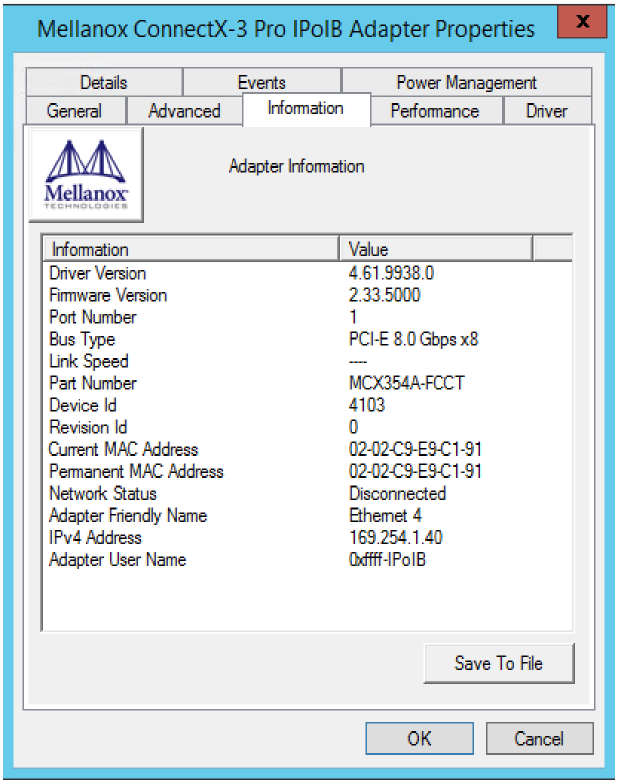
To display a summary of network adapter software, firmware- and hardware-related information such as driver version, firmware version, bus interface, adapter identity, and network port link information, perform the following steps:

Step 1. Display the Device Manager.



Intel(R) Xeon(R) Processor E5 Product Family/Core i7 IIO PCI Express Root Port 2d - 3C07
Intel(R) Xeon(R) Processor E5 Product Family/Core i7 IIO PCI Express Root Port 1b - 3C03
Intel(R) Xeon(R) Processor E5 Product Family/Core i7 IIO PCI Express Root Port 2a - 3C04
Intel(R) Xeon(R) Processor E5 Product Family/Core i7 IIO PCI Express Root Port 3a in PCI Express Mode - 3C08
Intel(R) Xeon(R) Processor E5 Product Family/Core i7 IIO PCI Express Root Port 3b - 3C09
Intel(R) Xeon(R) Processor E5 Product Family/Core i7 IIO PCI Express Root Port 3c - 3C0A
Intel(R) Xeon(R) Processor E5 Product Family/Core i7 IIO PCI Express Root Port 2b - 3C05
Intel(R) Xeon(R) Processor E5 Product Family/Core i7 IIO PCI Express Root Port 2c - 3C06
Intel(R) Xeon(R) Processor E5 Product Family/Core i7 IIO PCI Express Root Port 1a - 3C02
Intel(R) Xeon(R) Processor E5 Product Family/Core i7 IIO PCI Express Root Port 1b - 3C03
Intel(R) Xeon(R) Processor E5 Product Family/Core i7 IIO PCI Express Root Port 3d - 3C0B
Intel(R) Xeon(R) Processor E5 Product Family/Core i7 IIO PCI Express Root Port 2d - 3C07
Mellanox ConnectX-3 PRO VPI (MT04103) Network Adapter
Mellanox ConnectX-3 VPI (MT04099) Network Adapter
Microsoft ACPI-Compliant Power Meter Device
Microsoft ACPI-Compliant System
Microsoft Generic IPMI Compliant Device
Microsoft System Management BIOS Driver
Microsoft Virtual Drive Enumerator
NDIS Virtual Network Adapter Enumerator
PCI Express Root Complex
PCI Express Root Complex
PCI standard PCI-to-PCI bridge
Plug and Play Software Device Enumerator
Remote Desktop Device Redirector Bus
System CMOS/real time clock
System speaker
System timer

Step 2. Select the Information tab from the Properties sheet.



Adapter Information

Information	Value
Driver Version	4.61.9938.0
Firmware Version	2.33.5000
Port Number	1
Bus Type	PCI-E 8.0 Gbps x8
Link Speed	----
Part Number	MCX354A-FCCT
Device Id	4103
Revision Id	0
Current MAC Address	02-02-C9-E9-C1-91
Permanent MAC Address	02-02-C9-E9-C1-91
Network Status	Disconnected
Adapter Friendly Name	Ethernet 4
IPv4 Address	169.254.1.40
Adapter User Name	0xffff-IPoIB

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

3.2.5 Assigning Port IP After Installation

For more information on port configuration, please refer to Section 3.1.2, “Assigning Port IP After Installation”, on page 44 under the Ethernet Network.

3.2.6 Receive Side Scaling (RSS)

For more information on port configuration, please refer to Section 3.1.12, “Receive Side Scaling (RSS)”, on page 82 under the Ethernet Network.

3.2.7 Multiple Interfaces over non-default PKeys Support

3.2.7.1 System Requirements

Operating Systems: Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2 and Windows Server 2016.

3.2.7.2 Using Multiple Interfaces over non-default PKeys

OpenSM enables the configuration of partitions (PKeys) in an InfiniBand fabric. IPoIB supports the creation of multiple interfaces via the `part_man` tool. Each of those interfaces can be configured to use a different partition from the ones that were configured for OpenSM. This can allow partitioning of the IPoIB traffic between the different virtual IPoIB interfaces.

➢ ***To create a new interface on a new PKey on a native Windows machine:***

Step 1. Configure OpenSM to recognize the partition you would like to add.

For further details please refer to the section titled “Partitions” in *Mellanox OFED for Linux User Manual.*

Step 2. Create a new interface using the part_man tool.

For further details please refer to section 4.2 “part_man - Virtual IPoIB Port Creation Utility,” on page 180.

Step 3. Assign Port IPs to the new interfaces.

For further details please refer to 3.1.2 “Assigning Port IP After Installation,” on page 44

Make sure the OpenSM using the partitions configuration, and the new interfaces were configured to run over the same physical port.

➢ ***To create a new interface on a new PKey on a Windows virtual machine over a Linux host:***

On the Linux host:

Step 1. Configure the OpenSM to recognize the partition you would like to add.

For further details please refer to the section titled “Partitions” in *Mellanox OFED for Linux User Manual.*

Step 2. Map the physical PKey table to the virtual PKey table used by the VM.

For further details please refer to the section titled “Partitioning IPoIB Communication using PKeys” in *Mellanox OFED for Linux User Manual.*

On the Windows VM:

Step 1. Create a new interface using the `part_man` tool.

For further details please refer to section 4.2 “part_man - Virtual IPoIB Port Creation Utility,” on page 180.

Step 2. Assign Port IPs to the new interfaces.

For further details please refer to 3.1.2 “Assigning Port IP After Installation,” on page 44

> Make sure the OpenSM using the partitions configuration, the physical-to-virtual PKey table mapping and the new interfaces were all configured over the same physical port.

➢ ***To assign a non-default PKey to the physical IPoIB port on a Windows virtual machine over a Linux host:***

On the Windows VM:

Step 1. Disable the driver on the port or disable the bus driver with all the ports it carries through the device manger.

On the Linux host:

Step 2. Configure the OpenSM to recognize the partition you would like to add.

For further details please refer to the section titled “Partitions” in *Mellanox OFED for Linux User Manual.*

Step 3. Map the physical PKey table to the virtual PKey table used by the VM in the following way:

- Map the physical Pkey index you would like to use for the physical port to index 0 in the virtual Pkey table.
- Map the physical PKey index of the default PKey (index 0) to any index (for example: index1) in the virtual PKey table.

For further details please refer to the section titled “Partitioning IPoIB Communication using PKeys” in *Mellanox OFED for Linux User Manual.*

On the Windows VM:

Step 4. Enable the drivers which were disabled.

> Make sure the OpenSM using the partitions configuration, the physical-to-virtual PKey table mapping were configured over the same physical port.

➢ ***To change a configuration of an existing port:***

Step 1. Disable the driver on the port affected by the change you would like to make (or disable the bus driver with all the ports it carries) through the device manger in Windows OS.

Step 2. If required, configure the OpenSM to recognize the partition you would like to add or change. For further details please refer to the section titled “Partitions” in *Mellanox OFED for Linux User Manual.*

Step 3. If the change is on a VM over a Linux host, map the physical PKey table to the virtual PKey table as required.
For further details please refer to the section titled “Partitioning IPoIB Communication using PKeys” *Mellanox OFED for Linux User Manual.*

Step 4. Enable the drivers you disabled in Windows OS.

3.2.8 Teaming

Windows Server 2012, 2012 R2 and 2016 support teaming as part of the operating system. However, unlike Mellanox WinOF VPI, it does not support teaming for InfiniBand adapters.

> In this release, this feature is at beta level. In particular, IPv6, VMQ, and configuration through PowerShell are not supported.

3.2.8.1 System Requirements

IPoIB teaming is supported in all operating systems supported by WinOF

3.2.8.2 Adapter Teaming

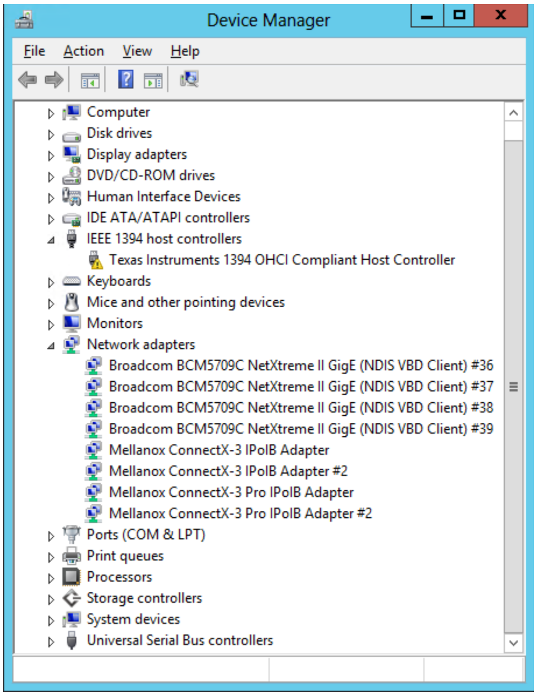
InfiniBand adapter teaming can group a set of interfaces inside a network adapter or a number of physical network adapters into a virtual interface that provides the fault-tolerance function. The fault-tolerance teaming type is the only mode supported in adapter teaming. The non-active interfaces in a team are in a standby mode and will take over the network traffic in the event of a link failure in the active interface. Only one interface is active at any given time.
Note: For InfiniBand, the only teaming mode supported is failover.

3.2.8.3 Creating a Team

Teaming is used to take over packet indications and information requests if the primary network interface fails.

The following steps describe the process of creating a team.

Step 1. Display the Device Manager.



- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Human Interface Devices
- IDE ATA/ATAPI controllers
- IEEE 1394 host controllers
  - Texas Instruments 1394 OHCI Compliant Host Controller
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
  - Broadcom BCM5709C NetXtreme II GigE (NDIS VBD Client) #36
  - Broadcom BCM5709C NetXtreme II GigE (NDIS VBD Client) #37
  - Broadcom BCM5709C NetXtreme II GigE (NDIS VBD Client) #38
  - Broadcom BCM5709C NetXtreme II GigE (NDIS VBD Client) #39
  - Mellanox ConnectX-3 IPoIB Adapter
  - Mellanox ConnectX-3 IPoIB Adapter #2
  - Mellanox ConnectX-3 Pro IPoIB Adapter
  - Mellanox ConnectX-3 Pro IPoIB Adapter #2
- Ports (COM & LPT)
- Print queues
- Processors
- Storage controllers
- System devices
- Universal Serial Bus controllers

Step 2. Right-click one of Mellanox ConnectX IPoIB adapters (under “Network adapters” list) and left click Properties. Select the Teaming tab from the Properties window.

> It is not recommended to open the Properties window of more than one adapter simultaneously.

The Teaming dialog enables creating, modifying or removing a team.

> Only Mellanox Technologies adapters can be part of the team.

➢ ***To create a new team, perform the following***

Step 1. Click Create.

Step 2. Enter a (unique) team name.

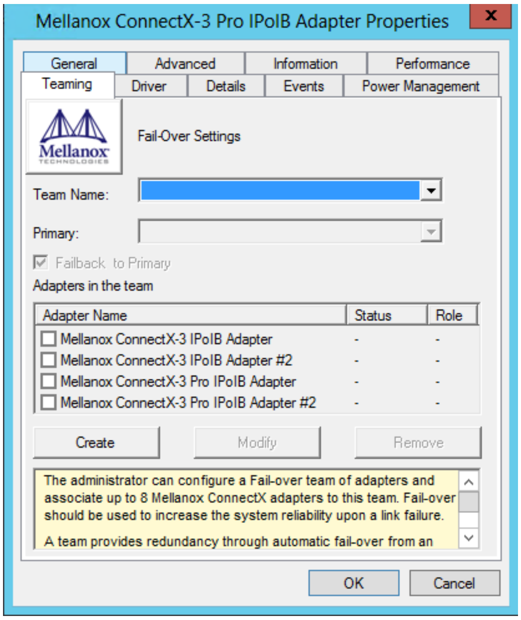
Step 3. Select the adapters to be included in the team.

Step 4. [Optional] Select Primary Adapter.
An InfiniBand team implements an active-passive scenario where only one interface is active at any given time. When the active one is disconnected, one of the other interfaces becomes active. When the primary link comes up, the team interface returns to transfer data using the primary interface. If the primary adapter is not selected, the primary interface is selected randomly.

Step 5. [Optional] Failback to Primary.
This checkbox specifies the team's behavior when the active adapter is not the primary one and the primary adapter becomes available (connected).

- <Failback to Primary> checked - when the primary adapter becomes available, the team will switch to the primary even though the current active adapter can continue functioning as the active one.
- <Failback to Primary> unchecked - when the primary adapter becomes available, the active adapter will remain active even though the primary can function as the active one

.



Mellanox ConnectX-3 Pro IPoIB Adapter Properties

Fail-Over Settings

Team Name:

Primary:

☑ Failback to Primary

Adapters in the team

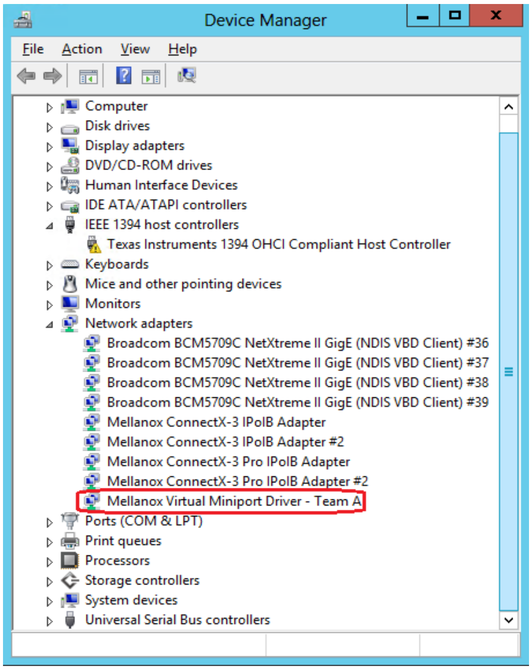
Adapter Name	Status	Role
□ Mellanox ConnectX-3 IPoIB Adapter	-	-
□ Mellanox ConnectX-3 IPoIB Adapter #2	-	-
□ Mellanox ConnectX-3 Pro IPoIB Adapter	-	-
□ Mellanox ConnectX-3 Pro IPoIB Adapter #2	-	-

The administrator can configure a Fail-over team of adapters and associate up to 8 Mellanox ConnectX adapters to this team. Fail-over should be used to increase the system reliability upon a link failure.

A team provides redundancy through automatic fail-over from an

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

```
Mellanox Virtual Miniport Driver - Team <team_name>
```



- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Human Interface Devices
- IDE ATA/ATAPI controllers
- IEEE 1394 host controllers
  - Texas Instruments 1394 OHCI Compliant Host Controller
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
  - Broadcom BCM5709C NetXtreme II GigE (NDIS VBD Client) #36
  - Broadcom BCM5709C NetXtreme II GigE (NDIS VBD Client) #37
  - Broadcom BCM5709C NetXtreme II GigE (NDIS VBD Client) #38
  - Broadcom BCM5709C NetXtreme II GigE (NDIS VBD Client) #39
  - Mellanox ConnectX-3 IPoIB Adapter
  - Mellanox ConnectX-3 IPoIB Adapter #2
  - Mellanox ConnectX-3 Pro IPoIB Adapter
  - Mellanox ConnectX-3 Pro IPoIB Adapter #2
  - Mellanox Virtual Miniport Driver - Team A
- Ports (COM & LPT)
- Print queues
- Processors
- Storage controllers
- System devices
- Universal Serial Bus controllers

➤ ***To modify an existing team, perform the following:***

a. Select the desired team and click Modify

b. Modify the team name and/or the participating adapters

c. Click the Commit button

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

Notes on this step:

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

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

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

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

9.

3.3 Management

3.3.1 PowerShell Configuration

PowerShell is a task automation and configuration management framework from Microsoft, consisting of a command-line shell and associated scripting language built on the .NET Framework. PowerShell provides full access to COM and WMI, enabling administrators to perform administrative tasks on both local and remote Windows systems as well as WS-Management and CIM enabling management of remote Linux systems and network devices.

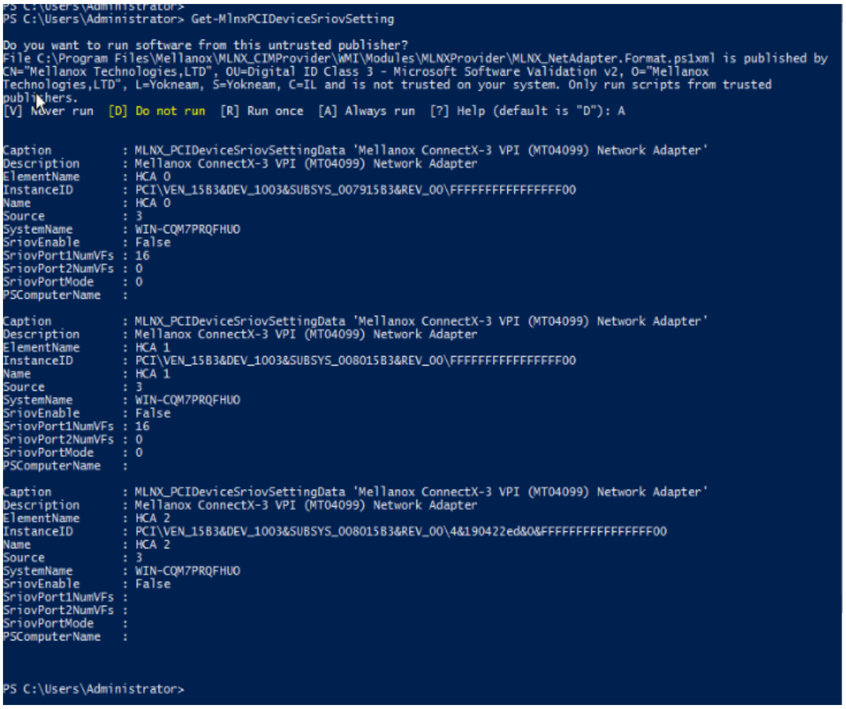
Prior to working with it, PowerShell must be configured as follow:

Step 1. Set the Execution policy to “AllSigned”.

```
  PS $ Set-ExecutionPolicy AllSigned
  Execution Policy Change
  The execution policy helps protect you from scripts that you do not trust. Changing the
  execution policy might expose
  you to the security risks described in the about_Execution_Policies help topic at
  http://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution pol-
icy?
  [Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): y
```

Step 2. Add Mellanox to the trusted publishers by selecting "[A] - Always run" as shown in the example below:

```
  PS $ Get-MlnxPCIDeviceSriovSetting
```



```
PS C:\Users\Administrator>
PS C:\Users\Administrator> Get-MlnxPCIDeviceSriovSetting

Do you want to run software from this untrusted publisher?
File C:\Program Files\Mellanox\MLNX_CIMProvider\WMI\Modules\MLNXProvider\MLNX_NetAdapter.Format.ps1xml is published by
CN="Mellanox Technologies,LTD", OU=Digital ID Class 3 - Microsoft Software Validation v2, O="Mellanox
Technologies,LTD", L=Yokneam, S=Yokneam, C=IL and is not trusted on your system. Only run scripts from trusted
publishers.
[V] Never run  [D] Do not run  [R] Run once  [A] Always run  [?] Help (default is "D"): A


Caption            : MLNX_PCIDeviceSriovSettingData 'Mellanox ConnectX-3 VPI (MT04099) Network Adapter'
Description        : Mellanox ConnectX-3 VPI (MT04099) Network Adapter
ElementName        : HCA 0
InstanceID         : PCI\VEN_15B3&DEV_1003&SUBSYS_007915B3&REV_00\FFFFFFFFFFFFFFFF00
Name               : HCA 0
Source             : 3
SystemName         : WIN-CQM7PRQFHUO
SriovEnable        : False
SriovPort1NumVFs   : 16
SriovPort2NumVFs   : 0
SriovPortMode      : 0
PSComputerName     :

Caption            : MLNX_PCIDeviceSriovSettingData 'Mellanox ConnectX-3 VPI (MT04099) Network Adapter'
Description        : Mellanox ConnectX-3 VPI (MT04099) Network Adapter
ElementName        : HCA 1
InstanceID         : PCI\VEN_15B3&DEV_1003&SUBSYS_008015B3&REV_00\FFFFFFFFFFFFFFFF00
Name               : HCA 1
Source             : 3
SystemName         : WIN-CQM7PRQFHUO
SriovEnable        : False
SriovPort1NumVFs   : 16
SriovPort2NumVFs   : 0
SriovPortMode      : 0
PSComputerName     :

Caption            : MLNX_PCIDeviceSriovSettingData 'Mellanox ConnectX-3 VPI (MT04099) Network Adapter'
Description        : Mellanox ConnectX-3 VPI (MT04099) Network Adapter
ElementName        : HCA 2
InstanceID         : PCI\VEN_15B3&DEV_1003&SUBSYS_008015B3&REV_00\4&190422ed&0&FFFFFFFFFFFFFFFF00
Name               : HCA 2
Source             : 3
SystemName         : WIN-CQM7PRQFHUO
SriovEnable        : False
SriovPort1NumVFs   :
SriovPort2NumVFs   :
SriovPortMode      :
PSComputerName     :


PS C:\Users\Administrator>
```

3.4 Storage Protocols

3.4.1 Deploying Windows Server 2012 and Above with SMB Direct

The Server Message Block (SMB) protocol is a network file sharing protocol implemented in Microsoft Windows. The set of message packets that defines a particular version of the protocol is called a dialect.

The Microsoft SMB protocol is a client-server implementation and consists of a set of data packets, each containing a request sent by the client or a response sent by the server.

SMB protocol is used on top of the TCP/IP protocol or other network protocols. Using the SMB protocol allows applications to access files or other resources on a remote server, to read, create, and update them. In addition, it enables communication with any server program that is set up to receive an SMB client request.

3.4.1.1 System Requirements

The following are hardware and software prerequisites:

- Two or more machines running Windows Server 2012 and above
- One or more Mellanox ConnectX®-3, or ConnectX®-3 Pro adapters for each server
- One or more Mellanox InfiniBand switches
- Two or more QSFP cables required for InfiniBand

3.4.1.2 SMB Configuration Verification

3.4.1.2.1 Verifying Network Adapter Configuration

Use the following PowerShell cmdlets to verify Network Direct is globally enabled and that you have NICs with the RDMA capability.

- Run on both the SMB server and the SMB client.

```
PS $ Get-NetOffloadGlobalSetting | Select NetworkDirect
PS $ Get-NetAdapterRDMA
PS $ Get-NetAdapterHardwareInfo
```

3.4.1.2.2 Verifying SMB Configuration

Use the following PowerShell cmdlets to verify SMB Multichannel is enabled, confirm the adapters are recognized by SMB and that their RDMA capability is properly identified.

- On the SMB client, run the following PowerShell cmdlets:

```
PS $ Get-SmbClientConfiguration | Select EnableMultichannel
PS $ Get-SmbClientNetworkInterface
```

- On the SMB server, run the following PowerShell cmdlets[1]:

```
PS $ Get-SmbServerConfiguration | Select EnableMultichannel
PS $ Get-SmbServerNetworkInterface
PS $ netstat.exe -xan | ? {$_ -match "445"}
```

3.4.1.2.3 Verifying SMB Connection

➢ ***To verify the SMB connection on the SMB client:***

[1] The NETSTAT command confirms if the File Server is listening on the RDMA interfaces.

Step 1. Copy the large file to create a new session with the SMB Server.

Step 2. Open a PowerShell window while the copy is ongoing.

Step 3. Verify the SMB Direct is working properly and that the correct SMB dialect is used.

```
PS $ Get-SmbConnection
PS $ Get-SmbMultichannelConnection
PS $ netstat.exe -xan | ? {$_ -match "445"}
```

> If you have no activity while you run the commands above, you might get an empty list due to session expiration and no current connections.

3.4.1.3 Verifying SMB Events that Confirm RDMA Connection

➢ ***To confirm RDMA connection, verify the SMB events:***

Step 1. Open a PowerShell window on the SMB client. Run the following cmdlets.
NOTE: Any RDMA-related connection errors will be displayed as well.

```
PS $ Get-WinEvent -LogName Microsoft-Windows-SMBClient/Operational | ? Message -match
"RDMA"
```

3.5 Virtualization

3.5.1 Virtual Ethernet Adapter

The Virtual Ethernet Adapter (VEA) provides a mechanism enabling multiple ethernet adapters on the same physical port. Each of these multiple adapters is referred to as a virtual ethernet adapter (VEA).

At present, one can have a total of two VEAs per port. The first VEA, normally the only adapter for the physical port, is referred to as a “physical VEA.” The second VEA, if present, is called a “virtual VEA”. currently only a single “Virtual VEA” is supported. The difference between a virtual and a physical VEA is that RDMA is only available through the physical VEA. In addition, certain settings for the port can only be configured on the physical VEA (see “VEA Feature Limitations” on page 95).

The VEA feature is designed to extend the OS capabilities and increase the usability of the network adapter. At present, once the user binds the RDMA capable network adapter to either teaming interface or Hyper-V, the RDMA capability (ND and NDK) is blocked by the OS. Hence if the user is interested to have RDMA and teaming or Hyper-V at the same time on the same physical Ethernet port, then he can take advantage of this feature: creating two VEAs the, first for RDMA and the second for the other use.

The user can manage VEAs using the “vea_man” tool. For further details on usage, please refer to “vea_man- Virtual Ethernet” on page 182.

Virtual Ethernet Interfaces created by VEA_man are not tuned by the automatic performance tuning script, for optimal performance please follow the performance tuning guide and apply relevant changes to the VEA interface

3.5.1.1 System Requirements

- Operating Systems: Windows Server 2012 and Windows Server 2012 R2
- Firmware version: 2.31.5050 and above

3.5.1.2 VEA Feature Limitations

- RoCE (RDMA) is supported only on the physical VEA
- MTU (*JumboFrame registry key), QoS and, Flow Control are only configured from physical VEA
- No bandwidth allocation between the two interfaces
- Both interfaces share the same link speed
- SR-IOV and VEA are not supported simultaneously. Only one of the features can be used at any given time.
- Starting from WinOF v5.22, VEA is disabled by default since SR-IOV is enable by default. In order to enable it, make sure to set the “PermitSriov” registry key to 0 before upgrading from an earlier version than WinOF v5.22. For further information on the registry key, please refer to Section 3.6.9.1, “SR-IOV Registry Keys”, on page 149.

3.5.2 Hyper-V with VMQ

3.5.2.1 System Requirements

Operating Systems: Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2 and Windows Server 2016.

3.5.2.2 Using Hyper-V with VMQ

Mellanox WinOF 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.

➢ ***To enable Hyper-V with VMQ using UI:***

Step 1. Open Hyper-V Manager.

Step 2. Right-click the desired Virtual Machine (VM), and left-click Settings in the pop-up menu.

Step 3. In the Settings window, under the relevant network adapter, select “Hardware Acceleration”.

Step 4. Check/uncheck the box “Enable virtual machine queue” to enable/disable VMQ on that specific network adapter.

➢ ***To enable Hyper-V with VMQ using PowerShell:***

Step 1. Enable VMQ on a specific VM: Set-VMNetworkAdapter <VM Name> -VmqWeight 100

Step 2. Disable VMQ on a specific VM: Set-VMNetworkAdapter <VM Name> -VmqWeight 0

3.5.3 Network Virtualization using Generic Routing Encapsulation (NVGRE)

> Network Virtualization using Generic Routing Encapsulation (NVGRE) off-load is currently supported in Windows Server 2012 R2/2016 with the latest updates for Microsoft.

3.5.3.1 System Requirements

Operating Systems: Windows Server 2012 R2/2016

Mellanox ConnectX®-3 Pro Adapter with firmware v2.30.8000 or higher

3.5.3.2 Using NVGRE

Network Virtualization using Generic Routing Encapsulation (NVGRE) is a network virtualization technology that attempts to alleviate the scalability problems associated with large cloud computing deployments. It uses Generic Routing Encapsulation (GRE) to tunnel layer 2 packets across an IP fabric, and uses 24 bits of the GRE key as a logical network discriminator (called a tenant network ID).

Configuring the Hyper-V Network Virtualization, requires two types of IP addresses:

- **Provider Addresses (PA)** - Unique IP addresses assigned to each Hyper-V host that are routable across the physical network infrastructure. Each Hyper-V host requires at least one PA to be assigned.

- **Customer Addresses (CA)** - Unique IP addresses assigned to each Virtual Machine that participate on a virtualized network. Using NVGRE, multiple CAs for VMs running on a Hyper-V host can be tunneled using a single PA on that Hyper-V host. CAs must be unique across all VMs on the same virtual network, but they do not need to be unique across virtual networks with different Virtual Subnet ID.

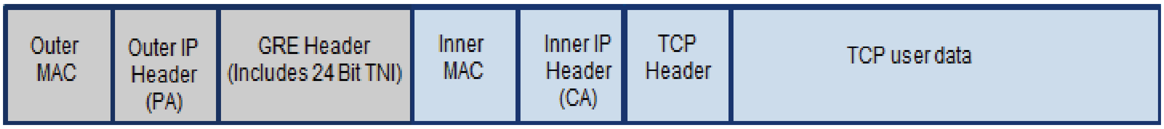
The VM generates a packet with the addresses of the sender and the recipient within the CA space. Then Hyper-V host encapsulates the packet with the addresses of the sender and the recipient in PA space.

PA addresses are determined by using virtualization table. Hyper-V host retrieves the received packet, identifies recipient and forwards the original packet with the CA addresses to the desired VM.

NVGRE can be implemented across an existing physical IP network without requiring changes to physical network switch architecture. Since NVGRE tunnels terminate at each Hyper-V host, the hosts handle all encapsulation and de-encapsulation of the network traffic. Firewalls that block GRE tunnels between sites have to be configured to support forwarding GRE (IP Protocol 47) tunnel traffic.

For further details on configuring NVGRE, please refer to Appendix A,"NVGRE Configuration Scripts Examples," on page 209

Figure 7: NVGRE Packet Structure



Outer MAC	Outer IP Header (PA)	GRE Header (Includes 24 Bit TNI)	Inner MAC	Inner IP Header (CA)	TCP Header	TCP user data

NVGRE is only supported in VMQ mode and not in SR-IOV mode.

3.5.3.3 Enabling/Disabling NVGRE Offloading

To leverage NVGRE to virtualize heavy network IO workloads, the Mellanox ConnectX®-3 Pro network NIC provides hardware support for GRE off-load within the network NICs by default.

➤ ***To enable/disable NVGRE off-loading:***

Step 1. Open the Device Manager.

Step 2. Go to the Network adapters.

Step 3. Right click 'Properties' on Mellanox ConnectX®-3 Pro Ethernet Adapter card.

Step 4. Go to Advanced tab.

Step 5. Choose the `'Encapsulate Task Offload'` option.

Step 6. Set one of the following values:

- Enable - GRE off-loading is Enabled by default
- Disabled - When disabled the Hyper-V host will still be able to transfer NVGRE traffic, but TCP and inner IP checksums will be calculated by software that significant reduces performance.

3.5.3.3.1 Configuring the NVGRE using PowerShell

Hyper-V Network Virtualization policies can be centrally configured using PowerShell 3.0 and PowerShell Remoting.

Step 1. **[Windows Server 2012 Only]** Enable the Windows Network Virtualization binding on the physical NIC of each Hyper-V Host (Host 1 and Host 2)

```
PS $ Enable-NetAdapterBinding <EthInterfaceName>(a)-ComponentID ms_netwnv
```

<EthInterfaceName> - Physical NIC name

Step 2. Create a vSwitch.

```
PS $ New-VMSwitch <vSwitchName> -NetAdapterName <EthInterfaceName>-AllowManagementOS
$true
```

Step 3. Shut down the VMs.

```
PS $ Stop-VM -Name <VM Name> -Force -Confirm
```

Step 4. Configure the Virtual Subnet ID on the Hyper-V Network Switch Ports for each Virtual Machine on each Hyper-V Host (Host 1 and Host 2).

```
PS $ Add-VMNetworkAdapter -VMName <VMName> -SwitchName <vSwitchName> -StaticMacAddress
<StaticMAC Address>
```

Step 5. Configure a Subnet Locator and Route records on all Hyper-V Hosts (same command on all Hyper-V hosts)

```
PS $ New-NetVirtualizationLookupRecord -CustomerAddress <VMInterfaceIPAddress 1/n> -
ProviderAddress <HypervisorInterfaceIPAddress1> -VirtualSubnetID <virtualsubnetID> -
MACAddress <VMmacaddress1>[a] -Rule "TranslationMethodEncap"

PS $ New-NetVirtualizationLookupRecord -CustomerAddress <VMInterfaceIPAddress 2/n> -
ProviderAddress <HypervisorInterfaceIPAddress2> -VirtualSubnetID <virtualsubnetID> -
MACAddress <VMmacaddress2>[a] -Rule "TranslationMethodEncap"
```

[a] This is the VM’s MAC address associated with the vSwitch connected to the Mellanox device.

Step 6. Add customer route on all Hyper-V hosts (same command on all Hyper-V hosts).

```
PS $ New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-
000000005001}" -VirtualSubnetID <virtualsubnetID> -DestinationPrefix <VMInterfaceIPAddress/Mask>
-NextHop "0.0.0.0" -Metric 255
```

Step 7. Configure the Provider Address and Route records on each Hyper-V Host using an appropriate interface name and IP address.

```
PS $ $NIC = Get-NetAdapter <EthInterfaceName>
PS $ New-NetVirtualizationProviderAddress -InterfaceIndex $NIC.InterfaceIndex -ProviderAddress
<HypervisorInterfaceIPAddress> -PrefixLength 24
```

```
PS $ New-NetVirtualizationProviderRoute -InterfaceIndex $NIC.InterfaceIndex -DestinationPrefix
"0.0.0.0/0" -NextHop <HypervisorInterfaceIPAddress>
```

Step 8. Configure the Virtual Subnet ID on the Hyper-V Network Switch Ports for each Virtual Machine on each Hyper-V Host (Host 1 and Host 2).

```
PS $ Get-VMNetworkAdapter -VMName <VMName> | where {$_.MacAddress -eq <VMmacaddress1>}
| Set-VMNetworkAdapter -VirtualSubnetID <virtualsubnetID>
```

Please repeat steps 5 to 8 on each Hyper-V after rebooting the Hypervisor.

3.5.3.4 Verifying the Encapsulation of the Traffic

Once the configuration using PowerShell is completed, verifying that packets are indeed encapsulated as configured is possible through any packet capturing utility. If configured correctly, an encapsulated packet should appear as a packet consisting of the following headers:

Outer ETH Header, Outer IP, GRE Header, Inner ETH Header, Original Ethernet Payload.

3.5.3.5 Removing NVGRE configuration

Step 1. Set VSID back to 0 (on each Hyper-V for each Virtual Machine where VSID was set)

```
PS $ Get-VMNetworkAdapter <VMName>(a) | where {$_.MacAddress -eq <VMMacAddress>(b)} |
Set-VMNetworkAdapter -VirtualSubnetID 0
```

- VMName - the name of Virtual machine
- VMMacAddress - the MAC address of VM's network interface associated with vSwitch that was connected to Mellanox device.

Step 2. Remove all lookup records (same command on all Hyper-V hosts).

```
PS $ Remove-NetVirtualizationLookupRecord
```

Step 3. Remove customer route (same command on all Hyper-V hosts).

```
PS $ Remove-NetVirtualizationCustomerRoute
```

Step 4. Remove Provider address (same command on all Hyper-V hosts).

```
PS $ Remove-NetVirtualizationProviderAddress
```

Step 5. Remove provider routed for a Hyper-V host.

```
PS $ Remove-NetVirtualizationProviderRoute
```

Step 6. For HyperV running Windows Server 2012 only disable network adapter binding to ms_-netwnv service

```
PS $ Disable-NetAdapterBinding <EthInterfaceName>(a) -ComponentID ms_netwnv
<EthInterfaceName> - Physical NIC name
```

3.5.4 Single Root I/O Virtualization (SR-IOV)

SR-IOV is not on a GA level in Windows 2012. It is recommended to use Windows 2012 R2 or a newer version, when SR-IOV is configured.

Single Root I/O Virtualization (SR-IOV) is a technology that allows a physical PCIe device to present itself multiple times through the PCIe bus. This technology enables multiple virtual instances of the device with separate resources. Mellanox adapters are capable of exposing in ConnectX®-3/ConnectX®-3 Pro adapter cards, up to 126 virtual instances called Virtual Functions (VFs). These virtual functions can then be provisioned separately. Each VF can be seen as an addition device connected to the Physical Function. It also shares resources with the Physical Function.

SR-IOV is commonly used in conjunction with an SR-IOV enabled hypervisor to provide virtual machines direct hardware access to network resources hence increasing its performance.

This guide demonstrates the setup and configuration of SR-IOV, using Mellanox ConnectX® VPI adapter cards family. SR-IOV VF is a single port device.

Mellanox device is a dual-port single-PCI function. Virtual Functions' pool belongs to both ports. To define how the pool is divided between the two ports use the Powershell "`SriovPort1NumVFs`" command (see Step 5 in Section 3.5.4.4.2, “Enabling SR-IOV in Mellanox WinOF Package (Ethernet SR-IOV Only)”, on page 112).

3.5.4.1 SR-IOV Ethernet over Hyper-V

3.5.4.1.1 System Requirements

- A server and BIOS with SR-IOV support. BIOS settings might need to be updated to enable virtualization support and SR-IOV support.
- Hypervisor OS: Windows Server 2012 R2 and above
- Virtual Machine (VM) OS:
  - The VM OS can be either Windows Server 2012 and above
- Mellanox ConnectX®-3/ ConnectX®-3 Pro VPI Adapter Card family with SR-IOV capability
- Mellanox WinOF 5.22 or higher
- It is recommended to use the same driver version in the hypervisor and the virtual machine
- Firmware version: 2.30.8000 or higher

3.5.4.1.2 Feature Limitations

- SR-IOV is supported only in Ethernet ports and can be enabled if all ports are set as Ethernet.

- RDMA (i.e RoCE) capability is available in SR-IOV mode only in Windows Server 2016 and above

3.5.4.2 SR-IOV InfiniBand over KVM

3.5.4.2.1 System Requirements

- A server and BIOS with SR-IOV support. BIOS settings might need to be updated to enable virtualization support and SR-IOV support.
- Hypervisor OS: Linux KVM using SR-IOV enabled drivers
- Virtual Machine (VM) OS:
  - The VM OS can be Windows Server 2012, 2012 R2 and 2016

    For further details about assigning a VF to the Windows VM, please refer to steps 1-5 in the section titled “Assigning the SR-IOV Virtual Function to the Red Hat KVM VM Server” in *Mellanox OFED for Linux User Manual.*
- Mellanox ConnectX®-3/ ConnectX®-3 Pro VPI Adapter Card family with SR-IOV capability
- Mellanox WinOF 4.80 or higher
- Firmware version: 2.30.8000 or higher

3.5.4.2.2 Feature Limitations (Compared to Native InfiniBand)

- OpenSM and Infiniband Fabric Diagnostic Utilities listed in Table 39, “Diagnostic Utilities,” on page 185 are not supported in guest OS.
- For a UD QP, only SGID index 0 is supported.
- The allocation of the GIDs (per port) in the VFs are accordingly:
  - 16 GIDs are allocated to the PF
  - 2 GIDs are allocated to every VF
  - The remaining GIDs (if such exist), will be assigned to the VFs, one GID to every VF - starting from the lower VF.
- Currently, Mellanox IB Adapter Diagnostic Counters and Mellanox IB Adapter Traffic Counters are not supported.
- Only Administrator assigned GUIDs are supported, please refer to the MLNX_OFED User Manual for instructions on how to configure Administrator assigned GUIDs.

3.5.4.3 Configuring SR-IOV Host Machines

The following are the necessary steps for configuring host machines:

3.5.4.3.1 Enabling SR-IOV in BIOS

Depending on your system, perform the steps below to set up your BIOS. The figures used in this section are for illustration purposes only.

For further information, please refer to the appropriate BIOS User Manual.

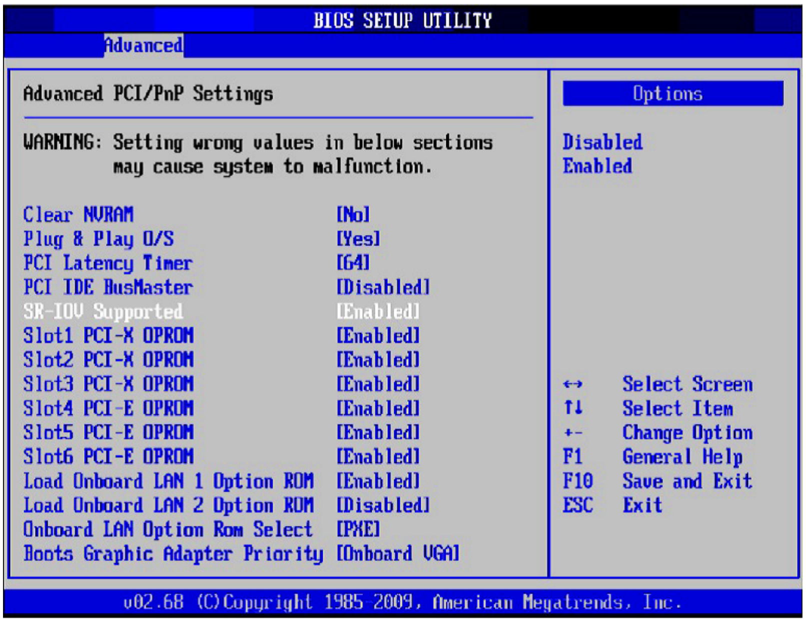
➢ ***To enable SR-IOV in BIOS:***

Step 1. Make sure the machine’s BIOS supports SR-IOV.

Please, consult BIOS vendor website for SR-IOV supported BIOS versions list. Update the BIOS version if necessary.

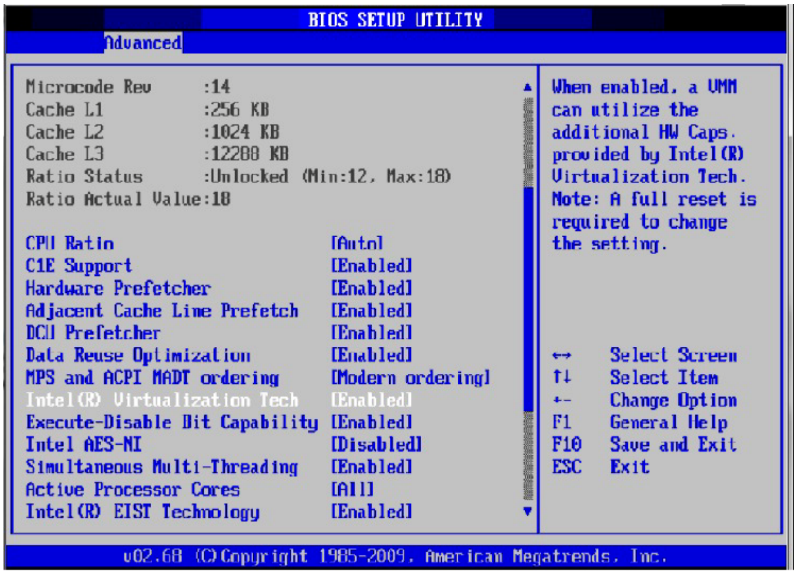
Step 2. Follow BIOS vendor guidelines to enable SR-IOV according to BIOS User Manual.
For example,

a.Enable SR-IOV.



```
                         BIOS SETUP UTILITY
       Advanced

 Advanced PCI/PnP Settings                         Options

 WARNING: Setting wrong values in below sections   Disabled
          may cause system to malfunction.         Enabled

 Clear NVRAM                   [No]
 Plug & Play O/S               [Yes]
 PCI Latency Timer             [64]
 PCI IDE BusMaster             [Disabled]
 SR-IOV Supported              [Enabled]
 Slot1 PCI-X OPROM             [Enabled]
 Slot2 PCI-X OPROM             [Enabled]
 Slot3 PCI-X OPROM             [Enabled]       ↔    Select Screen
 Slot4 PCI-E OPROM             [Enabled]       ↑↓   Select Item
 Slot5 PCI-E OPROM             [Enabled]       +-   Change Option
 Slot6 PCI-E OPROM             [Enabled]       F1   General Help
 Load Onboard LAN 1 Option ROM [Enabled]       F10  Save and Exit
 Load Onboard LAN 2 Option ROM [Disabled]      ESC  Exit
 Onboard LAN Option Rom Select [PXE]
 Boots Graphic Adapter Priority [Onboard VGA]

          v02.68 (C)Copyright 1985-2009, American Megatrends, Inc.
```

b. Enable “Intel Virtualization Technologhy” Support



```
                           BIOS SETUP UTILITY
      Advanced

 Microcode Rev      :14                                  When enabled, a VMM
 Cache L1           :256 KB                              can utilize the
 Cache L2           :1024 KB                             additional HW Caps.
 Cache L3           :12288 KB                            provided by Intel(R)
 Ratio Status       :Unlocked (Min:12, Max:18)           Virtualization Tech.
 Ratio Actual Value:18                                   Note: A full reset is
                                                         required to change
 CPU Ratio                        [Auto]                 the setting.
 C1E Support                      [Enabled]
 Hardware Prefetcher              [Enabled]
 Adjacent Cache Line Prefetch     [Enabled]
 DCU Prefetcher                   [Enabled]
 Data Reuse Optimization          [Enabled]              ↔    Select Screen
 MPS and ACPI MADT ordering       [Modern ordering]      ↑↓   Select Item
 Intel(R) Virtualization Tech     [Enabled]              +-   Change Option
 Execute-Disable Bit Capability   [Enabled]              F1   General Help
 Intel AES-NI                     [Disabled]             F10  Save and Exit
 Simultaneous Multi-Threading     [Enabled]              ESC  Exit
 Active Processor Cores           [All]
 Intel(R) EIST Technology         [Enabled]

        v02.68 (C)Copyright 1985-2009, American Megatrends, Inc.
```

For further details, please refer to the vendor's website.

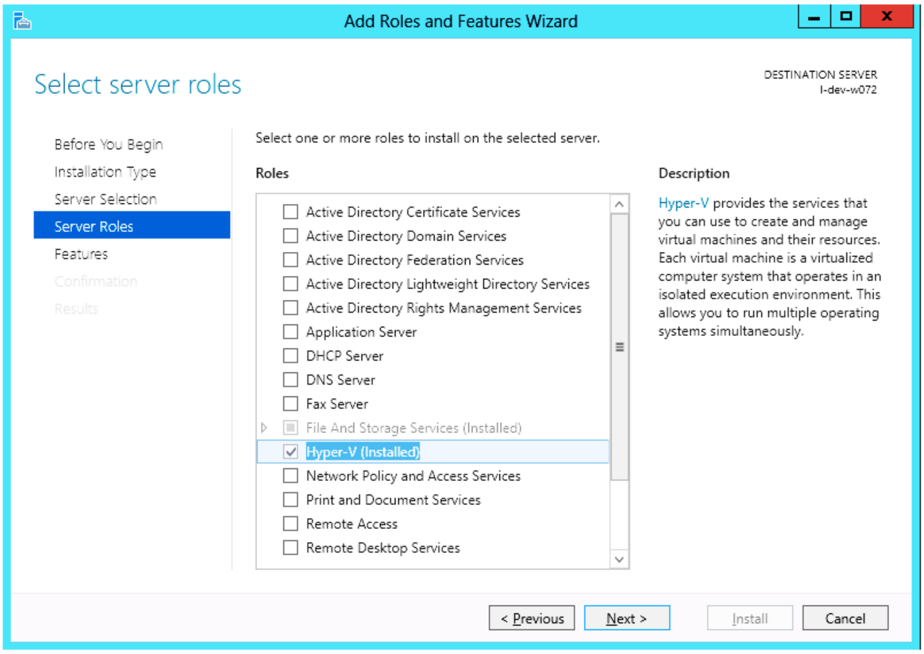
3.5.4.3.2 Installing Hypervisor Operating System (SR-IOV Ethernet Only)

➢ ***To install Hypervisor Operating System:***

Step 1. Install Windows Server 2012 R2 and above

Step 2. Install Hyper-V role:

- Go to: Server Manager -> Manage -> Add Roles and Features and set the following:
  - Installation Type -> Role-based or Feature-based Installation
  - Server Selection -> Select a server fro the server pool
  - Server Roles -> Hyper-V (see figures below)



Add Roles and Features Wizard

Select server roles

DESTINATION SERVER
l-dev-w072

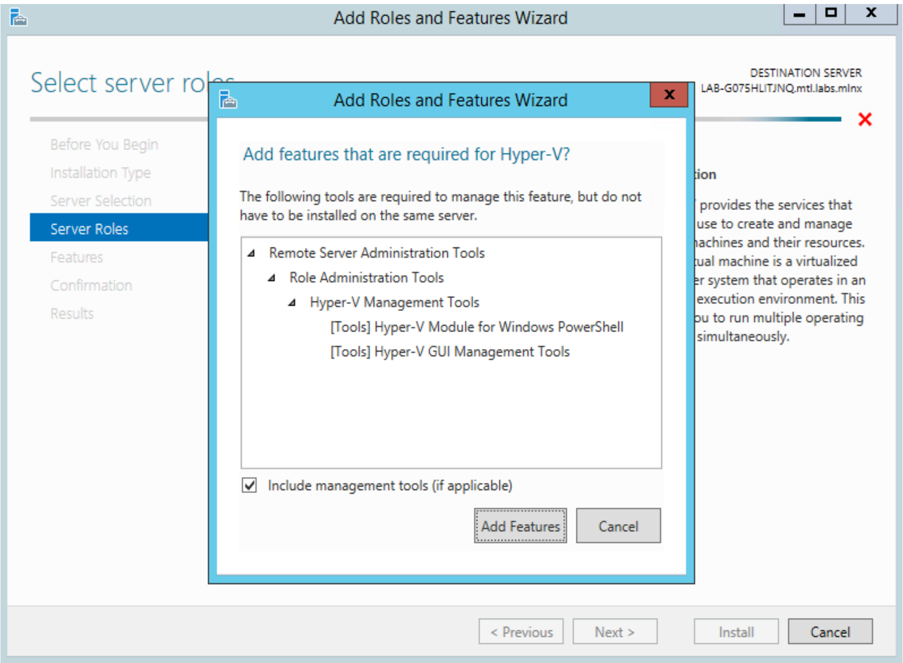
Select one or more roles to install on the selected server.

Roles

- ☐ Active Directory Certificate Services
- ☐ Active Directory Domain Services
- ☐ Active Directory Federation Services
- ☐ Active Directory Lightweight Directory Services
- ☐ Active Directory Rights Management Services
- ☐ Application Server
- ☐ DHCP Server
- ☐ DNS Server
- ☐ Fax Server
- ☐ File And Storage Services (Installed)
- ☑ Hyper-V (Installed)
- ☐ Network Policy and Access Services
- ☐ Print and Document Services
- ☐ Remote Access
- ☐ Remote Desktop Services

Description

Hyper-V provides the services that you can use to create and manage virtual machines and their resources. Each virtual machine is a virtualized computer system that operates in an isolated execution environment. This allows you to run multiple operating systems simultaneously.



Add Roles and Features Wizard

Select server roles

DESTINATION SERVER
LAB-G075HLITJNQ.mtl.labs.mlnx

Before You Begin
Installation Type
Server Selection
Server Roles
Features
Confirmation
Results

ion
provides the services that
use to create and manage
achines and their resources.
ual machine is a virtualized
r system that operates in an
execution environment. This
u to run multiple operating
simultaneously.

Add Roles and Features Wizard

Add features that are required for Hyper-V?

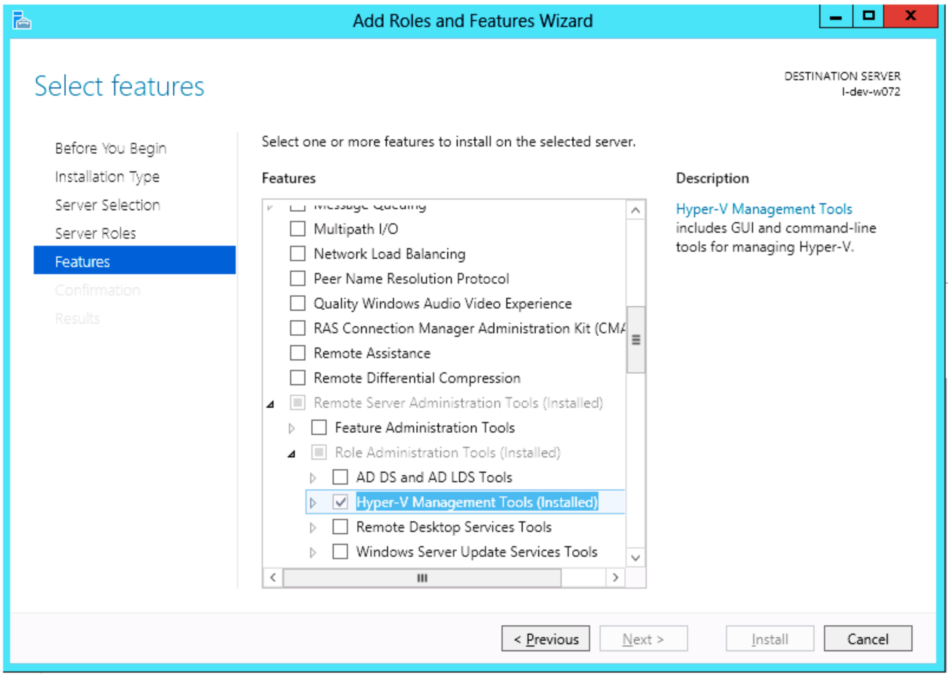
The following tools are required to manage this feature, but do not have to be installed on the same server.

- Remote Server Administration Tools
  - Role Administration Tools
    - Hyper-V Management Tools
      - [Tools] Hyper-V Module for Windows PowerShell
      - [Tools] Hyper-V GUI Management Tools

☑ Include management tools (if applicable)

Step 3. Install Hyper-V Management Tools.

Features - > Remote Server Administration Tools -> Role Administration Tools -> Hyper-V Administration Tool



Add Roles and Features Wizard

Select features

DESTINATION SERVER
l-dev-w072

Before You Begin
Installation Type
Server Selection
Server Roles
Features
Confirmation
Results

Select one or more features to install on the selected server.

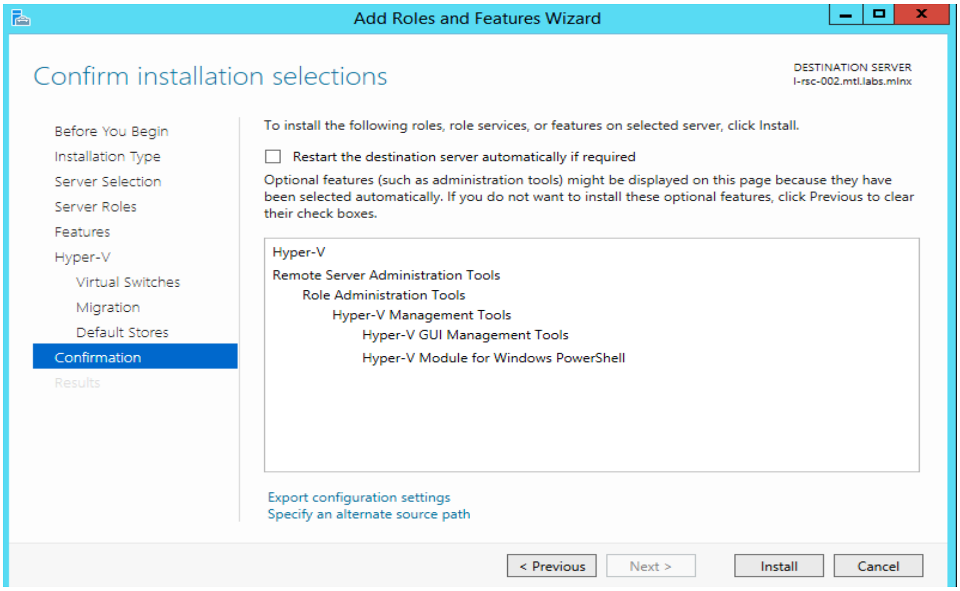
Features

- □ Message Queuing
- □ Multipath I/O
- □ Network Load Balancing
- □ Peer Name Resolution Protocol
- □ Quality Windows Audio Video Experience
- □ RAS Connection Manager Administration Kit (CMA
- □ Remote Assistance
- □ Remote Differential Compression
- ▣ Remote Server Administration Tools (Installed)
  - □ Feature Administration Tools
  - ▣ Role Administration Tools (Installed)
    - □ AD DS and AD LDS Tools
    - ☑ Hyper-V Management Tools (Installed)
    - □ Remote Desktop Services Tools
    - □ Windows Server Update Services Tools

Description

Hyper-V Management Tools includes GUI and command-line tools for managing Hyper-V.

Step 4. Confirm the Installation.



Confirm installation selections

DESTINATION SERVER
l-rsc-002.mtl.labs.mlnx

To install the following roles, role services, or features on selected server, click Install.

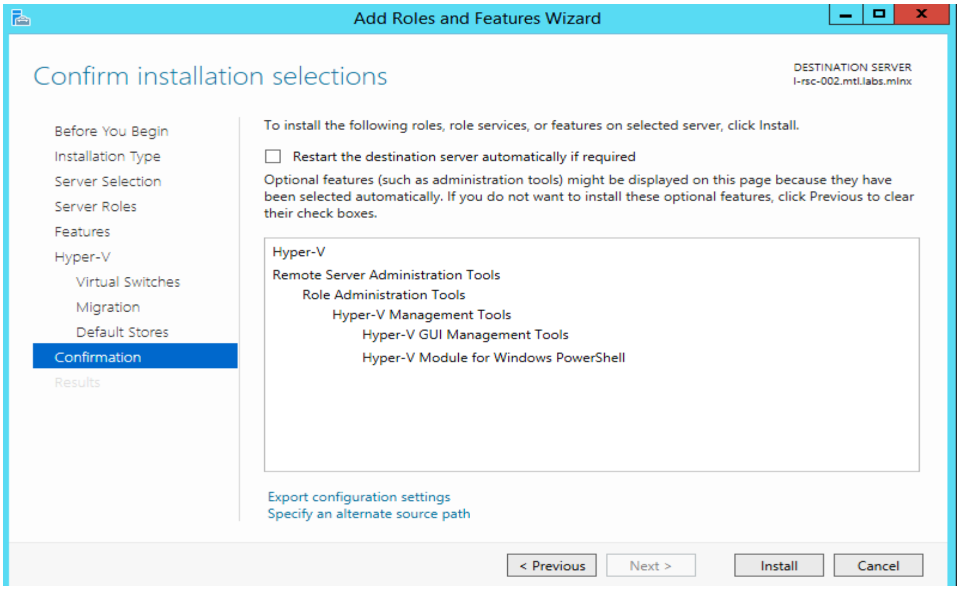
□ Restart the destination server automatically if required

Optional features (such as administration tools) might be displayed on this page because they have been selected automatically. If you do not want to install these optional features, click Previous to clear their check boxes.

Hyper-V
Remote Server Administration Tools
    Role Administration Tools
        Hyper-V Management Tools
            Hyper-V GUI Management Tools
            Hyper-V Module for Windows PowerShell

Export configuration settings
Specify an alternate source path

Step 5. Click Install.



Confirm installation selections

DESTINATION SERVER
l-rsc-002.mtl.labs.mlnx

To install the following roles, role services, or features on selected server, click Install.

□ Restart the destination server automatically if required

Optional features (such as administration tools) might be displayed on this page because they have been selected automatically. If you do not want to install these optional features, click Previous to clear their check boxes.

Hyper-V
Remote Server Administration Tools
    Role Administration Tools
        Hyper-V Management Tools
            Hyper-V GUI Management Tools
            Hyper-V Module for Windows PowerShell

Export configuration settings
Specify an alternate source path

Step 6. Reboot the system.

3.5.4.3.3 Verifying SR-IOV Support within the Host Operating System (SR-IOV Ethernet Only)

➢ ***To verify that the system is properly configured for SR-IOV:***

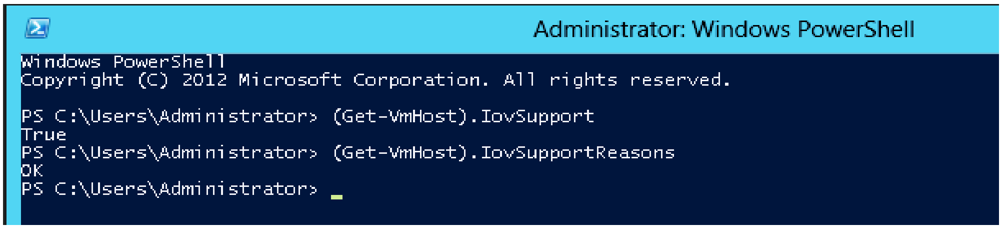
Step 1. Go to: Start-> Windows Powershell.

Step 2. Run the following PowerShell commands.

```
PS $ (Get-VmHost).IovSupport
PS $ (Get-VmHost).IovSupportReasons
```

In case that SR-IOV is supported by the OS, the output in the PowerShell is as in the figure below.

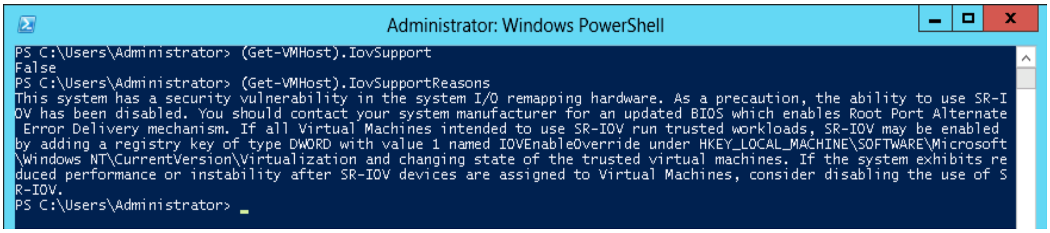
Figure 8: Operating System Supports SR-IOV



```
Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> (Get-VmHost).IovSupport
True
PS C:\Users\Administrator> (Get-VmHost).IovSupportReasons
OK
PS C:\Users\Administrator> _
```

Note: If BIOS was updated according to BIOS vendor instructions and you see the message displayed in the figure below, update the registry configuration as described in the (Get-VmHost). IovSupportReasons message.

Figure 9: SR-IOV Support



```
PS C:\Users\Administrator> (Get-VMHost).IovSupport
False
PS C:\Users\Administrator> (Get-VMHost).IovSupportReasons
This system has a security vulnerability in the system I/O remapping hardware. As a precaution, the ability to use SR-I
OV has been disabled. You should contact your system manufacturer for an updated BIOS which enables Root Port Alternate
 Error Delivery mechanism. If all Virtual Machines intended to use SR-IOV run trusted workloads, SR-IOV may be enabled
by adding a registry key of type DWORD with value 1 named IOVEnableOverride under HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft
\Windows NT\CurrentVersion\Virtualization and changing state of the trusted virtual machines. If the system exhibits re
duced performance or instability after SR-IOV devices are assigned to Virtual Machines, consider disabling the use of S
R-IOV.
PS C:\Users\Administrator> _
```

Step 3. Reboot

Step 4. Verify the system is configured correctly for SR-IOV as described in Steps 1/2.

3.5.4.3.4 Creating a Virtual Machine (SR-IOV Ethernet Only)

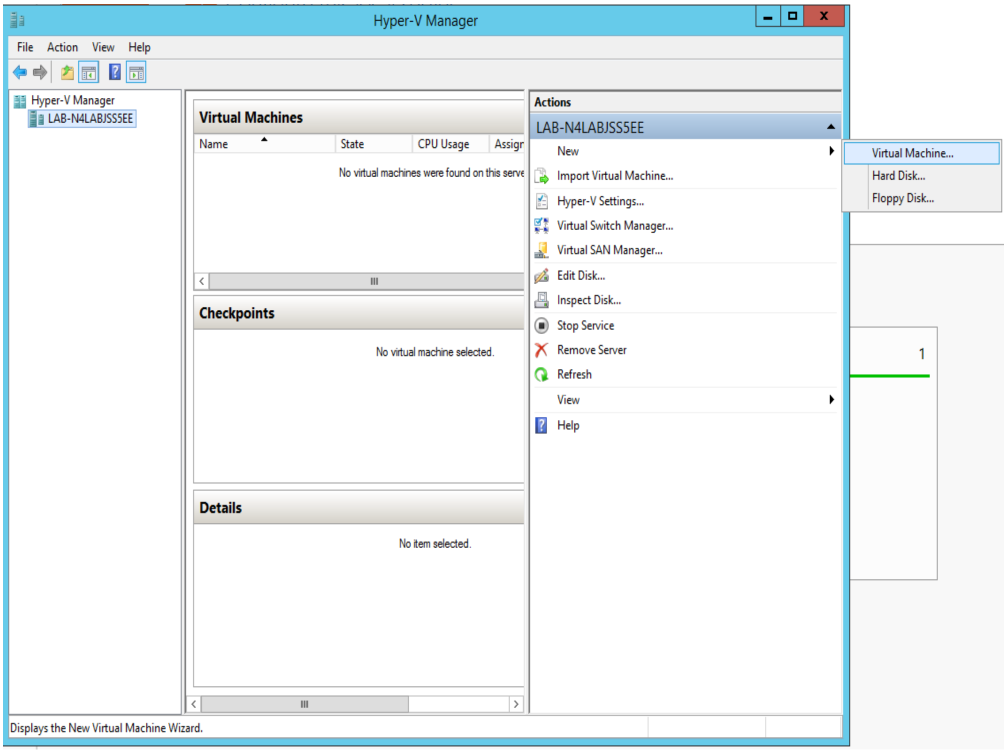
➢ ***To create a virtual machine***

Step 1. Go to: Server Manager -> Tools -> Hyper-V Manager.

Step 2. Go to: New->Virtual Machine and set the following:

- Name: <name>
- Startup memory: 4096 MB
- Connection: Not Connected

Figure 10: Hyper-V Manager



No virtual machines were found on this serve

No virtual machine selected.

No item selected.

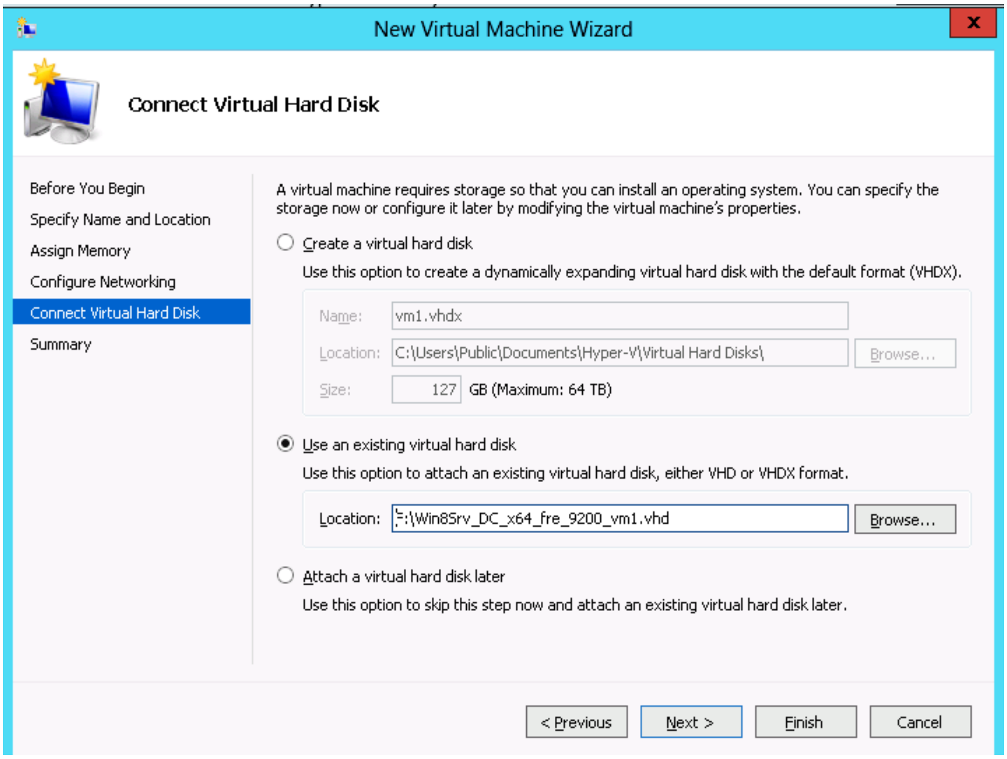
Displays the New Virtual Machine Wizard.

Step 3. Connect the virtual hard disk in the New Virtual Machine Wizard.

Step 4. Go to: Connect Virtual Hard Disk -> Use an existing virtual hard disk.

Step 5. Select the location of the vhd file.

Figure 11: Connect Virtual Hard Disk



A virtual machine requires storage so that you can install an operating system. You can specify the storage now or configure it later by modifying the virtual machine's properties.

○ Create a virtual hard disk

Use this option to create a dynamically expanding virtual hard disk with the default format (VHDX).

Name: vm1.vhdx

Location: C:\Users\Public\Documents\Hyper-V\Virtual Hard Disks\

Size: 127 GB (Maximum: 64 TB)

● Use an existing virtual hard disk

Use this option to attach an existing virtual hard disk, either VHD or VHDX format.

Location: E:\Win8Srv_DC_x64_fre_9200_vm1.vhd

○ Attach a virtual hard disk later

Use this option to skip this step now and attach an existing virtual hard disk later.

3.5.4.4 Configuring Mellanox Network Adapter for SR-IOV

The following are the steps for configuring Mellanox Network Adapter for SR-IOV:

3.5.4.4.1 Enabling SR-IOV in Firmware

For non-Mellanox (OEM) branded cards you may need to download and install the new firmware. For the latest OEM firmware, please go to:
http://www.mellanox.com/page/oem_firmware_download

As of firmware version 2.31.5000, SR-IOV can be enabled and managed by using the mlxconfig too. For older firmware versions, use the flint tool.

➢ ***To enable SR-IOV using mlxconfig:***

mlxconfig is part of MFT tools used to simplify firmware configuration. The tool is available with MFT tools 3.6.0 or higher.

Step 1. Download MFT for Windows.

www.mellanox.com > Products > Software > Firmware Tools

Step 2. Get the device ID (look for the "_pciconf" string in the output).

```
> mst status
```

Example:

```
MST devices:
------------
  mt4103_pci_cr0
  mt4103_pciconf0
```

Step 3. Check the current SR-IOV configuration.

```
> mlxconfig -d mt4103_pciconf0 q
```

Example:

```
Device #1:
----------

Device type:    ConnectX3Pro
PCI device:     mt4103_pciconf0

Configurations:          Current
         SRIOV_EN        N/A
         NUM_OF_VFS      N/A
         WOL_MAGIC_EN_P2 N/A
         LINK_TYPE_P1    N/A
         LINK_TYPE_P2    N/A
```

Step 4. Enable SR-IOV with 16 VFs.

```
> mlxconfig -d mt4103_pciconf0 s SRIOV_EN=1 NUM_OF_VFS=16
```

Warning: Care should be taken in increasing the number of VFs. All servers are guaranteed to support 16 VFs. More VFs can lead to exceeding the BIOS limit of MMIO available address space.

Example:

```
Device #1:
----------

Device type:    ConnectX3Pro
PCI device:     mt4103_pciconf0

Configurations:          Current New
         SRIOV_EN        N/A     1
         NUM_OF_VFS      N/A     16
         WOL_MAGIC_EN_P2 N/A     N/A
         LINK_TYPE_P1    N/A     N/A
         LINK_TYPE_P2    N/A     N/A

 Apply new Configuration? ? (y/n) [n] : y
Applying... Done!
-I- Please reboot machine to load new configurations.
```

Step 5. Reboot the machine (After the reboot, continue to Section 3.5.4.4.2, “Enabling SR-IOV in Mellanox WinOF Package (Ethernet SR-IOV Only)”, on page 112).

➢ ***To enable SR-IOV using flint:***

Step 1. Download MFT for Windows.

www.mellanox.com > Products > Software > Firmware Tools

Step 2. Get the device ID (look for the “_pciconf” string in the output).

```
> mst status
```

Example:

```
MST devices:
------------
  mt4103_pci_cr0
  mt4103_pciconf0
```

Step 3. Verify that HCA is configured for SR-IOV by dumping the device configuration file to user-chosen location <ini device file>.ini.

```
> flint -d <device> dc > <ini device file>.ini
```

Step 4. Verify in the [HCA] section of the .ini that the following fields appear:

```
[HCA]
num_pfs = 1
total_vfs = 16
sriov_en = true
```

> **Warning**: Care should be taken in increasing the number of VFs. All servers are guaranteed to support 16 VFs. More VFs can lead to exceeding the BIOS limit of MMIO available address space.

Step 5. If the fields do not appear, edit the .ini file and add them manually.

Table 12 - Recommended Values

Parameter	Recommended Value
num_pfs	1 **Note**: This field is optional and might not always appear.
total_vfs	<0-126> (The chosen value should be within BIOS limit of MMIO available address space)
sriov_en	true

Step 6. Create a binary image using the modified ini file.

```
> mlxburn -fw <fw name>.mlx -conf <ini device file>.ini -wrimage <file name>.bin
```

Step 7. Burn the firmware.

The file <file name>.bin is a firmware binary file with SR-IOV enabled that has 16 VFs.

```
> flint -dev <PCI device> -image <file name>.bin b
```

Step 8. Reboot the system for changes to take effect.

For more information, please, contact Mellanox Support.

3.5.4.4.2 Enabling SR-IOV in Mellanox WinOF Package (Ethernet SR-IOV Only)

➤ ***To enable SR-IOV in Mellanox WinOF Package***

Step 1. Install Mellanox WinOF package that supports SR-IOV.

Step 2. Configure HCA ports' type to Ethernet.
For further information, please refer to Section 3.1.1, “Port Configuration”, on page 41.

Note: SR-IOV cannot be enabled if one of the ports is InfiniBand.

Step 3. Set the Execution Policy specified in Section 3.3.1, “PowerShell Configuration”, on page 91.

Step 4. Query SR-IOV configuration with Powershell.

```
PS $ Get-MlnxPCIDeviceSriovSetting
```

Example:

```
Caption         : MLNX_PCIDeviceSriovSettingData 'Mellanox ConnectX-3 PRO VPI (MT04103)
Network Adapter'
Description      : Mellanox ConnectX-3 PRO VPI (MT04103) Network Adapter
ElementName      : HCA 0
InstanceID       : PCI\VEN_15B3&DEV_1007&SUBSYS_22F5103C&REV_00\24BE05FFFFB9E2E000
Name             : HCA 0
Source           : 3
SystemName       : LAB-N4LABJSS5EE
SriovPort1NumVFs : 16
SriovPort2NumVFs : 0
SriovPortMode    : 0
PSComputerName   :
```

Step 5. Enable SR-IOV through Powershell on both ports.[1]

```
PS $ Set-MlnxPCIDeviceSriovSetting -Name "HCA 0" -SriovPortMode 2
-SriovPort1NumVFs 8 -SriovPort2NumVFs 8
```

Example:

```
Confirm
Are you sure you want to perform this action?
Performing the operation "SetValue" on target "MLNX_PCIDeviceSriovSettingData: MLNX_P-
CIDeviceSriovSettingData 'Mellanox
 ConnectX-3 PRO VPI (MT04103) Network Adapter' (InstanceID =
"PCI\VEN_15B3&DEV_1007&SUBSYS_22F5103C&R...)".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "Y"):
y
```

Mellanox device is a dual-port single-PCI function. Virtual Functions' pool belongs to both ports. To define how the pool is divided between the two ports use the Powershell `“SriovPort1NumVFs”` command.

[1] **SriovPortMode 2** - Enables SR-IOV on both ports.
SriovPort1NumVFs 8 & **SriovPort2NumVFs 8** - Enable 8 Virtual Functions for each port when working in manual mode. By default, there are assigned 16 virtual functions on the first port.

SR-IOV mode configuration parameters:

Table 13 - SR-IOV Mode Configuration Parameters

Parameter Name	Values	Description
SriovPortMode	• 0 = auto_port1 • 1 = auto_port2 • 2 = manual (default)	Configures the number of VFs to be enabled by the bus driver to each port. **Note (for auto mode)**: In auto_portX mode, port X will have the number of VFs according to the burnt value in the device and the other port will have no SR-IOV and it will support native Ethernet (i.e. no RoCE). Setting this parameter to "Manual" will configure the number of VFs for each port according to the registry key MaxVFPortX. **Note (for manual mode)**: The number of VFs can be configured both on a Mellanox bus driver level and Network Interface level (i.e using Set-NetAdapterSriov Powershell cmdlet). The number of VFs actually available to the Network Interface is the minimum value between Mellanox bus driver configuration and Network Interface configuration. For example, if 8 VFs support was burnt in firmware, SriovPortMode is auto_port1, and Network Interface was allowed 32 VFs using SetNetAdapterSriov Powershell cmdlet, the actual number of VFs available to Network Interface will be 8.
MaxVFPort1 MaxVFPort2	• 16=(default)	MaxVFPort<i> specifies the maximum number of VFs that are allowed per port. This is the number of VFs the bus driver will open when working in manual mode. **Note**: If the total number of VFs requested is larger than the number of VFs burnt in firmware, each port X(1\2) will have the number of VFs according to the following formula: (SriovPortXNumVFs / (SriovPort1NumVFs+SriovPort2NumVFs))*number of VFs burnt in firmware.

Step 6. Verify the new values were set correctly.

```
PS $ Get-MlnxPCIDeviceSriovSetting
```

Example:

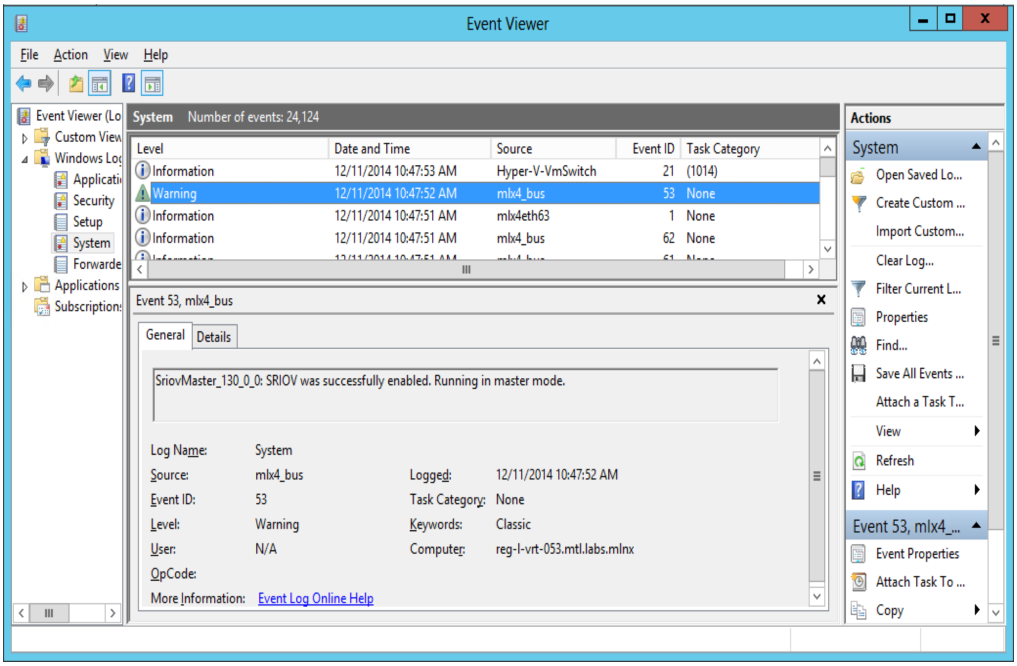
```
Caption          : MLNX_PCIDeviceSriovSettingData 'Mellanox ConnectX-3 PRO VPI (MT04103)
Network Adapter'
Description       : Mellanox ConnectX-3 PRO VPI (MT04103) Network Adapter
ElementName       : HCA 0
InstanceID        : PCI\VEN_15B3&DEV_1007&SUBSYS_22F5103C&REV_00\24BE05FFFFB9E2E000
Name              : HCA 0
Source            : 3
SystemName        : LAB-N4LABJSS5EE
SriovPort1NumVFs  : 8
SriovPort2NumVFs  : 8
SriovPortMode     : 2
PSComputerName    :
```

Step 7. Check in the System Event Log that SR-IOV is enabled.

Step a. Open the View Event Logs/Event Viewer.
Go to: Start -> Control Panel -> System and Security -> Administrative Tools -> View Event Logs/Event Viewer

Step b. Open the System logs.
Event Viewer (Local) -> Windows Logs -> System

Figure 12: System Event Log



System Number of events: 24,124

Level	Date and Time	Source	Event ID	Task Category
Information	12/11/2014 10:47:53 AM	Hyper-V-VmSwitch	21	(1014)
Warning	12/11/2014 10:47:52 AM	mlx4_bus	53	None
Information	12/11/2014 10:47:51 AM	mlx4eth63	1	None
Information	12/11/2014 10:47:51 AM	mlx4_bus	62	None

Event 53, mlx4_bus

SriovMaster_130_0_0: SRIOV was successfully enabled. Running in master mode.

Log Name: System
Source: mlx4_bus
Logged: 12/11/2014 10:47:52 AM
Event ID: 53
Task Category: None
Level: Warning
Keywords: Classic
User: N/A
Computer: reg-l-vrt-053.mtl.labs.mlnx
OpCode:
More Information: Event Log Online Help

3.5.4.5 Configuring Operating Systems

3.5.4.5.1 Configuring Virtual Machine Networking (InfiniBand SR-IOV Only)

For further details on enabling/configuring SR-IOV on KVM, please refer to the section titled “Single Root IO Virtualization (SR-IOV)” in *Mellanox OFED for Linux User Manual.*

3.5.4.5.2 Configuring Virtual Machine Networking (Ethernet SR-IOV Only)

➢ ***To configure Virtual Machine networking:***

Step 1. Create an SR-IOV-enabled Virtual Switch over Mellanox Ethernet Adapter.

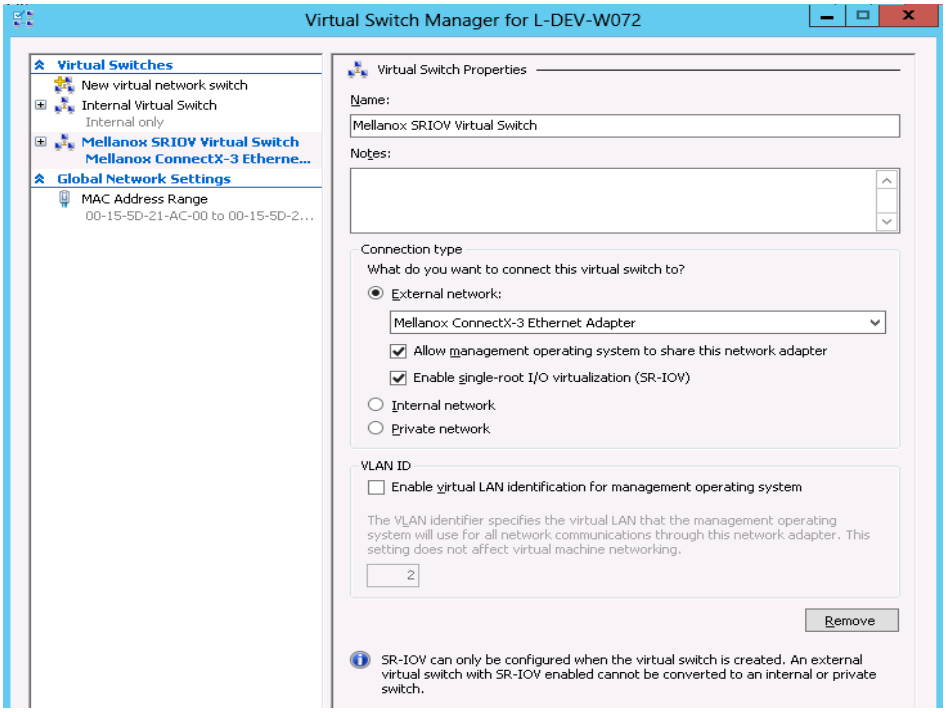
Go to: Start -> Server Manager -> Tools -> Hyper-V Manager

In the Hyper-V Manager: Actions -> Virtual SwitchManager -> External-> Create Virtual Switch

Step 2. Set the following:

- Name:
- External network:
- Enable single-root I/O virtualization (SR-IOV)

Figure 13: Virtual Switch with SR-IOV



Virtual Switch Manager for L-DEV-W072

Virtual Switches
New virtual network switch
Internal Virtual Switch
Internal only
Mellanox SRIOV Virtual Switch
Mellanox ConnectX-3 Etherne...
Global Network Settings
MAC Address Range
00-15-5D-21-AC-00 to 00-15-5D-2...

Virtual Switch Properties

Name:
Mellanox SRIOV Virtual Switch

Notes:

Connection type
What do you want to connect this virtual switch to?
● External network:
Mellanox ConnectX-3 Ethernet Adapter
☑ Allow management operating system to share this network adapter
☑ Enable single-root I/O virtualization (SR-IOV)
○ Internal network
○ Private network

VLAN ID
□ Enable virtual LAN identification for management operating system

The VLAN identifier specifies the virtual LAN that the management operating system will use for all network communications through this network adapter. This setting does not affect virtual machine networking.

2

Remove

SR-IOV can only be configured when the virtual switch is created. An external virtual switch with SR-IOV enabled cannot be converted to an internal or private switch.

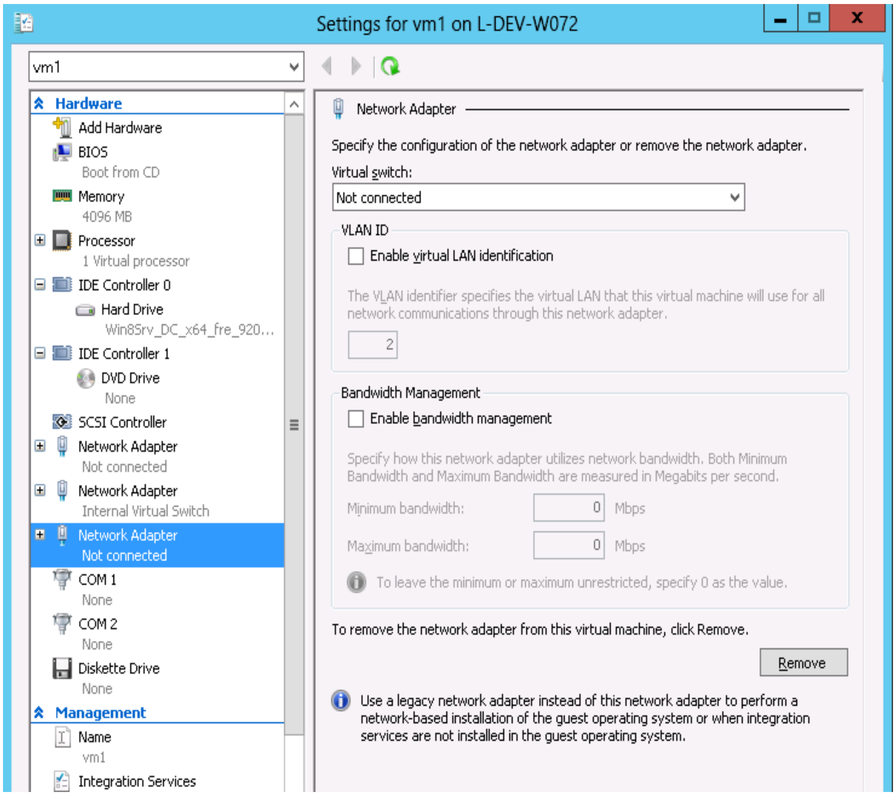
Step 3. Click **Apply**.

Step 4. Click **OK**.

Step 5. Add a VMNIC connected to a Mellanox vSwitch in the VM hardware settings:

- Under Actions, go to Settings -> Add New Hardware-> Network Adapter-> OK.
- In “Virtual Switch” dropdown box, choose Mellanox SR-IOV Virtual Switch.

Figure 14: Adding a VMNIC to a Mellanox V-switch



Specify the configuration of the network adapter or remove the network adapter.

Virtual switch: Not connected

VLAN ID

☐ Enable virtual LAN identification

The VLAN identifier specifies the virtual LAN that this virtual machine will use for all network communications through this network adapter.

2

Bandwidth Management

☐ Enable bandwidth management

Specify how this network adapter utilizes network bandwidth. Both Minimum Bandwidth and Maximum Bandwidth are measured in Megabits per second.

Minimum bandwidth: 0 Mbps

Maximum bandwidth: 0 Mbps

To leave the minimum or maximum unrestricted, specify 0 as the value.

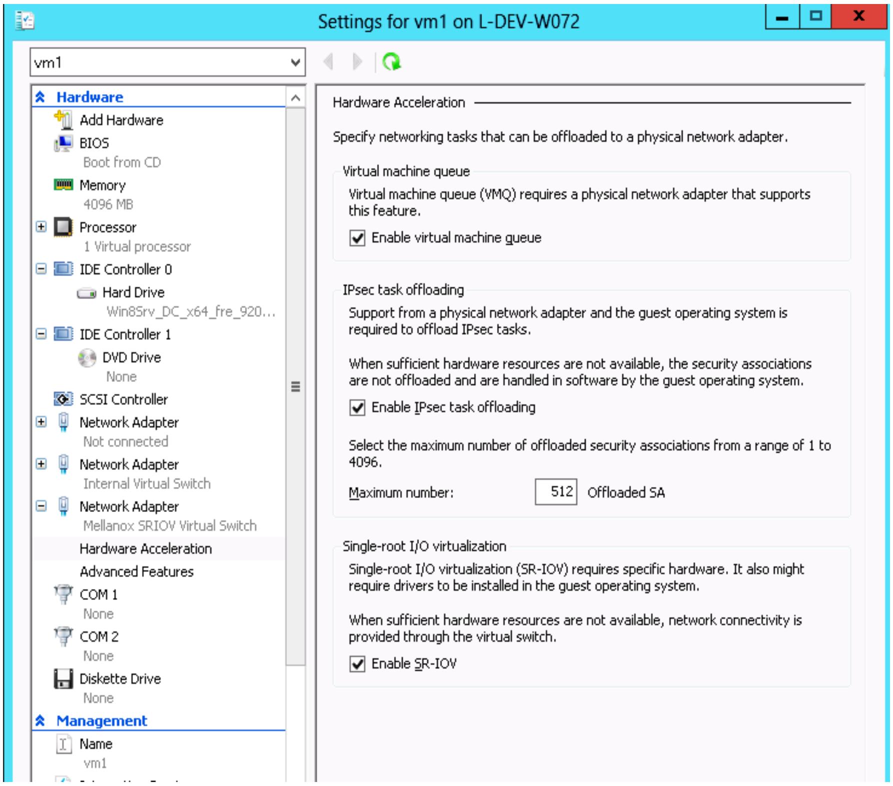
To remove the network adapter from this virtual machine, click Remove.

Use a legacy network adapter instead of this network adapter to perform a network-based installation of the guest operating system or when integration services are not installed in the guest operating system.

Step 6. Enable the SR-IOV for Mellanox VMNIC:

1. Open VM settings Wizard.
2. Open the Network Adapter and choose Hardware Acceleration.
3. Tick the “Enable SR-IOV” option.
4. Click OK.

Figure 15: Enable SR-IOV on VMNIC



Hardware Acceleration

Specify networking tasks that can be offloaded to a physical network adapter.

Virtual machine queue

Virtual machine queue (VMQ) requires a physical network adapter that supports this feature.

☑ Enable virtual machine queue

IPsec task offloading

Support from a physical network adapter and the guest operating system is required to offload IPsec tasks.

When sufficient hardware resources are not available, the security associations are not offloaded and are handled in software by the guest operating system.

☑ Enable IPsec task offloading

Select the maximum number of offloaded security associations from a range of 1 to 4096.

Maximum number: 512 Offloaded SA

Single-root I/O virtualization

Single-root I/O virtualization (SR-IOV) requires specific hardware. It also might require drivers to be installed in the guest operating system.

When sufficient hardware resources are not available, network connectivity is provided through the virtual switch.

☑ Enable SR-IOV

Step 7. Start and connect to the Virtual Machine:

Select the newly created Virtual Machine and go to: Actions panel-> Connect.
In the virtual machine window go to: Actions-> Start

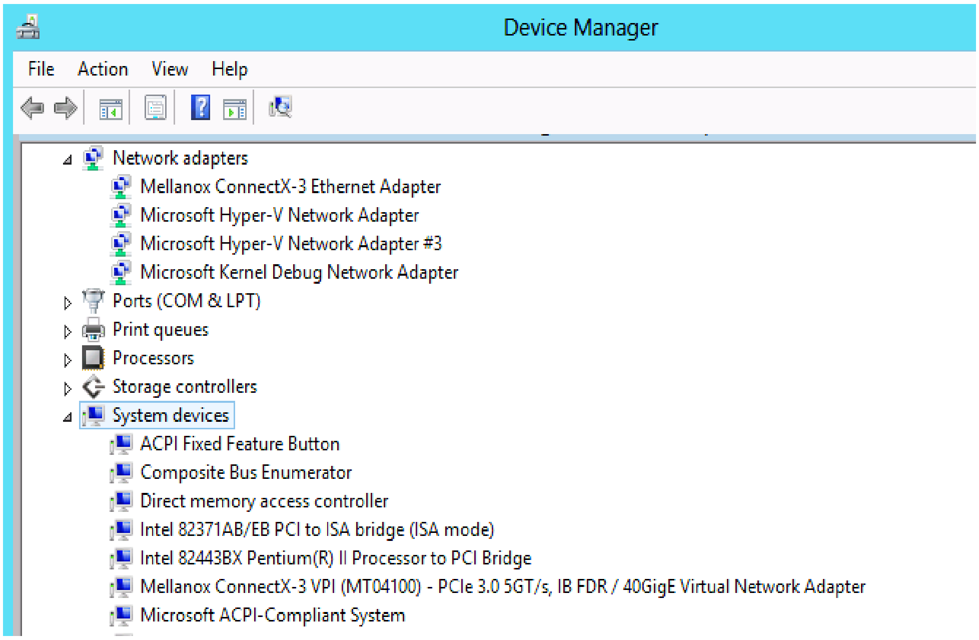
Step 8. Copy the WinOF driver package to the VM using Mellanox VMNIC IP address.

Step 9. Install WinOF driver package on the VM.

Step 10. Reboot the VM at the end of installation.

Step 11. Verify that Mellanox Virtual Function appears in the device manager.

Figure 16: Virtual Function in the VM



- Network adapters
  - Mellanox ConnectX-3 Ethernet Adapter
  - Microsoft Hyper-V Network Adapter
  - Microsoft Hyper-V Network Adapter #3
  - Microsoft Kernel Debug Network Adapter
- Ports (COM & LPT)
- Print queues
- Processors
- Storage controllers
- System devices
  - ACPI Fixed Feature Button
  - Composite Bus Enumerator
  - Direct memory access controller
  - Intel 82371AB/EB PCI to ISA bridge (ISA mode)
  - Intel 82443BX Pentium(R) II Processor to PCI Bridge
  - Mellanox ConnectX-3 VPI (MT04100) - PCIe 3.0 5GT/s, IB FDR / 40GigE Virtual Network Adapter
  - Microsoft ACPI-Compliant System

To achieve best performance on SR-IOV VF, please run the following powershell commands on the host:

For 10Gbe:

PS $ Set-VMNetworkAdapter -Name "Network Adapter" -VMName vm1 -IovQueuePairsRequested 4

For 40Gbe and 56Gbe:

PS $ Set-VMNetworkAdapter -Name "Network Adapter" -VMName vm1 -IovQueuePairsRequested 8

3.5.4.6 Ethertype Spoof Protection

This feature enables the hypervisor to control the allowed Ethertypes that the VF can transmit.

The hypervisor has an Ethertype table for VFs which includes a set of allowed Ethertypes values for transmission.

If a VF tries to transmit packets with undesired Ethertype value, the packets will be transmitted as corrupted but will still be counted as good packets.

The hypervisor has a default Ethertype table for VFs that contains the following values:

Table 14 - Default Ethertype Values

Values	Description
0x0800	Internet Protocol version 4 (IPv4)
0x0806	Address Resolution Protocol (ARP)
0x86DD	Internet Protocol Version 6 (IPv6)

3.5.4.6.1 Ethertype Spoof Protection Registry Keys

The feature could be configured via registry keys as follows:

Registry Keys location for configuration:

```
HKLM\SYSTEM\CurrentControlSet\Control\Class\{4d36e97d-e325-11ce-bfc1-08002be10318}\<nn>\Parameters
```

For more information on how to find device index nn, please refer to Section 3.6.1, “Finding the Index Value of the HCA”, on page 130.

The group below contains the registry keys through which you can configure the feature:

Table 15 - Ethertype Spoof Protection Registry Keys

Key Name	Key Type	Values	Description
VFAllowedTxEtherTypeListEnable	REG_DWORD	• 0 = Disabled • 1 = Enabled	Enables/disables the feature
VFAllowedTxEtherType0	REG_DWORD	Ethertype value	The first Ethertype to allow VF to transmit
VFAllowedTxEtherType1	REG_DWORD	Ethertype value	The second Ethertype to allow VF to transmit
VFAllowedTxEtherType2	REG_DWORD	Ethertype value	The third Ethertype to allow VF to transmit
VFAllowedTxEtherType3	REG_DWORD	Ethertype value	The fourth Ethertype to allow VF to transmit
VFAllowedTxEtherType4	REG_DWORD	Ethertype value	The fifth Ethertype to allow VF to transmit
VFAllowedTxEtherType5	REG_DWORD	Ethertype value	The sixth Ethertype to allow VF to transmit
VFAllowedTxEtherType6	REG_DWORD	Ethertype value	The seventh Ethertype to allow VF to transmit
VFAllowedTxEtherType7	REG_DWORD	Ethertype value	The eighth Ethertype to allow VF to transmit

When the feature is disabled, there is no restriction on the traffic that the VF can transmit.

Configuring at least one Ethertype in registry will override the default table of the Ethertypes mentioned above.

➢ ***Limitations:***

When one of the following Ethertypes is enabled/disabled, the other is automatically enabled/disabled:

- 0x8906 - Fibre Channel over Ethernet (FCoE)
- 0x8914 - FCoE Initialization Protocol

LLC Packets

Transmission of small packets that do not have Ethertype (also known as LLC packets) can be allowed by adding a registry key (one of VFAllowedTxEtherType0..7) with value of 0x0.

3.5.5 Virtual Machine Multiple Queue (VMMQ)

Virtual Machine Multiple Queues (VMMQ), formerly known as Hardware vRSS, is a NIC offload technology that provides scalability for processing network traffic of a VPort in the host (root partition) of a virtualized node. In essence, VMMQ extends the native RSS feature to the VPorts that are associated with the physical function (PF) of a NIC including the default VPort.

VMMQ is available for the VPorts exposed in the host (root partition) regardless of whether the NIC is operating in SR-IOV or VMQ mode. VMMQ is a feature available in Windows Server 2016.

3.5.5.1 System Requirements

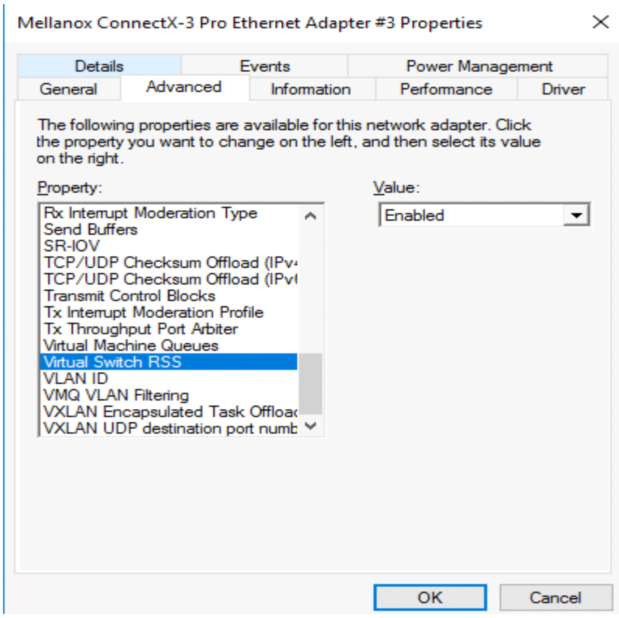
- Operating System(s): Windows Server 2016
- Mellanox ConnectX-3/ConnectX-3 Pro VPI adapter card family
- Available only for Ethernet (no IPOIB)

3.5.5.2 Enabling/Disabling VMMQ

3.5.5.2.1 On the Driver Level

➢ ***To enable/disable VMMQ:***

Step 1. Go to: Display Manager-> Network adapters->Mellanox ConnectX-3 Ethernet Adapter->Properties-> advanced tab->Virtual Switch Rss



The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property:

- Rx Interrupt Moderation Type
- Send Buffers
- SR-IOV
- TCP/UDP Checksum Offload (IPv4
- TCP/UDP Checksum Offload (IPv6
- Transmit Control Blocks
- Tx Interrupt Moderation Profile
- Tx Throughput Port Arbiter
- Virtual Machine Queues
- Virtual Switch RSS
- VLAN ID
- VMQ VLAN Filtering
- VXLAN Encapsulated Task Offloac
- VXLAN UDP destination port numb

Value: Enabled

Step 2. Select Enabled or Disabled

➢ ***To enable/disable VMMQ using a Registry Key:***

Set the `RssOnHostVPorts` registry key in the following path to either `1` (enabled) or `0` (disabled)

```
HKLM\SYSTEM\CurrentControlSet\Control\Class\{4d36e972-e325-11ce-bfc1-
08002be10318}\<nn>\* RssOnHostVPorts
```

3.5.5.2.2 On a VPort

➢ ***To enable VMMQ on a VPort:***

```
PS $ Set-VMNetworkAdapter -Name "Virtual Adapter Name" -VmmqEnabled $true
```

➢ ***To disable VMMQ on a VPort:***

```
PS $ Set-VMNetworkAdapter -Name "Virtual Adapter Name" -VmmqEnabled $false
```

Since the VMMQ is an offload feature for vRss, vRss must be enabled prior to enabling VMMQ.

3.5.5.3 Controlling the Number of Queues Allocated for a vPort

The requested number of queues for a virtual network adapter (vPort) can be set by invoking this PS cmdlet:

```
PS $ Set-VMNetworkAdapter -Name "VM Name" -name "Virtual Adapter Name" -VmmqQueuePairs <num-
ber>
```

The number provided to this cmdlet is the requested number of queues per vPort. However, the OS might decide to not fulfill the request due to some resources and other factors considerations.

3.5.6 Network Direct Kernel Provider Interface

As of v5.25, WinOF supports NDIS Network Direct Kernel Provider Interface version 2. The Network Direct Kernel Provider Interface (NDKPI) is an extension to NDIS that allows IHVs to provide kernel-mode Remote Direct Memory Access (RDMA) support in a network adapter.

3.5.6.1 System Requirement

- Operating System: Windows Server 2012 R2 (Without NDK from/to a VM) and Windows Server 2016
- Firmware Version: 2.36.5150

3.5.6.2 Configuring NDK

3.5.6.2.1 General Configurations

Step 1. Make sure the port is configured as Ethernet.

Step 2. Enable PFC according to your QoS settings (see Section 3.1.4.5.2, “Using Priority Flow Control (PFC)”, on page 53.

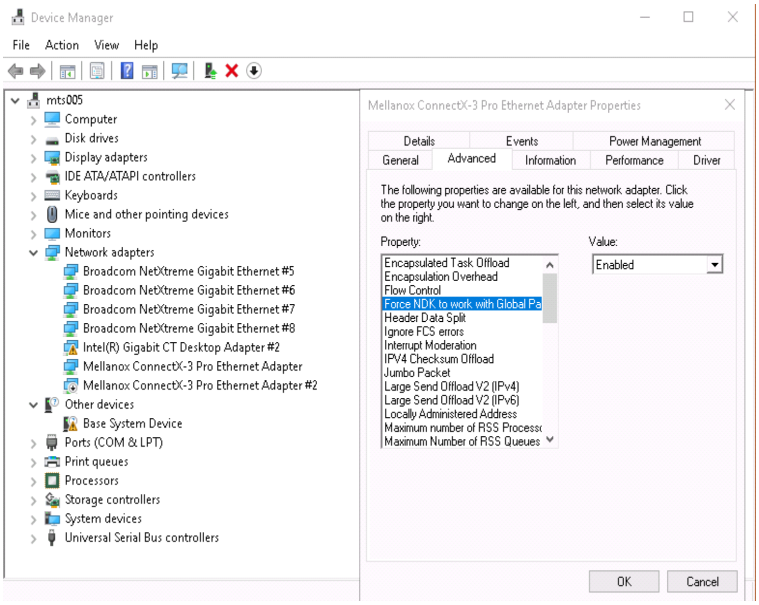
Step 3. Make sure "Force NDK to work with Global Pause" is set to Enabled.

Step a. Open Device Manager.

Step b. Go to Network adapters.

Step c. Right click the right Mellanox ConnectX-3 Pro Ethernet Adapter and choose "Properties".

Step d. Click "Advanced" tab and verify the "Force NDK to work with Global Pause" is set to "Enabled"..


Device Manager
File Action View Help
mts005
Computer
Disk drives
Display adapters
IDE ATA/ATAPI controllers
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Broadcom NetXtreme Gigabit Ethernet #5
Broadcom NetXtreme Gigabit Ethernet #6
Broadcom NetXtreme Gigabit Ethernet #7
Broadcom NetXtreme Gigabit Ethernet #8
Intel(R) Gigabit CT Desktop Adapter #2
Mellanox ConnectX-3 Pro Ethernet Adapter
Mellanox ConnectX-3 Pro Ethernet Adapter #2
Other devices
Base System Device
Ports (COM & LPT)
Print queues
Processors
Storage controllers
System devices
Universal Serial Bus controllers
Mellanox ConnectX-3 Pro Ethernet Adapter Properties
Details
Events
Power Management
General
Advanced
Information
Performance
Driver
Encapsulated Task Offload
Encapsulation Overhead
Flow Control
Force NDK to work with Global Pa
Header Data Split
Ignore FCS errors
Interrupt Moderation
IPV4 Checksum Offload
Jumbo Packet
Large Send Offload V2 (IPv4)
Large Send Offload V2 (IPv6)
Locally Administered Address
Maximum number of RSS Process
Maximum Number of RSS Queues
OK
Cancel

The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property: Force NDK to work with Global Pa

Value: Enabled

Step 4. Make sure the RoCE mode is configured the same on both ends, run vstat from the "Command Prompt". ROCE v2 is the default mode.

3.5.6.2.2 Configuring NDK for Virtual NICs

Step 1. Create a VMSwitch.

```
  PS $ New-VMSwitch -Name <vSwitchName> -NetAdapterName <EthInterfaceName> -AllowManage-
mentOS $False
```

Step 2. Create the virtual network adapters.

```
  PS $ Add-VMNetworkAdapter -SwitchName <vSwitchName> -Name <EthInterfaceName> -Manage-
mentOS
```

Step 3. Enable the "Network Direct(RDMA)" on the new virtual network adapters.

```
  PS $ Enable-NetAdapterRdma <EthInterfaceName>
```

3.5.6.2.3 Configuring the VM

Step 1. Make sure your machine supports SR-IOV.

Step 2. Add the following registry key to the host machine.

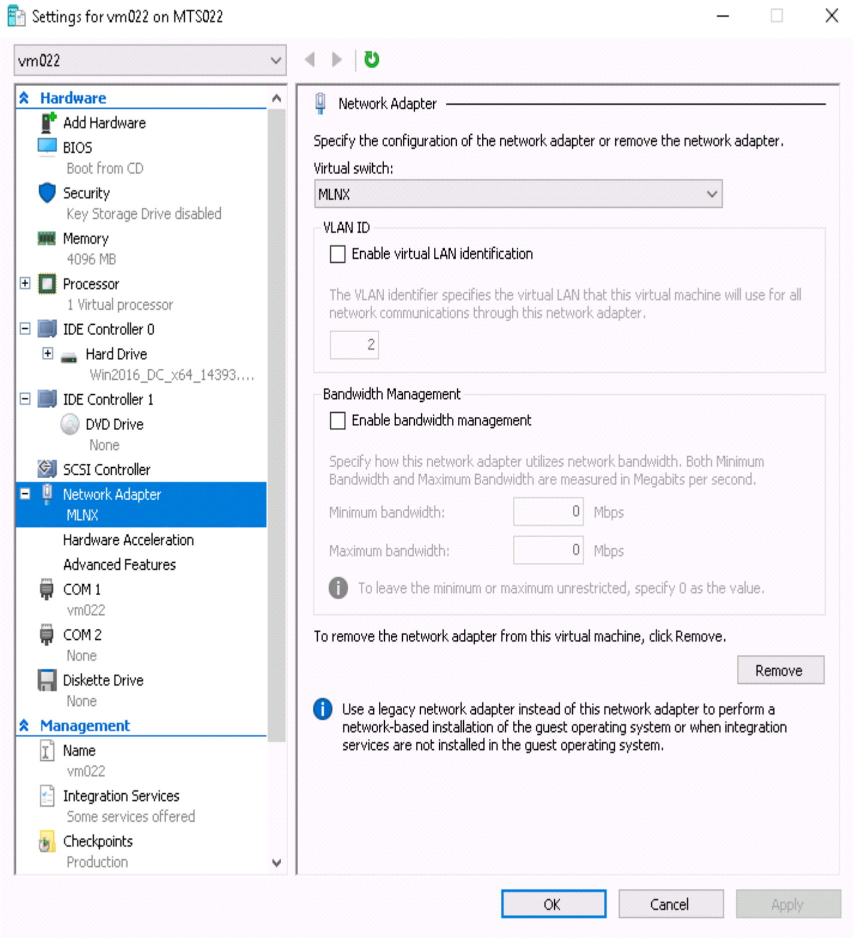
Add a key named "`EnableGuestRdma`" as DWORD in the path:
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\mlx4_bus\Parameters`,
and set it to 1.
You need to disable and re-enable the Mellanox ConnectX-3 device (under System devices) for the change to take effect.

Step 3. Create a VM (make sure the VM is running the same OS as host)

Step 4. Create an SR-IOV enabled VMSwitch.

```
 PS $ New-VMSwitch -Name <vSwitchName> -NetAdapterName <EthInterfaceName> -EnableIov
$True -AllowManagementOS $True
```

Step 5. Add a Network Adapter to the VM in the Hyper-V Manager, and choose the VMSwitch just created.



Settings for vm022 on MTS022

vm022

Hardware
- Add Hardware
- BIOS
  Boot from CD
- Security
  Key Storage Drive disabled
- Memory
  4096 MB
- Processor
  1 Virtual processor
- IDE Controller 0
  - Hard Drive
    Win2016_DC_x64_14393....
- IDE Controller 1
  - DVD Drive
    None
- SCSI Controller
- Network Adapter
  MLNX
  - Hardware Acceleration
  - Advanced Features
- COM 1
  vm022
- COM 2
  None
- Diskette Drive
  None

Management
- Name
  vm022
- Integration Services
  Some services offered
- Checkpoints
  Production

Network Adapter

Specify the configuration of the network adapter or remove the network adapter.

Virtual switch: MLNX

VLAN ID

□ Enable virtual LAN identification

The VLAN identifier specifies the virtual LAN that this virtual machine will use for all network communications through this network adapter.

2

Bandwidth Management

□ Enable bandwidth management

Specify how this network adapter utilizes network bandwidth. Both Minimum Bandwidth and Maximum Bandwidth are measured in Megabits per second.

Minimum bandwidth: 0 Mbps

Maximum bandwidth: 0 Mbps

To leave the minimum or maximum unrestricted, specify 0 as the value.

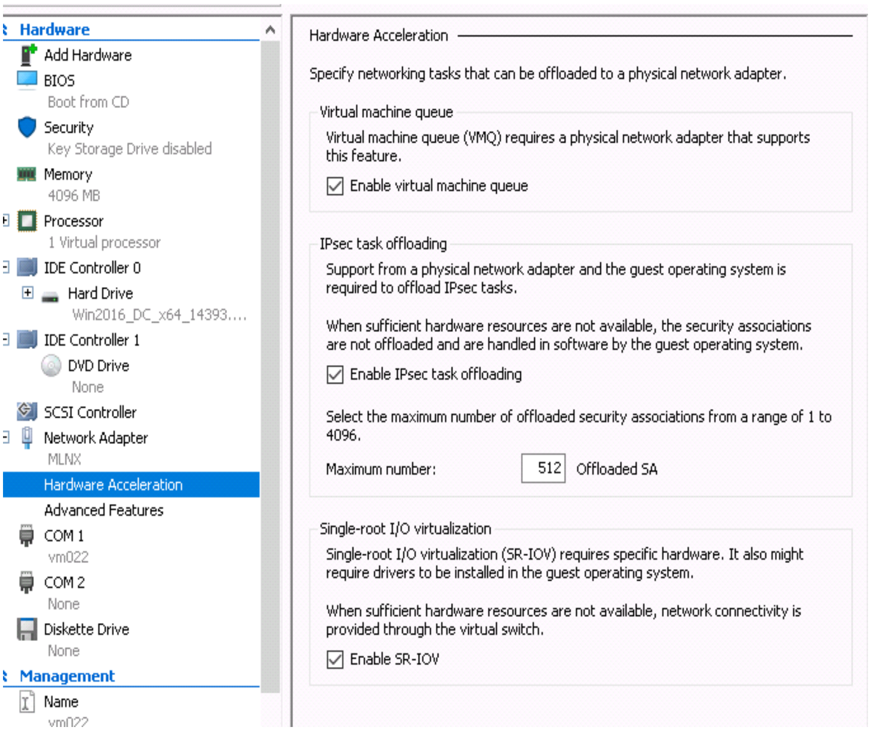
To remove the network adapter from this virtual machine, click Remove.

Remove

Use a legacy network adapter instead of this network adapter to perform a network-based installation of the guest operating system or when integration services are not installed in the guest operating system.

OK Cancel Apply

Step 6. Check the "Enable SR-IOV" option on the "Hardware Acceleration" under the Network Adapter.



Hardware Acceleration

Specify networking tasks that can be offloaded to a physical network adapter.

Virtual machine queue

Virtual machine queue (VMQ) requires a physical network adapter that supports this feature.

☑ Enable virtual machine queue

IPsec task offloading

Support from a physical network adapter and the guest operating system is required to offload IPsec tasks.

When sufficient hardware resources are not available, the security associations are not offloaded and are handled in software by the guest operating system.

☑ Enable IPsec task offloading

Select the maximum number of offloaded security associations from a range of 1 to 4096.

Maximum number: 512 Offloaded SA

Single-root I/O virtualization

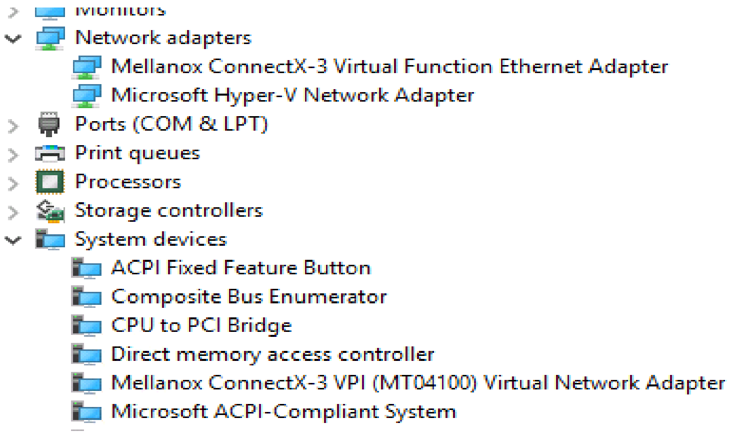
Single-root I/O virtualization (SR-IOV) requires specific hardware. It also might require drivers to be installed in the guest operating system.

When sufficient hardware resources are not available, network connectivity is provided through the virtual switch.

☑ Enable SR-IOV

If you turn ON the VM at this time in the VM Device Manager, you should see the Mellanox ConnectX-3 VPI Virtual Network Adapter under the System devices, and a Mellanox ConnectX-3 Virtual Function Ethernet Adapter under Network adapters.

There may be a yellow sign with the Mellanox ConnectX-3 VPI device initially, after installing the driver in the VM, it will go away.



Monitors
Network adapters
Mellanox ConnectX-3 Virtual Function Ethernet Adapter
Microsoft Hyper-V Network Adapter
Ports (COM & LPT)
Print queues
Processors
Storage controllers
System devices
ACPI Fixed Feature Button
Composite Bus Enumerator
CPU to PCI Bridge
Direct memory access controller
Mellanox ConnectX-3 VPI (MT04100) Virtual Network Adapter
Microsoft ACPI-Compliant System

Step 7. Install the Mellanox Driver in the VM.
Use the same package you installed on the host

Step 8. Enable RDMA on the corresponding network adapter in the VM (Run the command in the VM)

```
PS $ Enable-NetAdapterRdma <EthInterfaceName>
```

3.5.6.3 Utility to Run and Monitor NDK

3.5.6.3.1 Running NDK

Since SMB is NDK's client, it should be used to generate traffic. To generate traffic, do a big copy from one machine to the other.

For instance, use "xcopy" to recursively copy the entire c:\Windows directory or from a "Command Prompt" window, run:

```
xcopy /s c:\Windows \\<remote machine ip>\<remote machine directory for receiving>
```

For example:

```
xcopy /s c:\Windows \\11.0.0.5\c$\tmp
```

3.5.6.3.2 Validating NDK

During the run time of NDK test (xcopy), with "RDMA Activity" in the perfmon.

Use ibdump to see the protocol information of the traffic packet.

3.5.7 PacketDirect Provider Interface

As of v5.25, WinOF supports NDIS PacketDirect Provider Interface. PacketDirect extends NDIS with an accelerated I/O model, which can increase the number of packets processed per second by an order of magnitude and significantly decrease jitter when compared to the traditional NDIS I/O path.

PacketDirect is supported only on Ethernet ports.

3.5.7.1 System Requirements

- Hypervisor OS: Windows Server 2016
- Virtual Machine (VM) OS: Windows Server 2012 and above
- Mellanox ConnectX-3 Pro VPI Adapter Card family
- Mellanox WinOF 5.25 or higher
- Firmware version: 2.36.5150 or higher

3.5.7.2 Using PacketDirect for VM

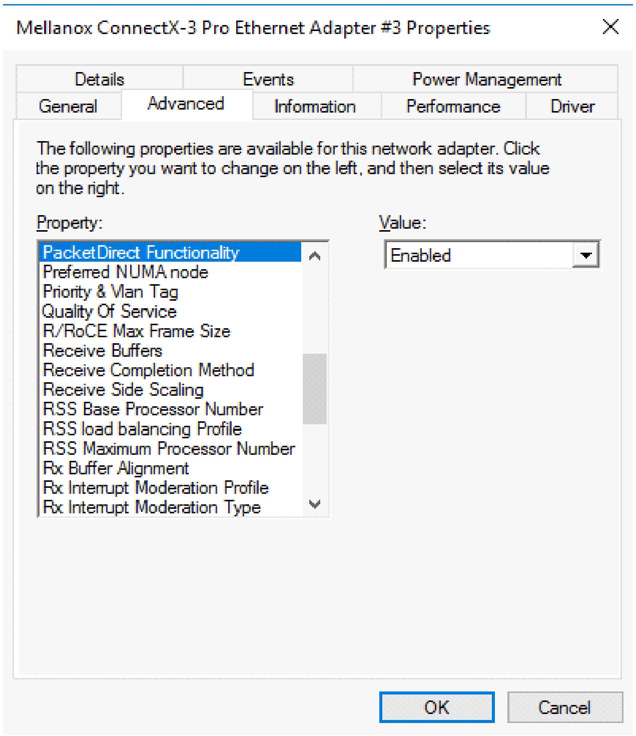
➢ ***To allow a VM to send/receive traffic in PacketDirect mode:***

Step 1. Enable PacketDirect:

- On the Ethernet adapter.

```
PS $ Enable-NetAdapterPacketDirect -Name <EthInterfaceName>
```

- In the Device Manager.



Mellanox ConnectX-3 Pro Ethernet Adapter #3 Properties

The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property:

- PacketDirect Functionality
- Preferred NUMA node
- Priority & Vlan Tag
- Quality Of Service
- R/RoCE Max Frame Size
- Receive Buffers
- Receive Completion Method
- Receive Side Scaling
- RSS Base Processor Number
- RSS load balancing Profile
- RSS Maximum Processor Number
- Rx Buffer Alignment
- Rx Interrupt Moderation Profile
- Rx Interrupt Moderation Type

Value: Enabled

Step 2. Create a vSwitch with PacketDirect enabled.

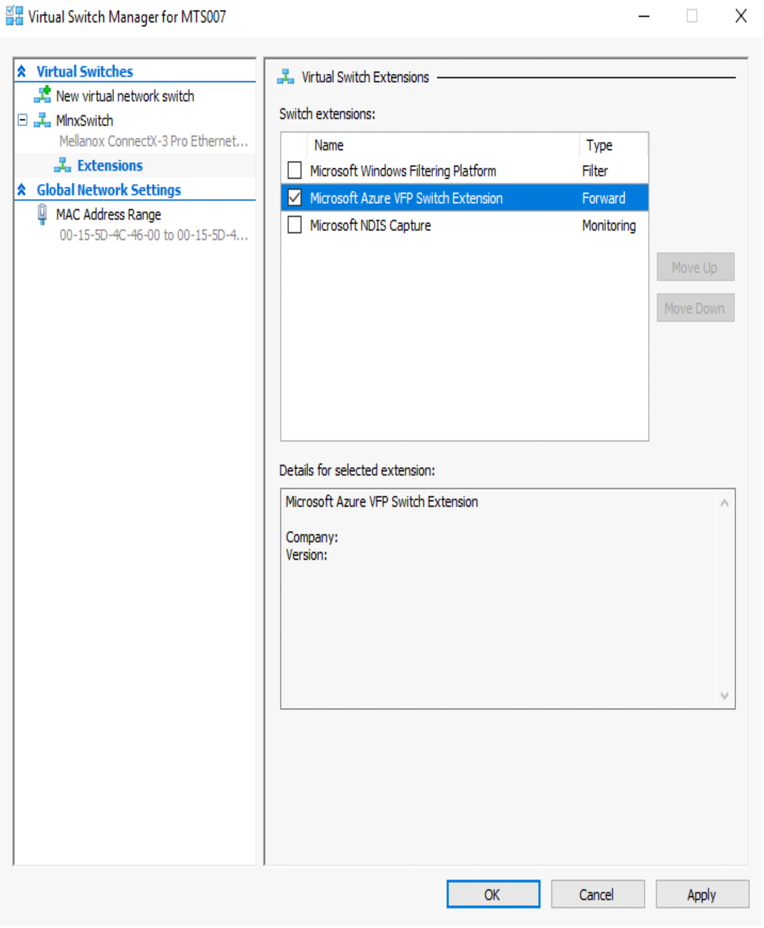
```
 PS $ New-VMSwitch <vSwitchName> -NetAdapterName <EthInterfaceName> -EnablePacketDirect
$true -AllowManagementOS $true
```

Step 3. Enable VFP extension:

- On the vSwitch.

```
 PS $ Enable-VMSwitchExtension -VmSwitchName <vSwitchName> -Name "Windows Azure VFP
Switch Extension"
```

- In the Hyper-V Manager: Action->Virtual Switch Manager...



Virtual Switch Manager for MTS007

Virtual Switches
New virtual network switch
MlnxSwitch
Mellanox ConnectX-3 Pro Ethernet...
Extensions
Global Network Settings
MAC Address Range
00-15-5D-4C-46-00 to 00-15-5D-4...
Virtual Switch Extensions
Move Up
Move Down
OK
Cancel
Apply

Switch extensions:

	Name	Type
□	Microsoft Windows Filtering Platform	Filter
☑	Microsoft Azure VFP Switch Extension	Forward
□	Microsoft NDIS Capture	Monitoring

Details for selected extension:

Microsoft Azure VFP Switch Extension

Company:
Version:

Step 4. Shut down the VM.

```
PS $ Stop-VM -Name <VMName> -Force -Confirm
```

Step 5. Add a virtual network adapter for the VM.

```
PS $ Add-VMNetworkAdapter -VMName <VMName> -SwitchName <vSwitchName> -StaticMacAddress
<StaticMAC Address>
```

Step 6. Start the VM.

```
PS $ Start-VM -Name <VMName>
```

Since VFP is enabled, without any forwarding rules, it will block all traffic going through the VM.

Follow the following steps to unblock the traffic

Step a. Find the port name for the VM.

```
 CMD > vfpctrl /list-vmswitch-port
 …....
 Port name              : E431C413-D31F-40EB-AD96-0B2D45FE34AA
 Port Friendly name     :
 Switch name            : 8B288106-9DB6-4720-B144-6CC32D53E0EC
 Switch Friendly name   : MlnxSwitch
 PortId                 : 3
 VMQ Usage              : 0
 SR-IOV Usage           : 0
 Port type              : Synthetic
  Port is Initialized.
  MAC Learning is Disabled.
 NIC name             : bd65960d-4215-4a4f-bddc-962a5d0e2fa0--e7199a49-6cca-
4d3c-a4cd-22907592527e
 NIC Friendly name    : testnic
 MTU                  : 1500
 MAC address          : 00-15-5D-4C-46-00
 VM name              : vm
 ......
 Command list-vmswitch-port succeeded!
```

Step 7. Disable the port to allow traffic.

```
CMD > vfpctrl /disable-port /port <PortName>
Command disable-port succeeded!
```

The port should be disabled after each reboot of the VM to allow traffic.

3.6 Configuration Using Registry Keys

Mellanox IPoIB and Ethernet drivers use registry keys to control the NIC operations. The registry keys receive default values during the installation of the Mellanox adapters. Most of the parameters are visible in the registry by default, however, certain parameters must be created in order to modify the default behavior of the Mellanox driver.

The adapter can be configured either from the User Interface (Device Manager -> Mellanox Adapter -> Right click -> Properties) or by setting the registry directly.

All Mellanox adapter parameters are located in the registry under the following registry key:

```
HKEY_LOCAL_MACHINE
\SYSTEM
\CurrentControlSet
\ Control
\ Class
\{4D36E972-E325-11CE-BFC1-08002bE10318}
\<Index>
```

The registry key can be divided into 4 different groups:

Table 16 - Registry Key Groups

Group	Description
Basic	Contains the basic configuration.
Offload Options	Controls the offloading operation that the NIC supports.
Performance Options	Controls the NIC operation in different environments and scenarios.
Flow Control Options	Controls the TCP/IP traffic.

Any registry key that starts with an asterisk ("*") is a well-known registry key. For more details regarding the registries, please refer to:

http://msdn.microsoft.com/en-us/library/ff570865(v=VS.85).aspx

3.6.1 Finding the Index Value of the HCA

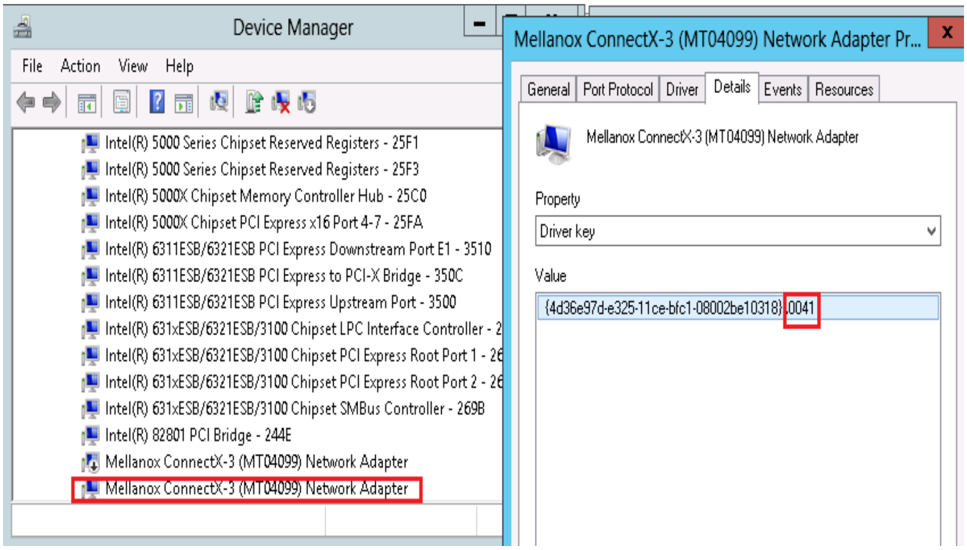
➢ ***To find the nn value of your HCA from the Device Manager please perform the following steps:***

Step 1. Open Device Manager, and go to System devices.

Step 2. Right click on a Mellanox -ConnectX® card -> properties.

Step 3. Go to Details tab.

Step 4. Select the Driver key, and obtain the nn number.
In the below example, the index equals 0041



Intel(R) 5000 Series Chipset Reserved Registers - 25F1
Intel(R) 5000 Series Chipset Reserved Registers - 25F3
Intel(R) 5000X Chipset Memory Controller Hub - 25C0
Intel(R) 5000X Chipset PCI Express x16 Port 4-7 - 25FA
Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E1 - 3510
Intel(R) 6311ESB/6321ESB PCI Express to PCI-X Bridge - 350C
Intel(R) 6311ESB/6321ESB PCI Express Upstream Port - 3500
Intel(R) 631xESB/6321ESB/3100 Chipset LPC Interface Controller - 2
Intel(R) 631xESB/6321ESB/3100 Chipset PCI Express Root Port 1 - 26
Intel(R) 631xESB/6321ESB/3100 Chipset PCI Express Root Port 2 - 26
Intel(R) 631xESB/6321ESB/3100 Chipset SMBus Controller - 269B
Intel(R) 82801 PCI Bridge - 244E
Mellanox ConnectX-3 (MT04099) Network Adapter
Mellanox ConnectX-3 (MT04099) Network Adapter

Mellanox ConnectX-3 (MT04099) Network Adapter

Property

Driver key

Value

{4d36e97d-e325-11ce-bfc1-08002be10318}\0041

3.6.2 Finding the Index Value of the Network Interface

To find the index value of your Network Interface from the Device Manager please perform the following steps:

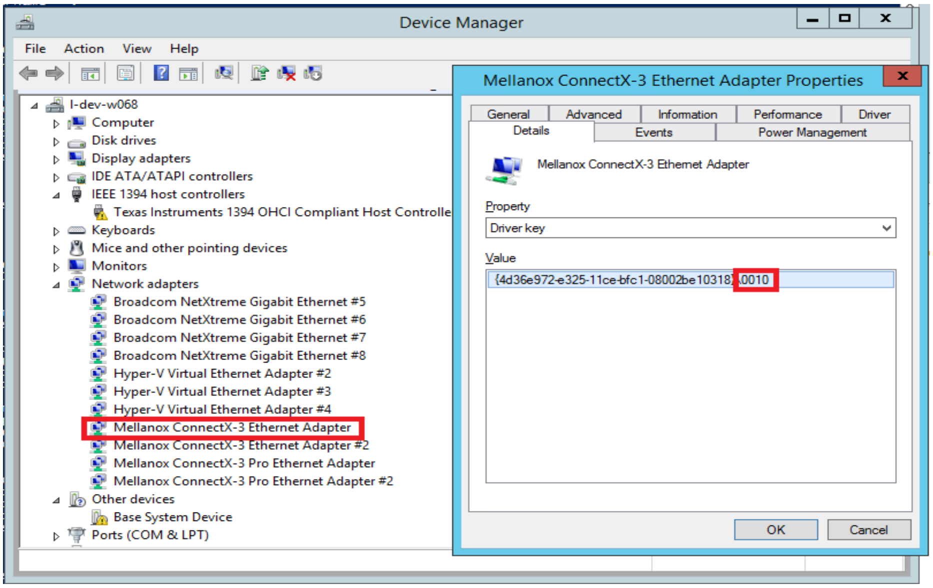
Step 1. Open Device Manager, and go to Network Adapters.

Step 2. Right click ->Properties on Mellanox Connect-X® Ethernet Adapter.

Step 3. Go to Details tab.

Step 4. Select the Driver key, and obtain the nn number.

In the below example, the index equals 0010


Device Manager
File Action View Help
l-dev-w068
Computer
Disk drives
Display adapters
IDE ATA/ATAPI controllers
IEEE 1394 host controllers
Texas Instruments 1394 OHCI Compliant Host Controlle
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Broadcom NetXtreme Gigabit Ethernet #5
Broadcom NetXtreme Gigabit Ethernet #6
Broadcom NetXtreme Gigabit Ethernet #7
Broadcom NetXtreme Gigabit Ethernet #8
Hyper-V Virtual Ethernet Adapter #2
Hyper-V Virtual Ethernet Adapter #3
Hyper-V Virtual Ethernet Adapter #4
Mellanox ConnectX-3 Ethernet Adapter
Mellanox ConnectX-3 Ethernet Adapter #2
Mellanox ConnectX-3 Pro Ethernet Adapter
Mellanox ConnectX-3 Pro Ethernet Adapter #2
Other devices
Base System Device
Ports (COM & LPT)
Mellanox ConnectX-3 Ethernet Adapter Properties
General
Advanced
Information
Performance
Driver
Details
Events
Power Management
Mellanox ConnectX-3 Ethernet Adapter
OK
Cancel

Property: Driver key

Value: {4d36e972-e325-11ce-bfc1-08002be10318}\0010

3.6.3 Basic Registry Keys

This group contains the registry keys that control the basic operations of the NIC.

Table 17 - Basic Registry Keys

Value Name	Default Value	Description
*JumboPacket	eth:1514 IPoIB:4096	The maximum size of a frame (or a packet) that can be sent over the wire. This is also known as the maximum transmission unit (MTU). The MTU may have a significant impact on the network's performance as a large packet can cause high latency. However, it can also reduce the CPU utilization and improve the wire efficiency. The standard Ethernet frame size is 1514 bytes, but Mellanox drivers support wide range of packet sizes. The valid values are: • Ethernet: 600 up to 9600 • IPoIB: 1500 up to 4092 **Notes**: • When the register is configured via VF running over Microsoft Hyper-V, the value cannot exceed the value configured in the host. • All the devices across the network (switches and routers) should support the same frame size. Be aware that different network devices calculate the frame size differently. Some devices include the header, i.e. information in the frame size, while others do not. Mellanox adapters do not include Ethernet header information in the frame size. (i.e when setting *JumboPacket to 1500, the actual frame size is 1514).
*ReceiveBuffers	eth:4096 IPoIB:512	The number of packets each ring receives. This parameter affects the memory consumption and the performance. Increasing this value can enhance receive performance, but also consumes more system memory. In case of lack of received buffers (dropped packets or out of order received packets), you can increase the number of received buffers. The valid values are 256 up to 4096.
*TransmitBuffers	eth:2048 IPoIB:2048	The number of packets each ring sends. Increasing this value can enhance transmission performance, but also consumes system memory. The valid values are 256 up to 4096.
*SpeedDuplex	7	The Speed and Duplex settings that a device supports. This registry key should not be changed and it can be used to query the device capability. Mellanox ConnectX device is set to 7 meaning10Gbps and Full Duplex. **Note**: Default value should not be modified.

Table 17 - Basic Registry Keys

Value Name	Default Value	Description
MaxNumOfMCList	eth:128 IPoIB:128	The number of multicast addresses that are filtered by the NIC. If the OS uses more multicast addresses than were defined, it sets the port to multicast promiscuous and the multicast addresses are filtered by OS at protocol level. The valid values are 64 up to 1024. **Note**: This registry value is not exposed via the UI.
*QOS	eth:1	Enables the NDIS Quality of Service (QoS) The valid values are: • 1: enable • 0: disable **Notes**: • This keyword is only valid for ConnectX-3 when using Windows Server 2012 and above. • This register cannot be configured via a VF that is running over Microsoft Hyper-V.
RxIntModerationProfile	eth:2 IPoIB:2	Enables the assignment of different interrupt moderation profiles for receive completions. Interrupt moderation can have a great effect on optimizing network throughput and CPU utilization. The valid values are: • 0: Low Latency Implies higher rate of interrupts to achieve better latency, or to handle scenarios where only a small number of streams are used. • 1: Moderate Interrupt moderation is set to midrange defaults to allow maximum throughput at minimum CPU utilization for common scenarios. • 2: Aggressive Interrupt moderation is set to maximal values to allow maximum throughput at minimum CPU utilization, for more intensive, multi-stream scenarios.

Table 17 - Basic Registry Keys

Value Name	Default Value	Description
TxIntModerationProfile	eth:1 IPoIB:1	Enables the assignment of different interrupt moderation profiles for send completions. Interrupt moderation can have great effect on optimizing network throughput and CPU utilization. The valid values are: • 0: Low Latency Implies higher rate of interrupts to achieve better latency, or to handle scenarios where only a small number of streams are used. • 1: Moderate Interrupt moderation is set to midrange defaults to allow maximum throughput at minimum CPU utilization for common scenarios. • 2: Aggressive Interrupt moderation is set to maximal values to allow maximum throughput at minimum CPU utilization for more intensive, multi-stream scenarios.

3.6.4 Off-load Registry Keys

This group of registry keys allows the administrator to specify which TCP/IP offload settings are handled by the adapter rather than by the operating system.

Enabling offloading services increases transmission performance. Due to offload tasks (such as checksum calculations) performed by adapter hardware rather than by the operating system (and, therefore, with lower latency). In addition, CPU resources become more available for other tasks.

Table 18 - Off-load Registry Keys

Value Name	Default Value	Description
*LsoV1IPv4	1	Large Send Offload Version 1 (IPv4). The valid values are: • 0: disable • 1: enable
*LsoV2IPv4	1	Large Send Offload Version 2 (IPv4). The valid values are: • 0: disable • 1: enable
*LsoV2IPv6	1	Large Send Offload Version 2 (IPv6). The valid values are: • 0: disable • 1: enable

Table 18 - Off-load Registry Keys

Value Name	Default Value	Description
LSOSize	eth:64000 IPoIB:64000	The maximum number of bytes that the TCP/IP stack can pass to an adapter in a single packet. This value affects the memory consumption and the NIC performance. The valid values are MTU+1024 up to 64000. **Note:** This registry key is not exposed to the user via the UI. If LSOSize is smaller than MTU+1024, LSO will be disabled.
LSOMinSegment	eth:2 IPoIB:2	The minimum number of segments that a large TCP packet must be divisible by, before the transport can offload it to a NIC for segmentation. The valid values are 2 up to 32. **Note**: This registry key is not exposed to the user via the UI.
LSOTcpOptions	eth:1 IPoIB:1	Enables that the miniport driver to segment a large TCP packet whose TCP header contains TCP options. The valid values are: • 0: disable • 1: enable **Note**: This registry key is not exposed to the user via the UI.
LSOIpOptions	eth:1 IPoIB:1	Enables its NIC to segment a large TCP packet whose IP header contains IP options. The valid values are: • 0: disable • 1: enable **Note**: This registry key is not exposed to the user via the UI.
*IPChecksumOffload-IPv4	eth:3 IPoIB:3	Specifies whether the device performs the calculation of IPv4 checksums. The valid values are: • 0: (disable) • 1: (Tx Enable) • 2: (Rx Enable) • 3: (Tx and Rx enable)

Table 18 - Off-load Registry Keys

Value Name	Default Value	Description
*TCPUDPChecksumOffloadIPv4	eth:3 IPoIB:3	Specifies whether the device performs the calculation of TCP or UDP checksum over IPv4. The valid values are: • 0: (disable) • 1: (Tx Enable) • 2: (Rx Enable) • 3: (Tx and Rx enable)
*TCPUDPChecksumOffloadIPv6	eth:3 IPoIB:3	Specifies whether the device performs the calculation of TCP or UDP checksum over IPv6. The valid values are: • 0: (disable) • 1: (Tx Enable) • 2: (Rx Enable) • 3: (Tx and Rx enable)
ParentBusRegPath	HKLM\SYSTEM\CurrentControlSet\Control\Class\{4d36e97d-e325-11ce-bfc1-08002be10318}\0073	TCP checksum off-load IP-IP.
*RssOnHostVPorts	1	Virtual Machine Multiple Queue (VMMQ) HW Offload The valid values are: • 0: disable • 1: enable

3.6.5 Performance Registry Keys

This group of registry keys configures parameters that can improve adapter performance.

Table 19 - Performance Registry Keys

Value Name	Default Value	Description
RecvCompletionMethod	eth:1 IPoIB:1	Sets the completion methods of the receive packets, and it affects network throughput and CPU utilization. The supported methods are: • Polling - increases the CPU utilization, because the system polls the received rings for incoming packets; however, it may increase the network bandwidth since the incoming packet is handled faster. • Adaptive - combines the interrupt and polling methods dynamically, depending on traffic type and network usage. The valid values are: • 0: polling • 1: adaptive

Table 19 - Performance Registry Keys

Value Name	Default Value	Description
*InterruptModeration	eth:1 IPoIB:1	Sets the rate at which the controller moderates or delays the generation of interrupts, making it possible to optimize network throughput and CPU utilization. When disabled, the interrupt moderation of the system generates an interrupt when the packet is received. In this mode, the CPU utilization is increased at higher data rates, because the system must handle a larger number of interrupts. However, the latency is decreased, since that packet is processed more quickly. When 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 the passing of 10 micro seconds from receiving the first packet. The valid values are: • 0: disable • 1: enable
RxIntModeration	eth:2 IPoIB:2	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 traffic type and network usage. Choosing a different setting may improve network and system performance in certain configurations. The valid values are: • 1: static • 2: adaptive The interrupt moderation count and time are configured dynamically, based on traffic types and rate.
pkt_rate_low	eth:150000 IPoIB:150000	Sets the packet rate below which the traffic is considered as latency traffic when using adaptive interrupt moderation. The valid values are 100 up to 1000000. **Note**: This registry value is not exposed via the UI.
pkt_rate_high	eth:170000 IPoIB:170000	Sets the packet rate above which the traffic is considered as bandwidth traffic. when using adaptive interrupt moderation. The valid values are 100 up to 1000000. **Note**: This registry value is not exposed via the UI.

Table 19 - Performance Registry Keys

Value Name	Default Value	Description
*RSS	eth:1 IPoIB:1	Sets the driver to use Receive Side Scaling (RSS) mode to improve the performance of handling incoming packets. This mode allows the adapter port to utilize the multiple CPUs in a multi-core system for receiving incoming packets and steering them to their destination. RSS can significantly improve the number of transactions per second, the number of connections per second, and the network throughput. This parameter can be set to one of two values: • 1: enable (default) Sets RSS Mode. • 0: disable The hardware is configured once to use the Toeplitz hash function and the indirection table is never changed. **Note**: the I/O Acceleration Technology (IOAT) is not functional in this mode.
TxHashDisrtibution	3	Sets the algorithm which is used to distribute the send-packets on different send rings. The adapter uses 3 methods: • 1: Size In this method only 2 Tx rings are used. The send-packets are distributed, based on the packet size. Packets that are smaller than 128 bytes use one ring, while the larger packets use the other ring. • 2: Hash In this method the adapter calculates a hash value based on the destination IP, the TCP source and the destination port. If the packet type is not IP, the packet uses ring number 0. • 3: Hash and size In this method for each hash value, 2 rings are used: one for small packets and another one for larger packets. The valid values are: • 1: size • 2: hash • 3: hash and size **Note**: This registry value is not exposed via the UI.

Table 19 - Performance Registry Keys

Value Name	Default Value	Description
RxSmallPacketBypass	eth:0 IPoIB:0	Specifies whether received small packets bypass larger packets when indicating received packet to NDIS. This mode is useful in bi-directional applications. Enabling this mode ensures that the ACK packet will bypass the regular packet and TCP/IP stack will issue the next packet more quickly. The valid values are: • 0: disable • 1: enable **Note**: This registry value is not exposed via the UI.
ReturnPacketThreshold	eth:341 IPoIB:341	The allowed number of free received packets on the rings. Any number above it will cause the driver to return the packet to the hardware immediately. When the value is set to 0, the adapter uses 2/3 of the received ring size. The valid values are: 0 to 4096. **Note**: This registry value is not exposed via the UI.
NumTcb	eth:16 IPoIB:16	The number of send buffers that the driver allocates for sending purposes. Each buffer is in LSO size, if LSO is enabled, or in MTU size, otherwise. The valid values are 1 up to 64. **Note**: This registry value is not exposed via the UI.
ThreadPoll	eth:10000 IPoIB:10000	The number of cycles that should be passed without receiving any packet before the polling mechanism stops when using polling completion method for receiving. Afterwards, receiving new packets will generate an interrupt that reschedules the polling mechanism. The valid values are 0 up to 200000. **Note**: This registry value is not exposed via the UI.
AverageFactor	eth:16 IPoIB:16	The weight of the last polling in the decision whether to continue the polling or give up when using polling completion method for receiving. The valid values are 0 up to 256. **Note**: This registry value is not exposed via the UI.

Table 19 - Performance Registry Keys

Value Name	Default Value	Description
AveragePollThreshold	eth:10 IPoIB:10	The average threshold polling number when using polling completion method for receiving. If the average number is higher than this value, the adapter continues to poll. The valid values are 0 up to 1000. **Note**: This registry value is not exposed via the UI.
ThisPollThreshold	eth:100 IPoIB:100	The threshold number of the last polling cycle when using polling completion method for receiving. If the number of packets received in the last polling cycle is higher than this value, the adapter continues to poll The valid values are 0 up to 1000. **Note**: This registry value is not exposed via the UI.
*HeaderDataSplit	eth:0 IPoIB:0	Enables the driver to use header data split. In this mode, the adapter uses two buffers to receive the packet. The first buffer holds the header, while the second buffer holds the data. This method reduces the cache hits and improves the performance. The valid values are: • 0: disable • 1: enable **Note**: This registry value is not exposed via the UI.
VlanId	eth:0 IPoIB:0	Enables packets with VlanId. It is used when no team intermediate driver is used. The valid values are: • 0: disable No Vlan Id is passed. • 1-4095 Valid Vlan Id that will be passed. **Note**: This register cannot be configured via a VF that is running over Microsoft Hyper-V.
TxForwardingProcessor	Automatically selected based on RSS configuration	The processor that will be used to forward the packets sent by the forwarding thread. Default is based on number of rings and number of cores on the machine. **Note**: This registry value is not exposed via the UI.

Table 19 - Performance Registry Keys

Value Name	Default Value	Description
DefaultRecvRingProcessor	Automatically selected based on RSS configuration	The type of processor which will be used for the default Receive ring. This variable handles packets that are not handled by RSS. This can be non TCP/UDP packets or even UDP packets, if they are configured to use the default ring. **Note**: This registry value is not exposed via the UI.
TxInterruptProcessor	Automatically selected based on RSS configuration	The type of processor which will be used to handle the TX completions. The default is based on a number of rings and a number of cores on the machine. **Note**: This registry value is not exposed via the UI.
*NumRSSQueues	eth:8 IPoIB:8	The maximum number of the RSS queues that the device should use. **Note**: This registry key is only in Windows Server 2012 and above.
BlueFlame	eth:1 IPoIB:1	The latency-critical Send WQEs to the device. When a BlueFlame is used, the WQEs are written directly to the PCI BAR of the device (in addition to memory), so that the device may handle them without having to access memory, thus shortening the execution latency. For best performance, it is recommended to use the BlueFlame when the HCA is lightly loaded. For high-bandwidth scenarios, it is recommended to use regular posting (without BlueFlame). The valid values are: • 0: disable • 1: enable **Note**: This registry value is not exposed via the UI.
*MaxRSSProcessors	eth:8 IPoIB:8	The maximum number of RSS processors. **Note**: This registry key is only in Windows Server 2012 and above.

3.6.6 Ethernet Registry Keys

The following section describes the registry keys that are only relevant to Ethernet driver.

Table 20 - Ethernet Registry Keys

Value Name	Default Value	Description
RoceMaxFrameSize	1024	The maximum size of a frame (or a packet) that can be sent by the RoCE protocol (a.k.a Maximum Transmission Unit (MTU). Using larger RoCE MTU will improve the performance; however, one must ensure that the entire system, including switches, supports the defined MTU. Ethernet packet uses the general MTU value, whereas the RoCE packet uses the RoCE MTU The valid values are: • 256 • 512 • 1024 • 2048 **Note**: This registry key is supported only in Ethernet drivers.
*PriorityVLANTag	3 (Packet Priority & VLAN Enabled)	Enables sending and receiving IEEE 802.3ac tagged frames, which include: • 802.1p QoS (Quality of Service) tags for priority-tagged packets. • 802.1Q tags for VLANs. When this feature is enabled, the Mellanox driver supports sending and receiving a packet with VLAN and QoS tag. **Note**: This register cannot be configured via a VF that is running over Microsoft Hyper-V.
PromiscuousVlan	0	Specifies whether a promiscuous VLAN is enabled or not. When this parameter is set, all the packets with VLAN tags are passed to an upper level without executing any filtering. The valid values are: • 0: disable • 1: enable **Note**: This registry value is not exposed via the UI.

Table 20 - Ethernet Registry Keys

Value Name	Default Value	Description
UseRSSForRawIP	1	The execution of RSS on UDP and Raw IP packets. In a forwarding scenario, one can improve the performance by disabling RSS on UDP or a raw packet. In such a case, the entire receive processing of these packets is done on the processor that was defined in DefaultRecvRingProcessor registry key. The valid values are: • 0: disable • 1: enable This is also relevant for IPoIB. **Note**: This registry value is not exposed via the UI.
UseRSSForUDP	1	Used to execute RSS on UDP and Raw IP packet. In forwarding scenario you can improve the performance by disable RSS on UDP or raw packet. In such a case all the receive processing of these packets is done on the processor that was defined in DefaultRecvRingProcessor registry key. The valid values are: • 0:disabled • 1: Enabled **Note**: This registry value is not exposed via UI.
SingleStream	0	It used to get the maximum bandwidth when using single stream traffic. When setting the registry key to enabled the driver will forward the sending packet to another CPU. This decrease the CPU utilization of the sender and allows sending in higher rate The valid values are: • 0:disabled • 1: Enabled **Note**: only relevant for Ethernet and IPoIB

Table 20 - Ethernet Registry Keys

Value Name	Default Value	Description
IgnoreFCS	0	The valid values are: • 0: disabled • 1: enabled When enabled, the device is configured to: 1. Pass packets with FCS error to the driver (the default is to drop FCS corrupted packets). 2. Pass the 4 bytes of the FCS to the driver (the default is to strip them).

3.6.6.1 Flow Control Options

This group of registry keys allows the administrator to control the TCP/IP traffic by pausing frame transmitting and/or receiving operations. By enabling the Flow Control mechanism, the adapters can overcome any TCP/IP issues and eliminate the risk of data loss.

Table 21 - Flow Control Options

Value Name	Default Value	Description
*FlowControl	0	When Rx Pause is enabled, the receiving adapter generates a flow control frame when its received queue reaches a pre-defined limit. The flow control frame is sent to the sending adapter. When TX Pause is enabled, the sending adapter pauses the transmission if it receives a flow control frame from a link partner. The valid values are: • 0: Flow control is disabled • 1: Tx Flow control is Enabled • 2: Rx Flow control is enabled • 3: Rx & Tx Flow control is enabled
PerPriRxPause	0	When Per Priority Rx Pause is configured, the receiving adapter generates a flow control frame when its priority received queue reaches a pre-defined limit. The flow control frame is sent to the sending adapter. **Notes**: • This registry value is not exposed via the UI. • RxPause and PerPriRxPause are mutual exclusive (i.e. at most, only one of them can be set).

Table 21 - Flow Control Options

Value Name	Default Value	Description
PerPriTxPause	0	When Per Priority TX Pause is configured, the sending adapter pauses the transmission of a specific priority, if it receives a flow control frame from a link partner. **Notes**: • This registry value is not exposed via the UI. • TxPause and PerPriTxPause are mutual exclusive (i.e. at most, only one of them can be set).

3.6.6.2 VMQ Options

This section describes the registry keys that are used to control the NDIS Virtual Machine Queue (VMQ). The VMQ supports Microsoft Hyper-V network performance, and is supported on Windows Server 2012, 2012 R2 and 2016.

For more details about VMQ please refer to Microsoft web site,
http://msdn.microsoft.com/en-us/library/windows/hardware/ff571034(v=vs.85).aspx

Table 22 - VMQ Options

Value Name	Default Value	Description
*VMQ	1	The support for the virtual machine queue (VMQ) features of the network adapter. The valid values are: • 1: enable • 0: disable
*RssOrVmqPreference	0	Specifies whether VMQ capabilities should be enabled instead of receive-side scaling (RSS) capabilities. The valid values are: • 0: Report RSS capabilities • 1: Report VMQ capabilities **Note**: This registry value is not exposed via the UI.
*VMQLookaheadSplit	1	Specifies whether the driver enables or disables the ability to split the receive buffers into lookahead and post-lookahead buffers. The valid values are: • 0: disable • 1: enable
*VMQVlanFiltering	1	Specifies whether the device enables or disables the ability to filter network packets by using the VLAN identifier in the media access control (MAC) header. The valid values are: • 0: disable • 1: enable

Table 22 - VMQ Options

Value Name	Default Value	Description
MaxNumVmqs	127	The number of VMQs that the device supports in parallel. This parameter can effect memory consumption of the interface, since for each VMQ, the driver creates a separate receive ring and an allocate buffer for it. In order to minimize the memory consumption, one can reduce the number of VMs that use VMQ in parallel. However, this can affect the performance. The valid values are 1 up to 127. **Note**: This registry value is not exposed via the UI.
MaxNumMacAddrFilters	127	The number of different MAC addresses that the physical port supports. This registry key affects the number of supported MAC addresses that is reported to the OS. The valid values are 1 up to 127. **Note**: This registry value is not exposed via the UI.
MaxNumVlanFilters	125	The number of VLANs that are supported for each port. The valid values are 1 up to 127. **Note**: This registry value is not exposed via the UI.

3.6.7 IPoIB Registry Keys

The following section describes the registry keys that are unique to IPoIB.

Table 23 - IPoIB Registry Keys

Value Name	Default Value	Description
GUIDMask	0xE7	Controls the way the MAC is generated for IPoIB interface. The driver uses the 8 bytes GUID to generate 6 bytes MAC. This value should be either 0 or contain exactly 6 non-zero digits, using binary representation. Zero (0) mask indicates its default value: 0xb' 11100111. That is, to take all, except intermediate bytes of GUID to form the MAC address. In case of an improper mask, the driver uses the default one. **Note**: This registry value is not exposed via the UI.

Table 23 - IPoIB Registry Keys

Value Name	Default Value	Description
MediumType802_3	0	Controls the way the interface is exposed to an upper level. By default, the IPoIB is exposed as an InfiniBand interface. The user can change it and cause the interface to be an Ethernet interface by setting this registry key. The valid values are: • 0 - the interface is exposed as NdisPhysicalMediumInfiniband • 1 - the interface is exposed as NdisPhysicalMedium802_3. **Note**: This registry value is not exposed via the UI.
SaTimeout	1000	The time, in milliseconds, before retransmitting an SA query request. The valid values are 250 up to 60000.
SaRetries	10	The number of times to retry an SA query request. The valid values are 1 up to 64.
McastIgmpMldGeneralQueryInterval	3	The number of runs of the multicast monitor before a general query is initiated. This monitor runs every 30 seconds. The valid values are 1 up to 10.
LocalEndpointMaxAge	5	The maximum number of runs of the local end point DB monitor, before an unused local endpoint is removed. The endpoint age is zeroed when it is used as a source in the send flow or a destination in the receive flow. Each monitor run will increment the age of all non VMQ local endpoints. When LocalEndpointMaxAge is reached - the endpoint will be removed. The valid values are 1 up to 20. **Note**: This registry value is not exposed via the UI.
LocalEndpointMonitorInterval	60000	The time interval (in ms) between each 2 runs of the local end point DB monitor, for aging, unused local endpoints. Each run will increment the age of all non VMQ local endpoints. The valid values are 10000 up to 1200000. **Note**: This registry value is not exposed via the UI.
EnableQPR	0	Enables query path record. The valid values are: • 0 - disable • 1 - enable

Table 23 - IPoIB Registry Keys

Value Name	Default Value	Description
McastQueryResponseInterval	2	The number of runs of the multicast monitor (which runs every 30 seconds) allowed until a response to the IGMP/ MLD queries is received. If after this period a response is not received, the driver leaves the multicast group. The valid values are 1 up to 10. **Note**: This registry value is not exposed via the UI.

3.6.8 General Registry Values

This section provides information on general registry keys that affect Mellanox driver operation.

Table 24 - General Registry Values

Value Name	Default Value	Description
MaxNumRssCpus	4	The number of CPUs that participate in the RSS. The Mellanox adapter can open multiple receive rings, each ring can be processed by a different processor. When RSS is disabled, the system opens a single Rx ring. The Rx ring number that is configured should be powered of two and less than the number of processors on the system. Value Type: DWORD The valid values are 1 up to number of processors on the system.
RssBaseCpu	1	The CPU number of the first CPU that the RSS can use. NDIS uses the default value of 0 for the base CPU number, however this value is configurable and can be changed. The Mellanox adapter reads this value from registry and sets it to NDIS on driver start-up. Value Type: DWORD The valid values are 0 up to the number of processors on the system.
CheckFwVersion	1	Configures the Mellanox driver to skip validation of the FW compatibility to the driver version. Skipping this check-up is not recommended and can cause unexpected behavior. It can be used for testing purposes only. Value Type: DWORD The valid values are: • 0: Don't check • 1: Check

Table 24 - General Registry Values

Value Name	Default Value	Description
MaximumWorkingThreads	2	The number of working threads which can work simultaneously on receive polling. By default, the Mellanox driver creates a working thread for each Rx rings if polling or adaptive receive completion is set. Value Type: DWORD The valid values are 1 up to number of Rx rings.

3.6.9 MLX BUS Registry Keys

3.6.9.1 SR-IOV Registry Keys

SR-IOV feature can be controlled, on a machine level or per device, using the same set of Registry Keys. However, only one level must be used consistently to control SR-IOV feature. If both levels were used, the per-machine level of configuration will be enforced by the driver.

Registry Keys location for machine configuration:

```
HKLM\SYSTEM\CurrentControlSet\Services\mlx4_bus\Parameters
```

Registry Keys location for device configuration:

```
HKLM\SYSTEM\CurrentControlSet\Control\Class\{4d36e97d-e325-11ce-bfc1-
08002be10318}\<nn>\Parameters
```

For more information on how to find device index nn, please refer to 3.6.1 “Finding the Index Value of the HCA,” on page 130

Table 25 - SR-IOV Registry Keys

Key Name	Key Type	Values	Description
PermitSriov	REG_DWORD	• 0 = Disabled • 1 = Enabled (default)	Configures the SR-IOV mode.

Table 25 - SR-IOV Registry Keys

Key Name	Key Type	Values	Description
SriovPortMode	REG_DWORD	• 0 = auto_port1 • 1 = auto_port2 • 2 = manual (default)	Configures the number of VFs to be enabled by the bus driver to each port. **Note**: In auto_portX mode, port X will have the number of VFs according to the burnt value in the device and the other port will have no SR-IOV and it will support native Ethernet (i.e. no RoCE). Setting this parameter to "Manual" will configure the number of VFs for each port according to the registry key MaxVFPortX. **Note**: The number of VFs can be configured both on a Mellanox bus driver level and Network Interface level (i.e using Set-NetAdapterSriov Powershell cmdlet). The number of VFs actually available to the Network Interface is the minimum value between mellanox bus driver configuration and Network Interface configuration. For example, if 8 VFs support was burnt in firmware, SriovPortMode is auto_port1, and Network Interface was allowed 32 VFs using SetNetAdapterSriov Powershell cmdlet, the actual number of VFs available to Network Interface will be 8.
MaxVFPort1 MaxVFPort2	REG_DWORD	• 16=(default)	MaxVFPort<i> The maximum number of VFs that are allowed per port. This is the number of VFs the bus driver will open when working in manual mode. **Note**: If the total number of VFs requested is larger than the number of VFs burnt in firmware, each port X(1\2) will have the number of VFs according to the following formula: (MaxVFPortX / (MaxVPort1+MaxVPort2)*number of VFs burnt in firmware.

3.6.9.2 RoCE Options

The following registry configuration is available for RoCE under:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\mlx4_bus\Parameters\Roce.

This registry is per-driver and it will apply to all available adapters.

Table 26 - RoCE Options

Parameters	Parameter type	Description	Allowed Values and Default
roce_mode	DWORD	Sets the RoCE mode. The following are the possible RoCE modes: • RoCE MAC Based (v1) • RoCE IP Based (v1) • RoCE over IP (v1.5) • RoCE over UDP (v2) • No RoCE	• RoCE MAC Based = 0 • RoCE IP Based = 5 • RoCE over IP = 1 • RoCE over UDP = 2 • No RoCE = 4 • **Default:** No RoCE **NOTE:** The default value depends on the WinOF package used.
roce_udp_dport	DWORD	Sets the RoCE v2 UDP destination port. Note that in order to communicate with RoCE v2, all machines in a fabric must be configured with the same value for the UDP port number.	• 1 - 65535 • **Default (IANA Port)**: 4791

3.6.9.3 NIC Resiliency Registry Keys

Table 27 - NIC Resiliency Registry Keys

Key Name	Key Type	Values	Description
DeviceRxStallWatermark	DWROD	0-8000 Default: 0	Time period for a single receive packet processing that indicates that the packet is about to become stalled. Value is given in mSec. 0x0 - indicates that processing time is not monitored.
DeviceRxStallTimeout	DWROD	0-8000 Default: 1000	Time period for a single receive packet processing that indicates that the device is not responsive. Value is given in mSec. 0x0 - indicates that processing time is not monitored.

3.6.9.4 General Registry Keys

Registry Keys location for machine configuration:

```
HKLM\SYSTEM\CurrentControlSet\Services\mlx4_bus\Parameters
```

Table 28 - General Registry Keys

Key Name	Key Type	Values	Description
AllowResetOnError	DWORD	• 0 - disable (default) • 1 - enable	When enabled, this setting will allow an SR-IOV- IB guest VM driver to gracefully recover from a case where the hypervisor driver is stuck by resetting the guest driver. otherwise, when a hypervisor is stuck the VM will require a restart to recover. **Caution**: This setting cannot be enabled when user-space RDMA applications such as MPI are running in the VM.
UpdateGIDTimerFrequency	DWORD	• 0-10000 • Default: 3000	Polling interval in milliseconds of local IP-address changes for updating RDMA IP-based GIDs.

3.7 Software Development Kit (SDK)

Software Development Kit (SDK) is 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 compile the examples provided with the SDK, you must install Windows Driver Kit (WDK) version 8.1 and higher over Visual Studio 2015.

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.

> IBAL API is no longer supported. The user should program the application over the ND API instead.

3.7.1 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 RDMA application. NDI exposes the advanced capabilities of the Mellanox networking devices and allows applications to leverage advances of RDMA.

Both RoCE and InfiniBand (IB) can implement NDI.

For further information, please refer to:

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

For code examples using NDI, you may refer to:

https://msdn.microsoft.com/library/cc853440(v=vs.85).aspx

3.7.2 Win-Linux nd_rping Test

The purpose of this test is to check interoperability between Linux and Windows via an RDMA ping. The Windows nd_rping was ported from Linux's RDMACM example: rping.c

- Windows
  - If you wish to use a built-in nd_rping.exe, you may find it in: Program Files\Mellanox\MLNX_VPI\IB\Tools
  - If you wish to build the nd_rping.exe from scratch, you can build it using the SDK example: choose the machine's OS in the configuration manager of the solution, and build the nd_rping.exe.
- Linux

  Installing the MLNX_OFED on a Linux server will also provide the "rping.exe" application.

3.7.2.1 Test Running

In order to run the test, follow the steps below:

1. Connect two servers to Mellanox adapters.
2. Verify ping between the two servers.
3. Configure the ROCE version to be:
   a. RoCE V1 (over IP):
      i. Linux side - V1

ii. Win side - V1.25

b. RoCE V2:

i. Linux side - V2

ii. Win side - V2

iii.Verify that ROCE udp_port is the same on the two servers. For the registry key, refer to Table 26 - "RoCE Options," on page 151.

4. Select the server side and the client side, and run accordingly:

a. Server:

```
nd_rping/rping -s [-v -V -d] [-S size] [-C count] [-a addr] [-p port]
```

b. Client:

```
nd_rping/rping -c [-v -V -d] [-S size] [-C count] -a addr  [-p port]
```

Executable Options:

Letter	Usage
-s	Server side
-P	Persistent server mode allowing multiple connections
-c	Client side
-a	Address
-p	Port

Debug Extensions:

Letter	Usage
-v	Displays ping data to stdout every test cycle
-V	Validates ping data every test cycle
-d	Shows debug prints to stdout
-S	Indicates ping data size - must be < (64*1024)
-C	Indicates the number of ping cycles to perform

Example:

➢ ***Linux server:***

```
rping -v -s -a <IP address> -C 10
```

➢ ***Windows client:***

```
nd_rping -v -c -a <same IP as above> -C 10
```

3.8 Performance Tuning and Counters

For further information on WinOF performance, please refer to the Performance Tuning Guide for Mellanox Network Adapters.

This section 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 section. 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.

3.8.1 General Performance Optimization and Tuning

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

3.8.1.1 Mellanox Specific Extensions to the ND Interface

IND2QueuePairsPool

The interface is an extension to the Network Direct SPI version 2. It reduces the creation time of the IND2QueuePair and IND2CompletionQueue interfaces, hence improves the client-server connection establishment time.

The interface exposes a pool of pre-allocated IND2QueuePair and IND2CompletionQueue interfaces associated with it. Pre-allocation is done using a background thread when a pre-configured threshold is reached.

The API for this interface is documented in the SDK header file ndspi_ext_mlx.h.

➢ ***Using IND2QueuePairsPool:***

1. Create a pool using IND2Adapter: QueryInterface with IID_IND2QueuePairsPool.
2. Set pool configuration using the SetQueuePairParams and SetCompletionQueueParams methods.
3. Set background creation thresholds using the SetLimits method
4. Fill the pool using the Fill method.
5. Create items IND2QueuePair and IND2CompletionQueue associated with it using the CreateObjects method.

- Statistics about the utilization of the resource pool are available to allow the programmer to select optimal thresholds

3.8.1.2 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.
```

3.8.1.3 Enable RSS

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

```
"netsh int tcp set global rss = enabled"
```

3.8.1.4 Tuning the IPoIB Network Adapter

The IPoIB Network Adapter tuning can be performed either during installation by modifying some of Windows registries as explained in Section 3.8.1.2, "Registry Tuning", on page 155. or can be set post-installation manually.

➢ ***To improve the network adapter performance, activate the performance tuning tool as follows:***

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

Step 2. Open "Network Adapters".

Step 3. Select Mellanox IPoIB adapter, right click and select Properties.

Step 4. Select the "Performance tab".

Step 5. Choose one of the tuning scenarios:

- Single port traffic - Improves performance for running single port traffic each time.
- Dual port traffic - Improves performance for running traffic on both ports simultaneously.
- Forwarding traffic - Improves performance for running scenarios that involve both ports (for example: via IXIA)
- Multicast traffic - Improves performance when the main traffic runs on multicast.

Step 6. Click on "Run Tuning" button.

Clicking the "Run Tuning" button changes several registry entries (described below), and checks for system services that may decrease network performance. It also generates 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).

A reboot may be required for the changes to take effect.

3.8.1.5 Tuning the Ethernet Network Adapter

The Ethernet Network Adapter general tuning can be performed during installation by modifying some of Windows registries as explained in section "Registry Tuning" on page 32. Specific scenarios tuning can be set post-installation manually.

➢ ***To improve the network adapter performance, activate the performance tuning tool as follows:***

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

Step 2. Open "Network Adapters".

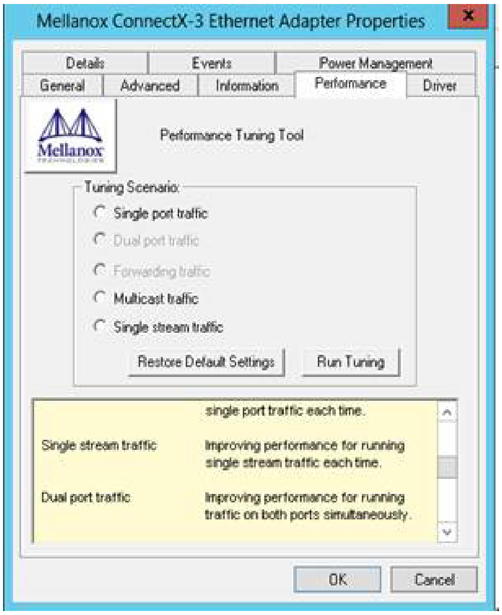
Step 3. Select Mellanox Ethernet adapter, right click and select Properties.

Step 4. Select the "Performance tab".

Step 5. Choose one of the tuning scenarios:

- Single port traffic - Improves performance for running single port traffic each time.
- Single stream traffic - Optimizes tuning for applications with single connection.
- Dual port traffic - Improves performance for running traffic on both ports simultaneously.
- Forwarding traffic - Improves performance for running scenarios that involve both ports (for example: via IXIA)
- Multicast traffic - Improves performance when the main traffic runs on multicast.

6. Click on “Run Tuning” button.



	single port traffic each time.
Single stream traffic	Improving performance for running single stream traffic each time.
Dual port traffic	Improving performance for running traffic on both ports simultaneously.

Clicking the "Run Tuning" button activates the general tuning as explained above and changes several driver registry entries for the current adapter and its sibling device once the sibling is an Ethernet device as well. It also generates 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.

3.8.1.5.1 Performance Tuning Tool Application

You can also activate the performance tuning through a script called perf_tuning.exe. This script has 4 options, which include the 3 scenarios described above and an additional manual tuning through which you can set the RSS base and number of processors for each Ethernet adapter. The adapters you wish to tune are supplied to the script by their name according to the "Network Connections".

Synopsis

```
perf_tuning.exe -s -c1 <first connection name> [-c2 <second connection name>]
perf_tuning.exe -d -c1 <first connection name> -c2 <second connection name>
perf_tuning.exe -f -c1 <first connection name> -c2 <second connection name>
perf_tuning.exe -m -c1 <first connection name> -b <base RSS processor number> -n <number of RSS
processors>
perf_tuning -st -c1 <first connection name> [-c2 <second connection name>]
```

Options

Table 29 - Performance Tuning Tool Application Options

Flag	Description
-s	Single port traffic scenario. This option can be followed by one or two connection names. The tuning will restore the default settings on the second connection and performed on the first connection. This option automatically sets: • SendCompletionMethod = 0 • RecvCompletionMethod = 2 • *ReceiveBuffers = 1024 • In Operating Systems support NDIS6.3: RssProfile = 4 Additionally, this option chooses the best processors to assign to: • DefaultRecvRingProcessor • TxInterruptProcessor • TxForwardingProcessor • In Operating Systems support NDIS6.2: RssBaseProcNumber MaxRssProcessors • In Operating Systems support NDIS6.3: NumRSSQueues RssMaxProcNumber

Table 29 - Performance Tuning Tool Application Options

Flag	Description
-d	Dual port traffic scenario. This option must be followed by two connection names. The tuning in this case is codependent. This option automatically sets: • SendCompletionMethod = 0 • RecvCompletionMethod = 2 • *ReceiveBuffers = 1024 • In Operating Systems support NDIS6.3: RssProfile = 4 Additionally, this option chooses the best processors to assign to: • DefaultRecvRingProcessor • TxForwardingProcessor • In Operating Systems support NDIS6.2: RssBaseProcNumber MaxRssProcessors • In Operating Systems support NDIS6.3: NumRSSQueues RssMaxProcNumber
-f	Forwarding traffic scenario. This option must be followed by two connection names. The tuning in this case is codependent. This option automatically sets: • SendCompletionMethod = 1 • RecvCompletionMethod = 0 • *ReceiveBuffers = 4096 • UseRSSForRawIP = 0 • UseRSSForUDP = 0 Additionally, this option chooses the best processors to assign to: • DefaultRecvRingProcessor • TxInterruptProcessor • TxForwardingProcessor • In Operating Systems support NDIS6.2: RssBaseProcNumber MaxRssProcessors • In Operating Systems support NDIS6.3: NumRSSQueues RssMaxProcNumber

Table 29 - Performance Tuning Tool Application Options

Flag	Description
-m	Manual configuration This option must be followed by one connection name. This option assigns the provided base and number of CPUs to: • *RssBaseProcNumber • *MaxRssProcessors Additionally, this option assigns the following with processors inside the range: • DefaultRecvRingProcessor • TxInterruptProcessor
-r	Restore default settings. This option can be followed by one or two connection names. This option automatically sets the driver registry values back to their default values: • SendCompletionMethod = 0 - IPoIB; 1 - ETH • RecvCompletionMethod = 2 • *ReceiveBuffers = 1024 • UseRSSForRawIP = 1 • DefaultRecvRingProcessor = -1 • TxInterruptProcessor = -1 • TxForwardingProcessor = -1 • UseRSSForUDP = 1 • In Operating Systems support NDIS6.2: MaxRssProcessors = 8 • In Operating Systems support NDIS6.3: NumRSSQueues = 8
-c1	Specifies first connection name. See examples
-c2	Specifies second connection name. See examples
-b	Specifies base RSS processor number. See examples. Used for manual option (-m) only.
-n	Specifies number of RSS processors. See examples. Used for manual option (-m) only.

Table 29 - Performance Tuning Tool Application Options

Flag	Description
-st	Single stream traffic scenario. This option must be followed by one or two connection names for an Ethernet adapter. The tuning will restore the default settings on the second connection and performed on the first connection. This option automatically sets: • SendCompletionMethod = 0 • RecvCompletionMethod = 2 • *ReceiveBuffers = 1024 • In Operating Systems support NDIS6.3: RssProfile = 4 • Additionally, this option chooses the best processors to assign to: • DefaultRecvRingProcessor • TxInterruptProcessor • TxForwardingProcessor • In Operating Systems support NDIS6.2: RssBaseProcNumber MaxRssProcessors • In Operating Systems support NDIS6.3: NumRSSQueues RssMaxProcNumber

Examples

For example, if the adapter is represented by "Local Area Connection 6" and "Local Area Connection 7"

```
For single port stream tuning type:
perf_tuning.exe -s -c1 "Local Area Connection 6" -c2 "Local Area Connection
7"
or to set one adapter only:
perf_tuning.exe -s -c1 "Local Area Connection 6"
For single stream tuning type:
perf_tuning.exe -st -c1 "Local Area Connection 6" -c2 "Local Area Connection
7"
or to set one adapter only:
perf_tuning.exe -st -c1 "Local Area Connection 6"
For dual port streams tuning type:
perf_tuning.exe -d -c1 "Local Area Connection 6" -c2 "Local Area Connection
7"
For forwarding streams tuning type:
perf_tuning.exe -f -c1 "Local Area Connection 6" -c2 "Local Area Connection
7"
```

```
For manual tuning of the first adapter to use RSS on CPUs 0-3:
perf_tuning.exe -m -c1 "Local Area Connection 6" -b 0 -n 4
In order to restore defaults type:
perf_tuning.exe -r -c1 "Local Area Connection 6" -c2 "Local Area Connection
7"
```

3.8.1.6 SR-IOV Tuning

To achieve best performance on SR-IOV VF, please run the following powershell commands on the host:

```
Set-VMNetworkAdapter -Name "Network Adapter" -VMName vm1 -IovQueuePairsRequested 4
OR
Set-VMNetworkAdapter -Name "Network Adapter" -VMName vm1 -IovQueuePairsRequested 8
for 40GbE
```

3.8.1.7 Improving Live Migration

In order to improve live migration over SMB direct performance, please set the following registry key to 0 and reboot the machine:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\LanmanServer\Parameters\RequireSe-
curitySignature
```

3.8.2 Application Specific Optimization and Tuning

3.8.2.1 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:***

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

Step 2. Open "Network Adapters".

Step 3. Right click the relevant Ethernet adapter and select Properties.

Step 4. Select the "Advanced" tab

Step 5. Modify performance parameters (properties) as desired.

3.8.2.1.1 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.

3.8.2.2 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 MLNX_VPI_WinOF Registry Keys following the path below:

http://www.mellanox.com/page/products_dyn?product_family=32&mtag=windows_sw_drivers

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

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

Step 2. Open "Network Adapters".

Step 3. Right click the relevant IPoIB adapter and select Properties.

Step 4. Select the "Advanced" tab

Step 5. Modify performance parameters (properties) as desired

3.8.3 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 it allows the OS to coalesce many small messages into a large one.

  - Valid MTU values range for an Ethernet driver is between 614 and 9614.
  - Valid MTU values range for an IPoIB driver is between 1500 and 4092.

  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).
- **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).

3.8.4 Adapter Proprietary Performance Counters

Proprietary Performance Counters are used to provide information on Operating System, application, service or the drivers' performance. Counters can be used for different system debugging purposes, help to determine system bottlenecks and fine-tune system and application performance. The Operating System, network, and devices provide counter data that the application can consume to provide users with a graphical view of the system's performance quality.

> All adapter cards related counters are cleared/reset only upon the card's restart or machine power-cycle.

WinOF counters hold the standard Windows CounterSet API that includes:

- Network Interface
- RDMA activity
- SMB Direct Connection

3.8.4.1 Supported Standard Performance Counters

3.8.4.1.1 Proprietary Mellanox Adapter Traffic Counters

Proprietary Mellanox adapter traffic counter set consists of global traffic statistics which gather information from ConnectX®-3 and ConnectX®-3 Pro network adapters, and includes traffic statistics, and various types of error and indications from both the Physical Function and Virtual Function.

Table 30 - Mellanox Adapter Traffic Counters

Mellanox Adapter Traffic Counters	Description
Bytes IN	
Bytes Received	Shows the number of bytes received by the adapter. The counted bytes include framing characters.
Bytes Received/Sec	Shows the rate at which bytes are received by the adapter. The counted bytes include framing characters.
Packets Received	Shows the number of packets received by ConnectX-3 and ConnectX-3Pro network interface.
Packets Received/Sec	Shows the rate at which packets are received by ConnectX-3 and ConnectX-3Pro network interface.
Bytes/ Packets OUT	
Bytes Sent	Shows the number of bytes sent by the adapter. The counted bytes include framing characters.

Table 30 - Mellanox Adapter Traffic Counters

Mellanox Adapter Traffic Counters	Description
Bytes Sent/Sec	Shows the rate at which bytes are sent by the adapter. The counted bytes include framing characters.
Packets Sent	Shows the number of packets sent by ConnectX-3 and ConnectX-3Pro network interface.
Packets Sent/Sec	Shows the rate at which packets are sent by ConnectX-3 and ConnectX-3Pro network interface.
Bytes' TOTAL	
Bytes Total	Shows the total of bytes handled by the adapter. The counted bytes include framing characters.
Bytes Total/Sec	Shows the total rate of bytes that are sent and received by the adapter. The counted bytes include framing characters.
Packets Total	Shows the total of packets handled by ConnectX-3 and ConnectX-3Pro network interface.
Packets Total/Sec	Shows the rate at which packets are sent and received by ConnectX-3 and ConnectX-3Pro network interface.
Control Packets	The total number of successfully received control frames
ERRORS, DROP, AND MISC. INDICATIONS	
Packets Outbound Errors[a]	Shows the number of outbound packets that could not be transmitted because of errors found in the physical layer.
Packets Outbound Discarded[a]	Shows the number of outbound packets to be discarded in the physical layer, even though no errors had been detected to prevent transmission. One possible reason for discarding packets could be to free up some buffer space.
Packets Received Errors[a]	Shows the number of inbound packets that contained errors in the physical layer, preventing them from being deliverable.
Packets Received with Frame Length Error	Shows the number of inbound packets that contained error where the frame has length error. Packets received with frame length error are a subset of packets received errors.
Packets Received with Symbol Error	Shows the number of inbound packets that contained symbol error or an invalid block. Packets received with symbol error are a subset of packets received errors.
Packets Received with Bad CRC Error	Shows the number of inbound packets that failed the CRC check. Packets received with bad CRC error are a subset of packets received errors.
Packets Received Discarded[a]	Shows the number of inbound packets that were chosen to be discarded in the physical layer, even though no errors had been detected to prevent their being deliverable. One possible reason for discarding such a packet could be a buffer overflow.

a. Those error/discard counters are related to layer-2 issues, such as CRC, length, and type errors. There is a possibility of an error/discard in the higher interface level. For example, a packet can be discarded for the lack of a receive buffer, or when there is no steering rule defined to receive it. To see the sum of all error/discard packetsTo see the sum of all error/discard packets, read the Windows Network-Interface Counters. Note that for IPoIB, the Mellanox counters are for IB layer-2 issues only, and Windows Network-Interface counters are for interface level issues.

3.8.4.1.2 Proprietary Mellanox Adapter Diagnostics Counters

Proprietary Mellanox adapter diagnostics counter set consists of the NIC diagnostics. These counters collect information from ConnectX®-3 and ConnectX®-3 Pro firmware flows.

Table 31 - Mellanox Adapter Diagnostics Counters

Mellanox Adapter Diagnostics Counters	Description
Requester length errors	Number of local length errors when the local machine generates outbound traffic.
Responder length errors	Number of local length errors when the local machine receives inbound traffic.
Requester QP operation errors	Number of local QP operation errors when the local machine generates outbound traffic.
Responder QP operation errors	Number of local QP operation errors when the local machine receives inbound traffic.
Requester protection errors	Number of local protection errors when the local machine generates outbound traffic.
Responder protection errors	Number of local protection errors when the local machine receives inbound traffic.
Requester CQE errors	Number of local CQE with errors when the local machine generates outbound traffic.
Responder CQE errors	Number of local CQE with errors when the local machine receives inbound traffic. **Note:** RDMA receivers need to post receive WQEs to handle incoming messages. If the application does not know how many messages are expected to be received (e.g. by maintaining high level message credits), they may post more receive WQEs than what will actually be used. On application teardown, if the application did not use up all of its received WQEs, the device will issue completion with error for these WQEs to indicate HW does not plan to use them. This is done with a clear syndrome indication of “Flushed with error”.
Requester Invalid request errors	Number of remote invalid request errors when the local machine generates outbound traffic, i.e. NAK was received indicating that the other end detected invalid OpCode request.
Responder Invalid request errors	Number of remote invalid request errors when the local machine receives inbound traffic.

Table 31 - Mellanox Adapter Diagnostics Counters

Mellanox Adapter Diagnostics Counters	Description
Requester Remote access errors	Number of remote access errors when the local machine generates outbound traffic, i.e. NAK was received indicating that the other end detected wrong rkey.
Responder Remote access errors	Number of remote access errors when the local machine receives inbound traffic, i.e. the local machine received RDMA request with wrong rkey.
Requester RNR NAK	Number of RNR (Receiver Not Ready) NAKs received when the local machine generates outbound traffic.
Responder RNR NAK	Number of RNR (Receiver Not Ready) NAKs sent when the local machine receives inbound traffic.
Requester out of order sequence NAK	Number of Out of Sequence NAK received when the local machine generates outbound traffic, i.e. the number of times the local machine received NAKs indicating OOS on the receiving side.
Responder out of order sequence received	Number of Out of Sequence packet received when the local machine receives inbound traffic, i.e. the number of times the local machine received messages that are not consecutive.
Requester resync	Number of resync operations when the local machine generates outbound traffic.
Responder resync	Number of resync operations when the local machine receives inbound traffic.
Requester Remote operation errors	Number of remote operation errors when the local machine generates outbound traffic, i.e. NAK was received indicating that the other end encountered an error that prevented it from completing the request.
Requester transport retries exceeded errors	Number of transport retries exceeded errors when the local machine generates outbound traffic.
Requester RNR NAK retries exceeded errors	Number of RNR (Receiver Not Ready) NAKs retries exceeded errors when the local machine generates outbound traffic.
Bad multicast received	Number of bad multicast packet received.
Discarded UD packets	Number of UD packets silently discarded on the receive queue due to lack of receives descriptor.
Discarded UC packets	Number of UC packets silently discarded on the receive queue due to lack of receives descriptor.
CQ overflows	Number of CQ overflows. **NOTE:** this value is evaluated for the entire NIC since there are cases where CQ might be associated with both ports (i.e. the value on all ports is identical).

Table 31 - Mellanox Adapter Diagnostics Counters

Mellanox Adapter Diagnostics Counters	Description
EQ overflows	Number of EQ overflows. **NOTE:** this value is evaluated for the entire NIC since there are cases where EQ might be associated with both ports (i.e. the value on all ports is identical).
Bad doorbells	Number of bad DoorBells
Responder duplicate request received (pending firmware implementation)	Number of duplicate requests received when the local machine receives inbound traffic.
Requester time out received (pending firmware implementation)	Number of time out received when the local machine generates outbound traffic.
Device detected stalled state	Number of times the device has entered the stalled state (per port).
Packet detected as stalled	Number of events where device was stalled for longer than the watermark.
Link down events	Number of times that the link operative state changes to down.

3.8.4.1.3 Proprietary Mellanox QoS Counters

Proprietary Mellanox QoS counter set consists of flow statistics per (VLAN) priority. Each QoS policy is associated with a priority. The counter presents the priority's traffic, pause statistic.

Table 32 - Mellanox Qos Counters

Mellanox Qos Counters	Description
Bytes/Packets IN	
Bytes Received	The number of bytes received that are covered by this priority. The counted bytes include framing characters (modulo 2^64).
Bytes Received/Sec	The number of bytes received per second that are covered by this priority. The counted bytes include framing characters.
Packets Received	The number of packets received that are covered by this priority (modulo 2^64).
Packets Received/Sec	The number of packets received per second that are covered by this priority.
Bytes/Packets OUT	
Bytes Sent	The number of bytes sent that are covered by this priority. The counted bytes include framing characters (modulo 2^64).

Table 32 - Mellanox Qos Counters

Mellanox Qos Counters	Description
Bytes Sent/Sec	The number of bytes sent per second that are covered by this priority. The counted bytes include framing characters.
Packets Sent	The number of packets sent that are covered by this priority (modulo 2^64).
Packets Sent/Sec	The number of packets sent per second that are covered by this priority.
Bytes and Packets Total	
Bytes Total	The total number of bytes that are covered by this priority. The counted bytes include framing characters (modulo 2^64).
Bytes Total/Sec	The total number of bytes per second that are covered by this priority. The counted bytes include framing characters.
Packets Total	The total number of packets that are covered by this priority (modulo 2^64).
Packets Total/Sec	The total number of packets per second that are covered by this priority.
PAUSE INDICATION	
Sent Pause Frames	The total number of pause frames sent from this priority to the far-end port. The untagged instance indicates the number of global pause frames that were sent.
Sent Pause Duration	The total duration of packets transmission being paused on this priority in microseconds.
Received Pause Frames	The number of pause frames that were received to this priority from the far-end port. The untagged instance indicates the number of global pause frames that were received.
Received Pause Duration	The total duration that far-end port was requested to pause for the transmission of packets in microseconds.
Sent Discard Frames	The number of packets discarded by the transmitter. **Note**: this counter is per TC and not per priority.

3.8.4.1.4 Propriety Mellanox Adapter RSS Counters

The Proprietary Adapter Mellanox RSS counter set consists of software counters per vport (in case of native RSS it is the physical port) and per CPU. These counters collect information about the RSS and non-RSS traffic received on a specific vport (or physical port for native RSS) and on a specific CPU. The counter set also defines a "Total" instance to collect on all CPUs for each available vport.

Table 33 - Mellanox Adapter RSS Counters

Mellanox Adapter RSS Counters	Description
Encapsulated NonRss IPv4 Only	Number of encapsulated Ipv4Only packets that have no RSS hash calculated by HW
Encapsulated NonRss IPv4/ Tcp	Number of encapsulated Ipv4Tcp packets that have no RSS hash calculated by HW
Encapsulated NonRss IPv4/ Udp	Number of encapsulated Ipv4Udp packets that have no RSS hash calculated by HW
Encapsulated NonRss IPv6 Only	Number of encapsulated Ipv6Only packets that have no RSS hash calculated by HW
Encapsulated NonRss IPv6/ Tcp	Number of encapsulated Ipv6Tcp packets that have no RSS hash calculated by HW
Encapsulated NonRss IPv6/ Udp	Number of encapsulated Ipv6Udp packets that have no RSS hash calculated by HW
Encapsulated NonRss Misc	Number of encapsulated packets that have no RSS hash calculated by HW without clear reason for that
Encapsulated Rss IPv4 Only	Number of received encapsulated packets that have RSS hash calculated on IPv4 header only
Encapsulated Rss IPv4/Tcp	Number of received encapsulated packets that have RSS hash calculated on IPv4 and Tcp headers
Encapsulated Rss IPv4/Udp	Number of received encapsulated packets that have RSS hash calculated on IPv4 and Udp headers
Encapsulated Rss IPv6 Only	Number of received encapsulated packets that have RSS hash calculated on IPv6 header only
Encapsulated Rss IPv6/Tcp	Number of received encapsulated packets that have RSS hash calculated on IPv6 and Tcp headers
Encapsulated Rss IPv6/Udp	Number of received encapsulated packets that have RSS hash calculated on IPv6 and Udp headers
Encapsulated Rss Misc	Number of received encapsulated packets that have RSS hash calculated with unknown RSS hash type
NonRss IPv4 Only	Number of Ipv4only packets that have no RSS hash calculated by HW
NonRss IPv4/Tcp	Number of Ipv4Tcp packets that have no RSS hash calculated by HW
NonRss IPv4/Udp	Number of Ipv4Udp packets that have no RSS hash calculated by HW

Table 33 - Mellanox Adapter RSS Counters

Mellanox Adapter RSS Counters	Description
NonRss IPv6 Only	Number of Ipv6Only packets that have no RSS hash calculated by H
NonRss IPv6/Tcp	Number of Ipv6Tcp packets that have no RSS hash calculated by HW
NonRss IPv6/Udp	Number of Ipv6Udp packets that have no RSS hash calculated by HW
NonRss Misc	Number of packets that have no RSS hash calculated by HW without clear reason for that
Rss IPv4 Only	Number of received packets that have RSS hash calculated on IPv4 header only
Rss IPv4/Tcp	Number of received packets that have RSS hash calculated on IPv4 and Tcp headers
Rss IPv4/Udp	Number of received packets that have RSS hash calculated on IPv4 and Udp headers
Rss IPv6 Only	Number of received packets that have RSS hash calculated on IPv6 header only
Rss IPv6/Tcp	Number of received packets that have RSS hash calculated on IPv6 and Tcp headers
Rss IPv6/Udp	Number of received packets that have RSS hash calculated on IPv6 and Udp headers
Rss Misc	Number of received packets that have RSS hash calculated with unknown RSS hash type

3.8.4.1.4.1Controlling the Counters

As of Version 5.30, the Adapter RSS Counter set is disabled by default. In order to enable it, the following registry value must be set to "1":

```
HKLM\SYSTEM\CurrentControlSet\Control\Class\{4d36e972-e325-11ce-bfc1-
08002be10318}\<nn>\RssMonitoringEnabled
```

Since these are software counters, and may have impact on the performance, they are activated only when the user has explicitly added the desired counters instance in Perfmon (or any other PCW application).

If "always on" behavior is desired (start counting upon driver load), the following registry key must be set to "1":

```
HKLM\SYSTEM\CurrentControlSet\Control\Class\{4d36e972-e325-11ce-bfc1-
08002be10318}\<nn>\RssCountersActivatedAtStartup
```

Since the Adapter RSS Counter set consists of software counters that are collected by the driver, unlike other adapter counters, they are not persistent and are cleared/reset upon driver restart.

3.8.4.1.5 Propriety RDMA Activity

Proprietary RDMA Activity counter set consists of NDK and NDSPI performance counters. These performance counters allow you to track Network Direct Kernel (RDMA) activity, including traffic rates, errors, and control plane activity.

Table 34 - RDMA Activity

RDMA Activity Counters	Description
RDMA Accepted Connections[a]	The number of inbound RDMA connections established.
RDMA Active Connections[a]	The number of active RDMA connections.
RDMA Completion Queue Errors[a]	This counter is not supported, and always is set to zero.
RDMA Connection Errors[a]	The number of established connections with an error before a consumer disconnected the connection.
RDMA Failed Connection Attempts[a]	The number of inbound and outbound RDMA connection attempts that failed.
RDMA Inbound Bytes/sec	The number of bytes for all incoming RDMA traffic. This includes additional layer two protocol overhead.
RDMA Inbound Frames/sec	The number, in frames, of layer two frames that carry incoming RDMA traffic.
RDMA Initiated Connections[a]	The number of outbound connections established.
RDMA Outbound Bytes/sec	The number of bytes for all outgoing RDMA traffic. This includes additional layer two protocol overhead.
RDMA Outbound Frames/sec	The number, in frames, of layer two frames that carry outgoing RDMA traffic.

[a] These counters are only implemented in NDK and are **not** implemented in NDSPI.

3.8.5 Device Proprietary Counters

Device propriety counters are per device and not per port.

These counters are intended for advanced debug of performance issues and may be used by Mellanox support to identify root cause in such cases. They do not necessarily indicate the existence of a problem but are often useful as additional information in the debug of performance issues.

Table 35 - Device Proprietary Counters

Name	Description
PCI Back-pressure/sec	Device core clocks without PCIe read/write credits. This value will be larger if the Host's ability to receive data from the NIC is lower. Possible causes: the memory accessed is not cached or aligned properly, or CPU frequency is low or throttled by power management.
No-WQE drops/sec	The amount of packet drops due to no available receive buffers in the host. This counter indicates that the NIC hardware was not able to post received data to the host due to lack of buffers. Possible causes: Slow or overloaded CPU cores. Possible fixes: Increase the number of receive buffers in the driver's advanced properties tab.
Scatter Back-pressure/sec	Device core clocks where Scatter delays Rx packet processing. Supported only on ConnectX3-Pro.
WQE fetch/Atomic Back-pressure/sec	Device core clocks where Work-Queue-Element fetch or Atomic operation delay Rx packet processing. Supported only on ConnectX3-Pro.
Steering/QPC Back-pressure/sec	Device core clocks where packet steering or queue-context handling delay Rx packet processing. Supported only on ConnectX3-Pro.
SQ Miss/sec	Transmit-queue/Requestor-QP context cache miss.
RQ Miss/sec	Receive-queue/Responder-QP context cache miss.
CQ Miss/sec	Completion-Queue (CQ) context cache miss.
EQ Miss/sec	Event-Queue (EQ) context cache miss.
MTT Miss/sec	Address translation page table (MTT) cache miss.
MPT Miss/sec	Address translation region table (MPT) cache miss.
External Blueflame hit/sec	Latency critical work-queue-element (BlueFlame) read from NIC buffer.
External Blueflame replace/sec	Latency critical work-queue-element (BlueFlame) swap out from NIC buffer.
External Doorbell push/sec	Amount of doorbells received.
External Doorbell drop/sec	Amount of doorbells dropped.

3.8.5.1 Mellanox Proprietary WinOF Bus Counters

This set of counters contains device’s low-level counters used for debugging and behavior analysis.

Table 36 - Mellanox Proprietary WinOF Bus Counters

Mellanox WinOF Bus Counters	Description
PCI Back-pressure/sec	Device core clocks without PCIe read/write credits
No-WQE Drops/sec	Amount of packet drops due to no available receive buffers in the host.
Scatter Back-pressure/sec	Device core clocks where Scatter delays Rx packet processing. Supported only on Connectx3-Pro.
WQE fetch/Atomic Back-pressure/sec	Device core clocks where Work-Queue-Element fetch or Atomic operation, delay Rx packet processing. Supported only on Connectx3-Pro.
Steering/QPC Back-pressure/sec	Device core clocks where packet steering or queue-context handling delay Rx packet processing. Supported only on Connectx3-Pro.
Receive WQE cache hit/sec	Number of receive WQE cache lookups resulted in a hit.
Receive WQE cache lookup/sec	Number of receive WQE cache lookups.
SQ Miss/sec	Transmit-queue/Requestor-QP context cache miss.
RQ Miss/sec	Receive-queue/Responder-QP context cache miss.
CQ Miss/sec	Completion-Queue (CQ) context cache miss.
EQ Miss/sec	Event-Queue(EQ) context cache miss.
MTT Miss/sec	Address translation page table (MTT) cache miss.
MPT Miss/sec	Address translation region table (MPT) cache miss.
External Blueflame hit/sec	Latency critical work-queue-element (BlueFlame) read from NIC buffer.
External Blueflame Replace/sec	Latency critical work-queue-element (BlueFlame) swap out from NIC buffer.
External Doorbell Push/sec	Amount of doorbells received.
External Doorbell Drop/sec	Amount of doorbells dropped.
MPT entries used for QP	Number of Memory Protection Table (MPT) entries used for QPs.
MPT entries used for CQ	Number of Memory Protection Table (MPT) entries used for CQs.
MPT entries used for EQ	Number of Memory Protection Table (MPT) entries used for EQs.
MPT entries used for MR	Number of Memory Protection Table (MPT) entries used for MRs.

Table 36 - Mellanox Proprietary WinOF Bus Counters

Mellanox WinOF Bus Counters	Description
MTT entries used for QP	Number of Memory Translation Table (MTT) entries used for QPs.
MTT entries used for CQ	Number of Memory Translation Table (MTT) entries used for CQs.
MTT entries used for EQ	Number of Memory Translation Table (MTT) entries used for EQs.
MTT entries used for MR	Number of Memory Translation Table (MTT) entries used for MRs.
CPU MEM-pages (4K) mapped by TPT for QP	Total number of CPU memory pages (4K) mapped by TPT for QPs.
CPU MEM-pages (4K) mapped by TPT for CQ	Total number of CPU memory pages (4K) mapped by TPT for CQs.
CPU MEM-pages (4K) mapped by TPT for EQ	Total number of CPU memory pages (4K) mapped by TPT for EQs.
CPU MEM-pages (4K) mapped by TPT for MR	Total number of CPU memory pages (4K) mapped by TPT for MRs.

3.9 System Recovery upon Error Detection

Upon error detection, WinOF can initiate reset in order to recover from the error automatically.

WinOF differentiates between two types of resets:

- Software reset: upon error detection, WinOF automatically closes and re-opens all NDIS resources. No HCA reset is performed
- Hardware reset: HCA is reset, all driver resources (NDK and NDIS) automatically close and re-open.

WinOF handles the reset flow as follows:

Table 37 - RDMA Activity

Configuration	IPoIB/ RoCE	Native Ethernet	HyperV with VMQ	SR-IOV VF over HyperV	SR-IOV VF over KVM	SR-IOV Host Machine (PF)
Reset type	Software reset	Software reset	No operation (silent success)	Software reset	Software reset	No operation (silent success)

For example, in the configuration of HyperV with VMQ, in case of an error detection, no action will be taken.

3.10 NIC Resiliency

NIC may unexpectedly hang due to failures in either one of the hardware, firmware or software. In these cases, the problematic device should be isolated in order to prevent the non-responsive NIC from back-pressuring the entire cluster. In addition to isolating the device, this feature helps maintaining the ability to recover when exiting the hang state.

For information about the relevant registry keys for this feature, please refer to Section 3.6.9.3, "NIC Resiliency Registry Keys", on page 152

3.11 RDMA Configuration

3.11.1 Memory Translation Table (MTT) Optimization

For RDMA communication, the adapter performs virtual-to-physical memory address translation internally. This translation is defined by a Memory Translation Table (MTT) that is built internally by the WinOF device drivers.

The adapter contains internal cache for the address translations. The utilization of this cache can be improved by reducing the size of the MTT. This is possible when memory buffers that are mapped to the adapter are contiguous in the physical address space and fulfill some address alignment constraints. To facilitate this optimization, buffer allocations that are internal to the WinOF drivers (such as QP and CQ buffers) are first attempted such that they satisfy these constraints. If the Operating System (OS) cannot satisfy this allocation, the allocation is attempted again with no constraints.

The physical contiguity and alignment are more demanding on the OS. Therefore, the optimization can be turned OFF (thus reducing the demand on the OS) by setting the following registry key to zero:

- Key: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ndfltr\Parameters`
- Value Name: `ContiguousMemPoolBucketSize`
- Value Type: `REG_DWORD`
- Data: `0`

4 Utilities

4.1 Snapshot Tool

The snapshot tool scans the machine and provide information on the current settings of the operating system, networking and hardware.

It is highly recommended to add this report when you contact the support team

4.1.1 Snapshot Usage

The snapshot tool can be found at:

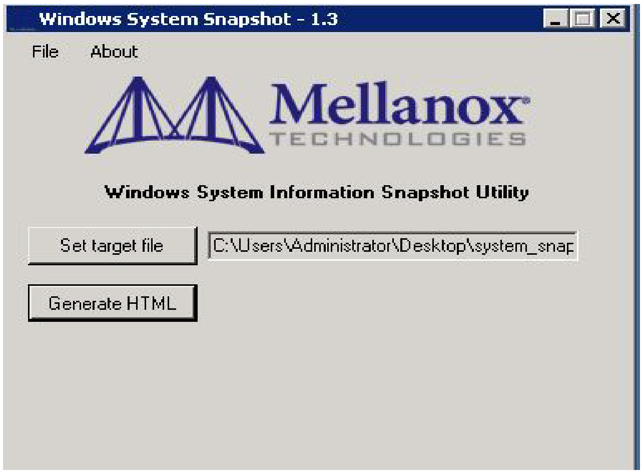
```
<installation_directory>\tools\MLNX_System_Snapshot.exe
```

The user can set the report location.

➢ ***To generate the snapshot report:***

Step 1. [Optional] Change the location of the generated file by setting the full path of the file to be generated or by pressing “Set target file” and choosing the directory that will hold the generated file and its file name.

Step 2. Click on Generate HTML button



Windows System Snapshot - 1.3
File
About
Mellanox
TECHNOLOGIES
Windows System Information Snapshot Utility
Set target file
Generate HTML

C:\Users\Administrator\Desktop\system_snap

Once the report is ready the folder which contains the report will be opened automatically.

4.2 part_man - Virtual IPoIB Port Creation Utility

part_man is used to add/remove/show virtual IPoIB ports. Each Mellanox IPoIB port can have multiple virtual IPoIB ports, which can use the default PKey value (0xffff) or a non-default value supplied by the user.

➢ ***Usage***

```
part_man.exe [-v] <add|rem> <network connection name> [iname] [pkey]
part_man.exe [-v] <show|remall>
part_man.exe -help
```

Table 38 - Virtual IPoIB Port Creation Utility

Options	Description
add	Add a virtual adapter.
rem	Remove a virtual adapter. When using the rem command, provide the connection name of the newly created virtual adapter. You may also specify the iname and pkey, if needed to disambiguate. All are provided by part_man show.
remall	Removal all virtual adapters.
show	Show the existing virtual adapters.
-help	Provide help text.
-v	Increases the verbosity level.
-h	Provides a help text.
network connection name	The name of a local area connection, as in Network Connections in Control Panel. For example: "Local Area Connection 2" (quotes are necessary around the name only if it contains a space).
iname	Any printable name without ':', ',', ';' '-' and ' ' and starting with an 'i'. If no iname is specified for an "add" command, one will be auto-generated by the tool. This parameter, which was previously mandatory, is now optional for these commands.
pkey	a 4 hex-digit value. It can be specified if a non-default pkey should be used.

When using the add/rem commands, only one virtual adapter can be added or removed in a single operation.

➢ ***Example***

Adding and removing a virtual adapter using defaults:

```
> part_man add "Ethernet 4" ipoib_4_1
Done...
> part_man show
Ethernet 6 ipoib_4_1 FFFF
> part_man rem "Ethernet 6"  ipoib_4_1
Done
```

Adding and removing a virtual adapter using non-defaults:

```
> part_man add "Ethernet 5" ipoib_5_1 F123
Done...
> part_man show
Ethernet 7    ipoib_5_1    F123
> part_man rem "Ethernet 7" ipoib_5_1 F123
Or simply...
part_man rem "Ethernet 7"
```

Adding a partial membership PKey value with the upper bit turned off:

```
part_man add "Ethernet 5" 7123
```

The new port will use the partial PKey only in the absence of a full membership PKey of the same value (0xf123 for the example above) in the OpenSM configuration. Otherwise the full membership PKey will be chosen.

> Make sure that the PKeys used in the part_man commands are supported by the OpenSM running on this port and the membership type of them is consistent with the one defined by OpenSM. If the PKeys are not supported, the new vIPoIB port will stay in a disconnected state until the configuration is fixed.

For further details about partitions configurations for OpenSM, please refer to the section titled "Partitions" in *Mellanox OFED for Linux User Manual.*

For further details about pre and post configurations for the new vIPoIB port, please refer to 3.2.7 "Multiple Interfaces over non-default PKeys Support," on page 86

> The part_man tool allows the creation of up to 64 vIPoIB interfaces (32 per port).

4.3 vea_man- Virtual Ethernet

vea_man is a tool that consists of a set of commands that allow you to add or remove a VEA, or query the existing Mellanox Ethernet adapters and see which are virtual and which are physical.

> vea_man is not supported in Windows 2016 and above.

4.3.1 Adding a New Virtual Adapter

➤ ***To add a new virtual adapter, run the following command:***

```
> vea_man -a <adapter name>
```

<adapter name> is the name of the existing physical adapter which will be, essentially, cloned. The new adapter will be named by system default rules.

4.3.2 Removing a Virtual Ethernet Adapter

➢ ***To remove a virtual ethernet adapter, run the following command:***

```
> vea_man -r <adapter name>
```

4.3.3 Querying the Virtual Ethernet Database

Querying the virtual ethernet database reports all physical and virtual ethernet adapters on all Mellanox cards in the system.

➢ ***To query the virtual ethernet database, run the following command:***

```
> vea_man -q
> vea_man
```

4.3.4 Help Message

➢ ***To view the help message, run the following command:***

```
> vea_man -?
> vea_man -h
```

If your adapter name has spaces in it, you need to surround it with quotes.

Examples:

> vea_man -a "Ethernet 9"   - Adds a new adapter as a virtual duplicate of Ethernet 9
> vea_man -r "Ethernet 13"   - Removes virtual ethernet adapter Ethernet 13

4.4 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.

4.4.1 Utilities Usage: Common Configuration, Interface and Addressing

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.

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 a 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

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>'

Define the environment variable IBDIAG_DEV_IDX

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.

For further information on the following tools, please refer to the tool's man page.

Table 39 - Diagnostic Utilities

Utility	Description
ibdiagnet	Scans the fabric using directed route packets and extracts all the available information regarding its connectivity and devices. It's only supported in Windows Server 2012 and above, or Windows Client 8.1 and above.
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 switch 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
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.
ibdump	Dumps InfiniBand, Ethernet and all RoCE versions' traffic that flows to and from Mellanox ConnectX®-3/ConnectX®-3 Pro NIC's ports. It provides a similar functionality to the tcpdump tool on a 'standard' Ethernet port. The ibdump tool generates packet dump file in .pcap format. This file can be loaded by the Wireshark tool (www.wireshark.org) for graphical traffic analysis. This provides the ability to analyze network behavior and performance, and to debug applications that send or receive RDMA network traffic. Run "ibdump -h" to display a help message which details the tools options.
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.
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.

Table 39 - Diagnostic Utilities

Utility	Description
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.
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.
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.
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
ibclearerrors	Clears the PMA error counters in PortCounters by either waking the InfiniBand subnet topology or using an already saved topology file.
ibstat	Displays basic information obtained from the local IB driver. Output includes LID, SMLID, port state, link width active, and port physical state.
vstat	Displays information on the HCA attributes.
osmtest	Validates 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.
ibaddr	Displays the lid (and range) as well as the GID address of the port specified (by DR path, lid, or GUID) or the local port by default.
ibcacheedit	Allows users to edit an ibnetdiscover cache created through the --cache option in ibnetdiscover(8)
iblinkinfo	Reports link info for each port in an IB fabric, node by node. Optionally, iblinkinfo can do partial scans and limit its output to parts of a fabric.
ibqueryerrors	Reports the port error counters which exceed a threshold for each port in the fabric. The default threshold is zero (0). Error fields can also be suppressed entirely. In addition to reporting errors on every port. ibqueryerrors can report the port transmit and receive data as well as report full link information to the remote port if available.

Table 39 - Diagnostic Utilities

Utility	Description
ibsysstat	Uses vendor MADs to validate connectivity between InfiniBand nodes and obtain other information about the InfiniBand node. ibsysstat is run as client/server. Default is to run as client.
saquery	Issues the selected SA query. Node records are queried by default.
smpdump	Gets SM attributes from a specified SMA. The result is dumped in hex by default.

4.5 Fabric Performance Utilities

The performance utilities described in this chapter are intended to be used as a performance micro-benchmark. They support both InfiniBand and RoCE. F

For further information on the following tools, please refer to the help text of the tool by running the --help command line parameter.

Table 40 - Fabric Performance Utilities

Utility	Description
nd_write_bw	This test is used for performance measuring of RDMA-Write requests in Microsoft Windows Operating Systems. nd_write_bw is performance oriented for RDMA-Write with maximum throughput, and runs over Microsoft's NetworkDirect standard. The level of customizing for the user is relatively high. User may choose to run with a customized message size, customized number of iterations, or alternatively, customized test duration time. nd_write_bw runs with all message sizes from 1B to 4MB (powers of 2), message inlining, CQ moderation.
nd_write_lat	This test is used for performance measuring of RDMA-Write requests in Microsoft Windows Operating Systems. nd_write_lat is performance oriented for RDMA-Write with minimum latency, and runs over Microsoft's NetworkDirect standard. The level of customizing for the user is relatively high. User may choose to run with a customized message size, customized number of iterations, or alternatively, customized test duration time. nd_write_lat runs with all message sizes from 1B to 4MB (powers of 2), message inlining, CQ moderation.
nd_read_bw	This test is used for performance measuring of RDMA-Read requests in Microsoft Windows Operating Systems. nd_read_bw is performance oriented for RDMA-Read with maximum throughput, and runs over Microsoft's NetworkDirect standard. The level of customizing for the user is relatively high. User may choose to run with a customized message size, customized number of iterations, or alternatively, customized test duration time. nd_read_bw runs with all message sizes from 1B to 4MB (powers of 2), message inlining, CQ moderation.

Table 40 - Fabric Performance Utilities

Utility	Description
nd_read_lat	This test is used for performance measuring of RDMA-Read requests in Microsoft Windows Operating Systems. nd_read_lat is performance oriented for RDMA-Read with minimum latency, and runs over Microsoft's NetworkDirect standard. The level of customizing for the user is relatively high. User may choose to run with a customized message size, customized number of iterations, or alternatively, customized test duration time. nd_read_lat runs with all message sizes from 1B to 4MB (powers of 2), message inlining, CQ moderation.
nd_send_bw	This test is used for performance measuring of Send requests in Microsoft Windows Operating Systems. nd_send_bw is performance oriented for Send with maximum throughput, and runs over Microsoft's NetworkDirect standard. The level of customizing for the user is relatively high. User may choose to run with a customized message size, customized number of iterations, or alternatively, customized test duration time. nd_send_bw runs with all message sizes from 1B to 4MB (powers of 2), message inlining, CQ moderation.
nd_send_lat	This test is used for performance measuring of Send requests in Microsoft Windows Operating Systems. nd_send_lat is performance oriented for Send with minimum latency, and runs over Microsoft's NetworkDirect standard. The level of customizing for the user is relatively high. User may choose to run with a customized message size, customized number of iterations, or alternatively, customized test duration time. nd_send_lat runs with all message sizes from 1B to 4MB (powers of 2), message inlining, CQ moderation.

4.6 mlxtool

mlxtool is a general utility used for debugging the driver using a command line.

➢ ***Usage***

```
mlxtool.exe <tool-name> <tool-arguments>
```

➢ ***Installation***

The tool is installed as part of the WinOF package, starting from WinOF v5.10. The package can be found in the following path:

C:\Program Files\Mellanox\MLNX_VPI\Tools

➢ ***Extracting the Tool from the Package***

In case the previous WinOF version is used (v5.10), the tool can be extracted from WinOF Rev 5.35 package.

In order to extract the tool, please follow the steps in Section 2.5, “Extracting Files Without Running Installation”, on page 31. Make sure to copy the tool and the following files to the same location: mlxtool.exe, complib.dll, mlxapi.dll.

4.6.1 mlxtool Help

mlxtool help has all the data required to operate all sub-tools and their arguments, and it is highly recommended to use it.

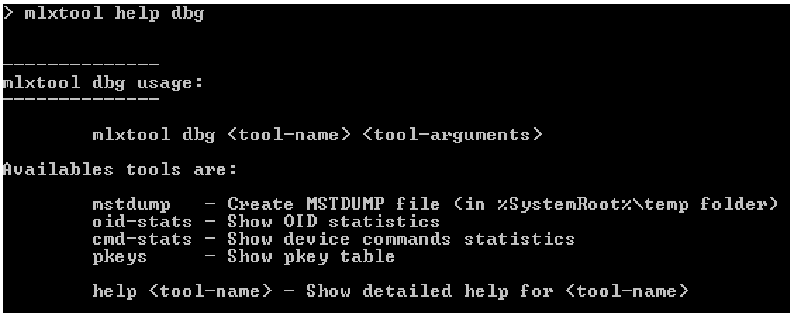
- `mlxtool` or `mlxtool help`

  Lists the general usage and the available sub-tools supported in this version.

- `mlxtool help <sub-tool>`

  Lists all the available arguments that can be used in a certain tool and provides a short description per argument.

Example:

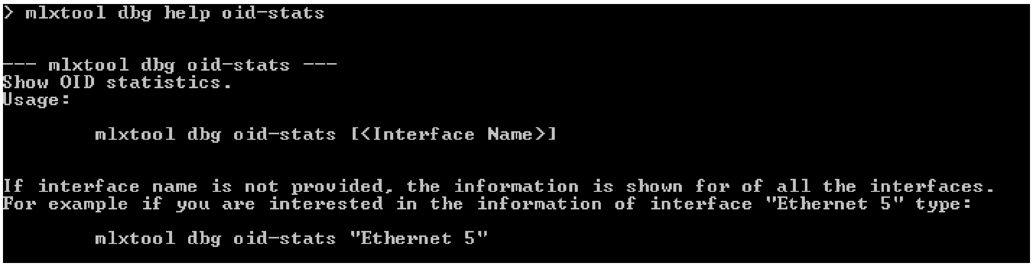


```
> mlxtool help dbg

--------------
mlxtool dbg usage:
--------------

        mlxtool dbg <tool-name> <tool-arguments>

Availables tools are:

        mstdump   - Create MSTDUMP file (in %SystemRoot%\temp folder)
        oid-stats - Show OID statistics
        cmd-stats - Show device commands statistics
        pkeys     - Show pkey table

        help <tool-name> - Show detailed help for <tool-name>
```

- `mlxtool <sub-tool> help <argument>`

  Shows details about the sub-tool and its arguments with a usage example.

  Example:



```
> mlxtool dbg help oid-stats

--- mlxtool dbg oid-stats ---
Show OID statistics.
Usage:

        mlxtool dbg oid-stats [<Interface Name>]

If interface name is not provided, the information is shown for of all the interfaces.
For example if you are interested in the information of interface "Ethernet 5" type:

        mlxtool dbg oid-stats "Ethernet 5"
```

4.6.2 dbg sub-tool

This sub-tool is used to extract debug information.

➢ ***Usage***

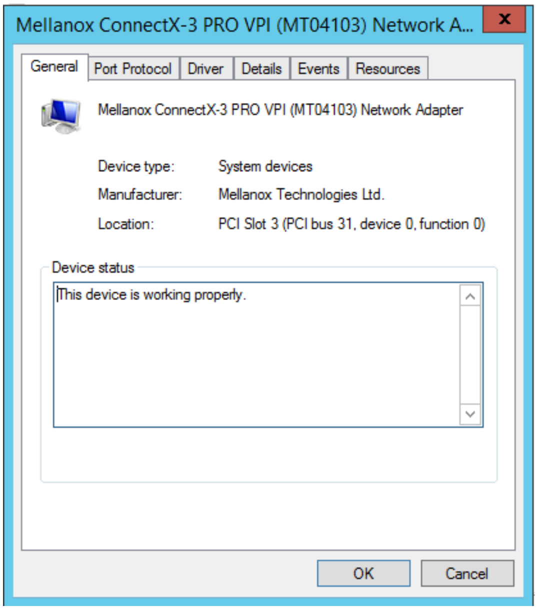
```
mlxtool.exe dbg <sub-tool> <arguments>
```

4.6.2.1 mstdump

This tool is used to create 6 mstdump files upon user request. For further information on the files created, you may refer to Table 49, “Events Causing Automatic State Dumps,” on page 207.

The parameters that should be used in following command are the PCI information found under “Location” as demonstrated in the image below:

```
<bus#> <device#> <function#>
```



Mellanox ConnectX-3 PRO VPI (MT04103) Network A...

Mellanox ConnectX-3 PRO VPI (MT04103) Network Adapter

Device type: System devices

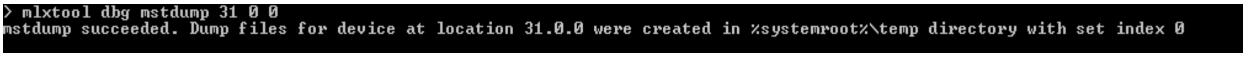
Manufacturer: Mellanox Technologies Ltd.

Location: PCI Slot 3 (PCI bus 31, device 0, function 0)

Device status

This device is working properly.

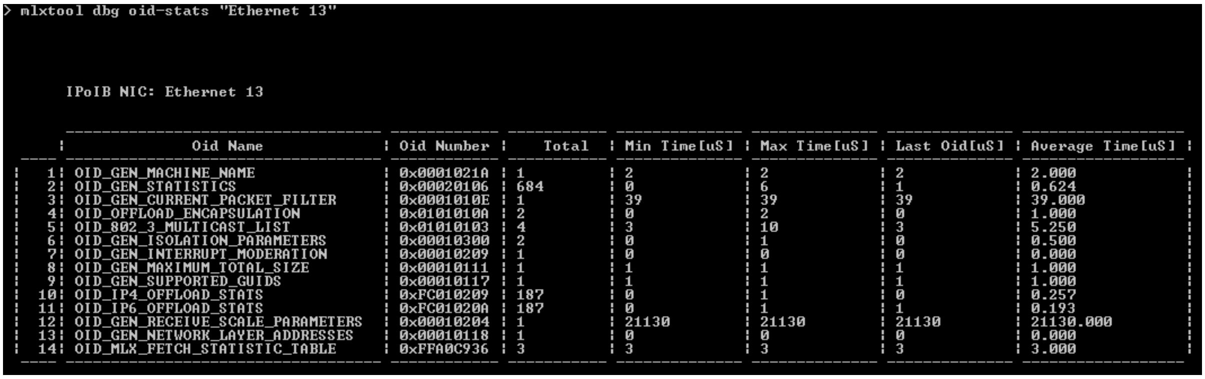
The output will indicate the files location and the index in the file name for this execution:



```
> mlxtool dbg mstdump 31 0 0
mstdump succeeded. Dump files for device at location 31.0.0 were created in %systemroot%\temp directory with set index 0
```

4.6.2.2 oid-stats

This tool displays the OIDs statistics from Ethernet and IPoIB drivers in microsecond:

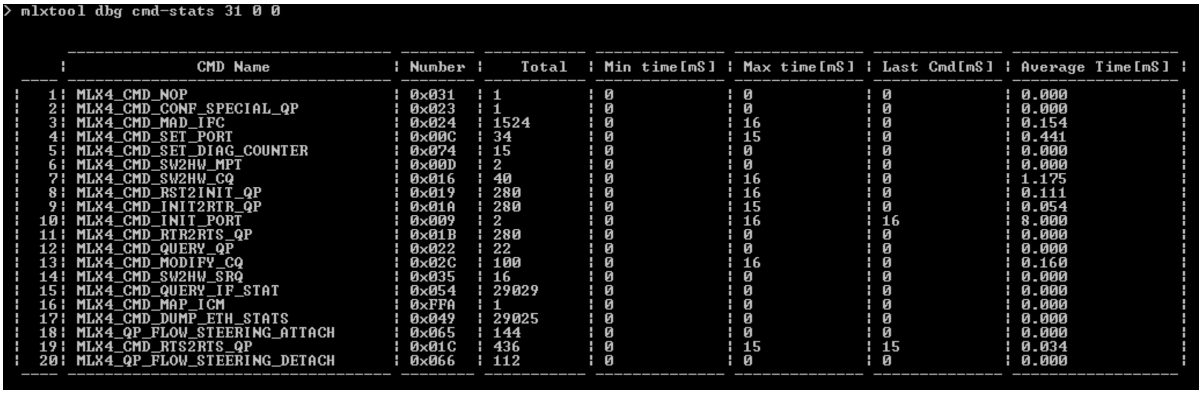


```
> mlxtool dbg oid-stats "Ethernet 13"



     IPoIB NIC: Ethernet 13

       ----------------------------------- -------------- ------------- ---------------- ---------------- ---------------- -------------------
      ¦              Oid Name              ¦ Oid Number ¦    Total    ¦ Min Time[uS] ¦ Max Time[uS] ¦ Last Oid[uS] ¦ Average Time[uS] ¦
 ---- ----------------------------------- -------------- ------------- ---------------- ---------------- ---------------- -------------------
¦   1¦ OID_GEN_MACHINE_NAME               ¦ 0x0001021A ¦ 1           ¦ 2              ¦ 2              ¦ 2              ¦ 2.000             ¦
¦   2¦ OID_GEN_STATISTICS                 ¦ 0x00020106 ¦ 684         ¦ 0              ¦ 6              ¦ 1              ¦ 0.624             ¦
¦   3¦ OID_GEN_CURRENT_PACKET_FILTER      ¦ 0x0001010E ¦ 1           ¦ 39             ¦ 39             ¦ 39             ¦ 39.000            ¦
¦   4¦ OID_OFFLOAD_ENCAPSULATION          ¦ 0x0101010A ¦ 2           ¦ 0              ¦ 2              ¦ 0              ¦ 1.000             ¦
¦   5¦ OID_802_3_MULTICAST_LIST           ¦ 0x01010103 ¦ 4           ¦ 3              ¦ 10             ¦ 3              ¦ 5.250             ¦
¦   6¦ OID_GEN_ISOLATION_PARAMETERS       ¦ 0x00010300 ¦ 2           ¦ 0              ¦ 1              ¦ 0              ¦ 0.500             ¦
¦   7¦ OID_GEN_INTERRUPT_MODERATION       ¦ 0x00010209 ¦ 1           ¦ 0              ¦ 0              ¦ 0              ¦ 0.000             ¦
¦   8¦ OID_GEN_MAXIMUM_TOTAL_SIZE         ¦ 0x00010111 ¦ 1           ¦ 1              ¦ 1              ¦ 1              ¦ 1.000             ¦
¦   9¦ OID_GEN_SUPPORTED_GUIDS            ¦ 0x00010117 ¦ 1           ¦ 1              ¦ 1              ¦ 1              ¦ 1.000             ¦
¦  10¦ OID_IP4_OFFLOAD_STATS              ¦ 0xFC010209 ¦ 187         ¦ 0              ¦ 1              ¦ 0              ¦ 0.257             ¦
¦  11¦ OID_IP6_OFFLOAD_STATS              ¦ 0xFC01020A ¦ 187         ¦ 0              ¦ 1              ¦ 1              ¦ 0.193             ¦
¦  12¦ OID_GEN_RECEIVE_SCALE_PARAMETERS   ¦ 0x00010204 ¦ 1           ¦ 21130          ¦ 21130          ¦ 21130          ¦ 21130.000         ¦
¦  13¦ OID_GEN_NETWORK_LAYER_ADDRESSES    ¦ 0x00010118 ¦ 1           ¦ 0              ¦ 0              ¦ 0              ¦ 0.000             ¦
¦  14¦ OID_MLX_FETCH_STATISTIC_TABLE      ¦ 0xFFA0C936 ¦ 3           ¦ 3              ¦ 3              ¦ 3              ¦ 3.000             ¦
 ---- ----------------------------------- -------------- ------------- ---------------- ---------------- ---------------- -------------------
```

In order to view the list of all valid ports, please refer to Section 4.6.3.1, “show port list”, on page 195

4.6.2.3 cmd-stats

This tool displays the device commands statistics in millisecond:



```
> mlxtool dbg cmd-stats 31 0 0


       ------------------------------------ --------- ------------- ---------------- ---------------- ---------------- -------------------
      ¦              CMD Name               ¦ Number ¦    Total    ¦ Min time[mS] ¦ Max time[mS] ¦ Last Cmd[mS] ¦ Average Time[mS] ¦
 ---- ------------------------------------ --------- ------------- ---------------- ---------------- ---------------- -------------------
¦   1¦ MLX4_CMD_NOP                        ¦ 0x031  ¦ 1           ¦ 0              ¦ 0              ¦ 0              ¦ 0.000             ¦
¦   2¦ MLX4_CMD_CONF_SPECIAL_QP            ¦ 0x023  ¦ 1           ¦ 0              ¦ 0              ¦ 0              ¦ 0.000             ¦
¦   3¦ MLX4_CMD_MAD_IFC                    ¦ 0x024  ¦ 1524        ¦ 0              ¦ 16             ¦ 0              ¦ 0.154             ¦
¦   4¦ MLX4_CMD_SET_PORT                   ¦ 0x00C  ¦ 34          ¦ 0              ¦ 15             ¦ 0              ¦ 0.441             ¦
¦   5¦ MLX4_CMD_SET_DIAG_COUNTER           ¦ 0x074  ¦ 15          ¦ 0              ¦ 0              ¦ 0              ¦ 0.000             ¦
¦   6¦ MLX4_CMD_SW2HW_MPT                  ¦ 0x00D  ¦ 2           ¦ 0              ¦ 0              ¦ 0              ¦ 0.000             ¦
¦   7¦ MLX4_CMD_SW2HW_CQ                   ¦ 0x016  ¦ 40          ¦ 0              ¦ 16             ¦ 0              ¦ 1.175             ¦
¦   8¦ MLX4_CMD_RST2INIT_QP                ¦ 0x019  ¦ 280         ¦ 0              ¦ 16             ¦ 0              ¦ 0.111             ¦
¦   9¦ MLX4_CMD_INIT2RTR_QP                ¦ 0x01A  ¦ 280         ¦ 0              ¦ 15             ¦ 0              ¦ 0.054             ¦
¦  10¦ MLX4_CMD_INIT_PORT                  ¦ 0x009  ¦ 2           ¦ 0              ¦ 16             ¦ 16             ¦ 8.000             ¦
¦  11¦ MLX4_CMD_RTR2RTS_QP                 ¦ 0x01B  ¦ 280         ¦ 0              ¦ 0              ¦ 0              ¦ 0.000             ¦
¦  12¦ MLX4_CMD_QUERY_QP                   ¦ 0x022  ¦ 22          ¦ 0              ¦ 0              ¦ 0              ¦ 0.000             ¦
¦  13¦ MLX4_CMD_MODIFY_CQ                  ¦ 0x02C  ¦ 100         ¦ 0              ¦ 16             ¦ 0              ¦ 0.160             ¦
¦  14¦ MLX4_CMD_SW2HW_SRQ                  ¦ 0x035  ¦ 16          ¦ 0              ¦ 0              ¦ 0              ¦ 0.000             ¦
¦  15¦ MLX4_CMD_QUERY_IF_STAT              ¦ 0x054  ¦ 29029       ¦ 0              ¦ 0              ¦ 0              ¦ 0.000             ¦
¦  16¦ MLX4_CMD_MAP_ICM                    ¦ 0xFFA  ¦ 1           ¦ 0              ¦ 0              ¦ 0              ¦ 0.000             ¦
¦  17¦ MLX4_CMD_DUMP_ETH_STATS             ¦ 0x049  ¦ 29025       ¦ 0              ¦ 0              ¦ 0              ¦ 0.000             ¦
¦  18¦ MLX4_QP_FLOW_STEERING_ATTACH        ¦ 0x065  ¦ 144         ¦ 0              ¦ 0              ¦ 0              ¦ 0.000             ¦
¦  19¦ MLX4_CMD_RTS2RTS_QP                 ¦ 0x01C  ¦ 436         ¦ 0              ¦ 15             ¦ 15             ¦ 0.034             ¦
¦  20¦ MLX4_QP_FLOW_STEERING_DETACH        ¦ 0x066  ¦ 112         ¦ 0              ¦ 0              ¦ 0              ¦ 0.000             ¦
 ---- ------------------------------------ --------- ------------- ---------------- ---------------- ---------------- -------------------
```

The parameters used in this command are:

```
<bus#> <device#> <function#>
```

For further details on these parameters, refer to Section 4.6.2.1, “mstdump”, on page 190.

In order to view the list of all devices, please refer to Section 4.6.3.2, “show device list”, on page 196.

4.6.2.4 pkeys

This tool displays the pkeys (indexes and values) available for each IPoIB interface.

Example:

If you wish to display the information of "Ethernet 5" interface, run the following command:

```
mlxtool dbg pkeys "Ethernet 5"
```

This command can be invoked on a specific IPoIB interface. If no interface name is provided, the information will be shown for all the interfaces.

```
ConnectX IPoIB NIC: Ethernet 7
              ---------------- ------------
             |   PKEY index   |    PKEY   |
              ---------------- ------------
             |       0        |    ffff   |
             |       1        |    f123   |
             |       2        |    9563   |
              ---------------- ------------
```

4.6.2.5 Res

This tool is used for pulling resource tracker information for VFs.

The parameters that should be used in the following command are detailed in the PCI information found under "Location", as demonstrated in the image below::

```
<bus#> <device#> <function#>
```

For further details on these parameters, please refer to Section 4.6.2.1, “mstdump”, on page 190.

In order to view the list of all devices, please refer to Section 4.6.3.2, “show device list”, on page 196.

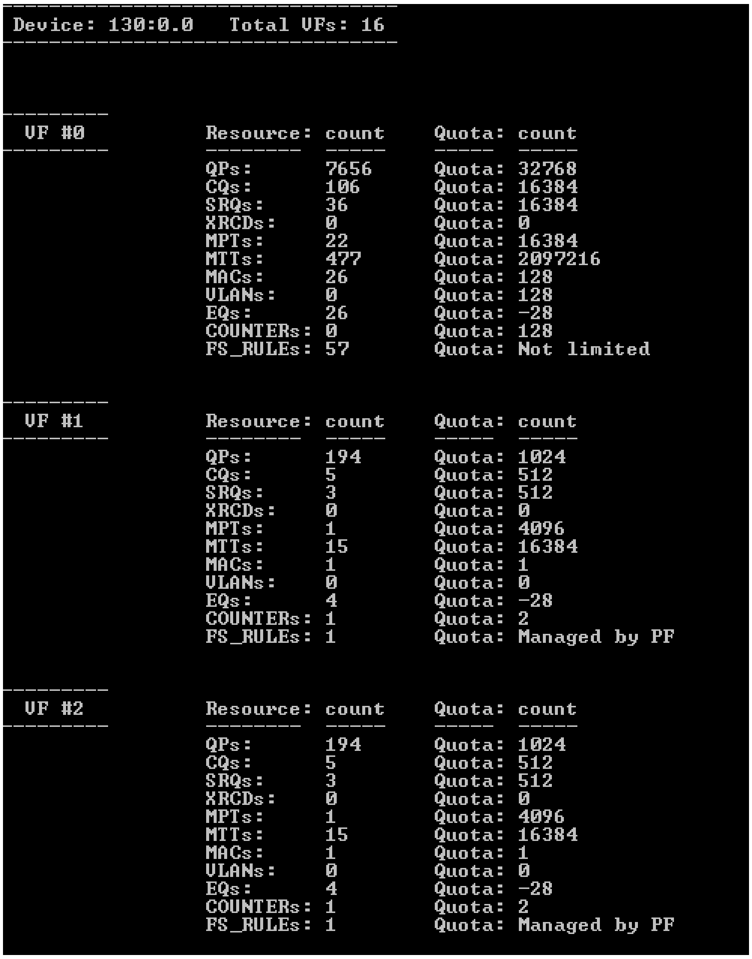
Usage:

```
mlxtool.exe dbg res [<pci-bus#> <pci-device#> <pci-function#>]
```

Example:

If "Location" is "PCI Slot 3 (PCI bus 130, device 0, function 0)", type:

```
mlxtool.exe dbg res 130 0 0
```



```
----------------------------------
 Device: 130:0.0   Total VFs: 16
----------------------------------



---------
  VF #0           Resource: count     Quota: count
---------         --------- -----     ------ -----
                  QPs:      7656      Quota: 32768
                  CQs:      106       Quota: 16384
                  SRQs:     36        Quota: 16384
                  XRCDs:    0         Quota: 0
                  MPTs:     22        Quota: 16384
                  MTTs:     477       Quota: 2097216
                  MACs:     26        Quota: 128
                  VLANs:    0         Quota: 128
                  EQs:      26        Quota: -28
                  COUNTERs: 0         Quota: 128
                  FS_RULEs: 57        Quota: Not limited


---------
  VF #1           Resource: count     Quota: count
---------         --------- -----     ------ -----
                  QPs:      194       Quota: 1024
                  CQs:      5         Quota: 512
                  SRQs:     3         Quota: 512
                  XRCDs:    0         Quota: 0
                  MPTs:     1         Quota: 4096
                  MTTs:     15        Quota: 16384
                  MACs:     1         Quota: 1
                  VLANs:    0         Quota: 0
                  EQs:      4         Quota: -28
                  COUNTERs: 1         Quota: 2
                  FS_RULEs: 1         Quota: Managed by PF


---------
  VF #2           Resource: count     Quota: count
---------         --------- -----     ------ -----
                  QPs:      194       Quota: 1024
                  CQs:      5         Quota: 512
                  SRQs:     3         Quota: 512
                  XRCDs:    0         Quota: 0
                  MPTs:     1         Quota: 4096
                  MTTs:     15        Quota: 16384
                  MACs:     1         Quota: 1
                  VLANs:    0         Quota: 0
                  EQs:      4         Quota: -28
                  COUNTERs: 1         Quota: 2
                  FS_RULEs: 1         Quota: Managed by PF
```

If PCI is not provided, the information is shown for of all the devices.

4.6.2.6 Ipoib-ep

This tool is used to show end points known to the ipoib driver.

Usage:

```
mlxtool.exe dbg ipoib-ep [<-l>][<-m>][<-r>][<Interface Name>]
```

Flags:

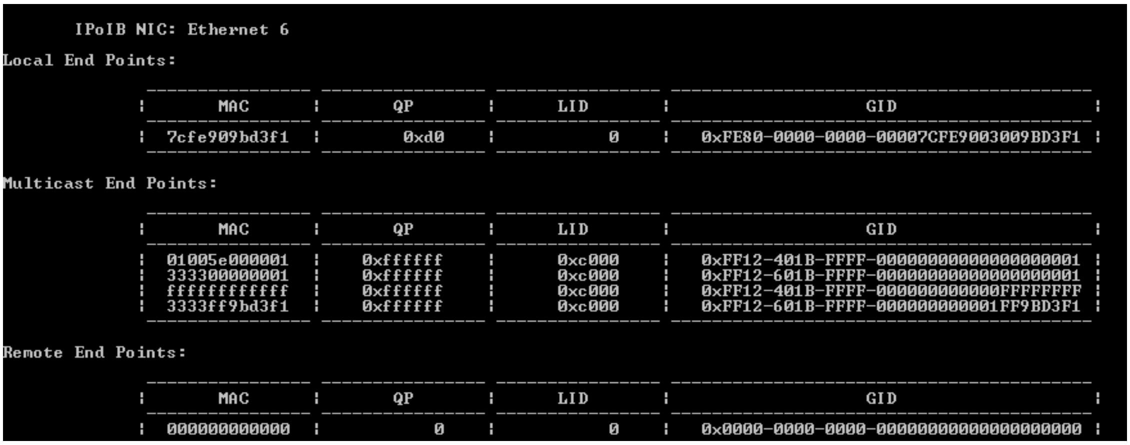
```
-l local end points
-m multicast end points
-r remote end points
```

There is no meaning to the order of the flags.

Example:

To show multicast and remote EP for interface "Ethernet 5", type:

```
mlxtool dbg ipoib-ep -m -r "Ethernet 5"
```



```
       IPoIB NIC: Ethernet 6
Local End Points:

              ---------------- ---------------- ---------------- ------------------------------------------
              |      MAC       |       QP       |      LID       |                   GID                    |
              ---------------- ---------------- ---------------- ------------------------------------------
              |  7cfe909bd3f1  |          0xd0  |             0  |  0xFE80-0000-0000-00007CFE9003009BD3F1  |
              ---------------- ---------------- ---------------- ------------------------------------------

Multicast End Points:

              ---------------- ---------------- ---------------- ------------------------------------------
              |      MAC       |       QP       |      LID       |                   GID                    |
              ---------------- ---------------- ---------------- ------------------------------------------
              |  01005e000001  |    0xffffff    |        0xc000  |  0xFF12-401B-FFFF-00000000000000000001  |
              |  333300000001  |    0xffffff    |        0xc000  |  0xFF12-601B-FFFF-00000000000000000001  |
              |  ffffffffffff  |    0xffffff    |        0xc000  |  0xFF12-401B-FFFF-000000000000FFFFFFFF  |
              |  3333ff9bd3f1  |    0xffffff    |        0xc000  |  0xFF12-601B-FFFF-000000000001FF9BD3F1  |
              ---------------- ---------------- ---------------- ------------------------------------------

Remote End Points:

              ---------------- ---------------- ---------------- ------------------------------------------
              |      MAC       |       QP       |      LID       |                   GID                    |
              ---------------- ---------------- ---------------- ------------------------------------------
              |  000000000000  |             0  |             0  |  0x0000-0000-0000-00000000000000000000  |
```

If no flags are provided, then local, multicast and remote EP tables should be displayed.

If interface name is not provided, the information is shown for of all the interfaces.

4.6.2.7 Get-state

This tool is used to show the device state.

The parameters that should be used in the following command are detailed in the PCI information found under "Location”, as demonstrated in the image below:

```
<bus#> <device#> <function#>
```

For further details on these parameters, please refer to Section 4.6.2.1, “mstdump”, on page 190.

In order to view the list of all devices, please refer to Section 4.6.3.2, “show device list”, on page 196.

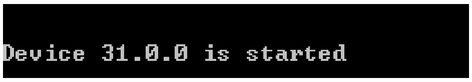
Usage:

```
mlxtool.exe dbg get-state [<pci-bus#> <pci-device#> <pci-function#>]
```

Example:

If "Location" is "PCI Slot 3 (PCI bus 31, device 0, function 0)", type:

```
mlxtool dbg get-state 31 0 0
```



```
Device 31.0.0 is started
```

If PCI is not provided, the information is shown for of all the devices.

4.6.2.8 Restart

Restart interface network adapter.

Usage:

```
 mlxtool dbg restart <Interface Name>
```

4.6.3 show

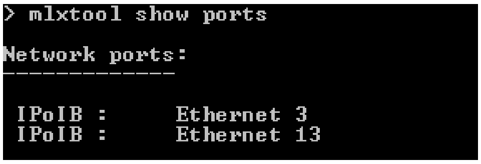
This tool is used to show specific information.

Usage:

```
mlxtool.exe show <tool-name> <tool-arguments>
```

4.6.3.1 show port list

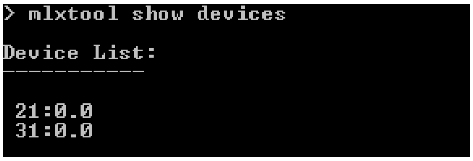
This tool is used to show the Ethernet and IPoIB port list:



```
> mlxtool show ports

Network ports:
--------------

 IPoIB :     Ethernet 3
 IPoIB :     Ethernet 13
```

4.6.3.2 show device list

This tool is used to show the PCI list for devices:



```
> mlxtool show devices

Device List:
-----------

 21:0.0
 31:0.0
```

4.6.4 modify Tool

This tool is used to modify driver parameters.

Usage:

```
mlxtool.exe modify <tool-name> <tool-arguments>
```

4.6.4.1 Modify Traffic Classes Bandwidth (BW) Limit Configuration Tool

This tool is used to modify the BW limitations' configuration for the different traffic classes.

Usage:

```
mlxtool.exe modify tc-bw <Interface Name> <traffic class> <max BW in Mbps units> [<traffic class 2> <max BW in Mbps units 2>... <traffic class n> <max BW in Mbps units n>]
```

This tool requires an interface name and at least one traffic class to change. Any number of traffic class and BW limitation pairs may follow.

For example, to change for adapter "Ethernet 6" traffic class number 3 limitation to 100 Mbps, and traffic class number 5 limitation to 300 Mbps, run the following command:

```
mlxtool.exe modify tc-bw "Ethernet 6" 3 100 5 300
```

The tool supports rates in units of 100Mbps or 1GBps. The requested BW limitations values will be rounded down to the closest supported value. A rate below the minimal value supported by the device will be rounded up to the minimal value supported (100Mbps).

5 Troubleshooting

You may be able to easily resolve the issues described in this section. If a problem persists and you are unable to resolve it, please contact your Mellanox representative or Mellanox Support at support@mellanox.com.

5.1 Installation Related Troubleshooting

5.1.1 Installation Error Codes and Troubleshooting

Table 41 - Installation Related Issues

Issue	Cause	Solution
Machine may become unresponsive during driver upgrade from WinOF v4.70 or earlier.	Upgrade requires unloading the old driver first, and this is when the machine may become unresponsive.	There are two solutions for this issue: • If possible, load an OS image with the new driver installed. • Reboot the machine prior to the upgrade operation to reduce the probability of hitting the machine freeze issue.
The installation of WinOF fails with the following error message: `"This installation package is not supported by this processor type. Contact your product vendor".`	An incorrect driver version might have been installed, e.g., you are trying to install a 64-bit driver on a 32-bit machine (or vice versa).	Use the correct driver package according to the CPU architecture.
The installation of WinOF fails and reads as follows: `"The installation cannot be done while the RDSH service is enabled, please disable it. You may re-enable it after the installation is complete".`	A known issue in windows installer when using the chain MSI feature, as described in the following link: http://rcmtech.wordpress.com/2013/08/27/server-2012-remote-desktop-session-host-installation-hangs-at-windows-installer-coordinator/	Follow the recommendation in the article.

5.1.1.1 Setup Return Codes

Table 42 - Setup Return Codes

Error Code	Description	Troubleshooting
1603	Fatal error during installation	Contact support
1633	The installation package is not supported on this platform.	Make sure you are installing the right package for your platform

For additional details on Windows installer return codes, please refer to:
http://support.microsoft.com/kb/229683

5.1.1.2 Firmware Burning Warning Codes

Table 43 - Firmware Burning Warning Codes

Error Code	Description	Troubleshooting
1004	Failed to open the device	Contact support
1005	Could not find an image for at least one device	The firmware for your device was not found. Please try to manually burn the firmware.
1006	Found one device that has multiple images	Burn the firmware manually and select the image you want to burn.
1007	Found one device for which force update is required	Burn the firmware manually with the force flag.
1008	Found one device that has mixed versions	The firmware version or the expansion rom version does not match.

For additional details, please refer to the MFT User Manual:

http://www.mellanox.com > Products > Firmware Tools

5.1.1.3 Restore Configuration Warnings

Table 44 - Restore Configuration Warnings

Error Code	Description	Troubleshooting
3	Failed to restore the configuration	Please see log for more details and contact the support team

5.2 InfiniBand Related Troubleshooting

Table 45 - InfiniBand Related Issues

Issue	Cause	Solution
The InfiniBand interfaces are not up after the first reboot after the installation process is completed.	Port status might be `PORT_DOWN`: Switch port state might be “disabled” or cable is disconnected.	Enable switch admin or connect cable.
	Port status might be `PORT_INITIALIZED`: SM might not be running on the fabric.	Run the SM on the fabric.
	Port status might be `PORT_ARMED`: Firmware issue.	Please contact Mellanox Support.
Ethernet interface is started instead of InfiniBand.	BMC is enabled.	Disable BMC
	The firmware version is not up-to-date.	Burn the updated version.[a]

[a] This issue can occur when using firmware version 2.40.5000. To avoid it, upgrade to version 2.40.5030 and above.

5.3 Ethernet Related Troubleshooting

For further performance related information, please refer to the *Performance Tuning Guide* and to Section 3.8, “Performance Tuning and Counters”, on page 154

Table 46 - Ethernet Related Issues

Issue	Cause	Solution
Low performance.	Non-optimal system configuration might have occurred.	See section “Performance Tuning and Counters” on page 154. to take advantage of Mellanox 10/40/56 GBit NIC performance.
The driver fails to start.	There might have been an RSS configuration mismatch between the TCP stack and the Mellanox adapter.	1. Open the event log and look under `"System"` for the `"mlx4ethX"` source. 2. If found, enable RSS, run: `"netsh int tcp set global rss = enabled"`. or a less recommended suggestion (as it will cause low performance): • Disable RSS on the adapter, run: `"netsh int tcp set global rss = no dynamic balancing"`.

Table 46 - Ethernet Related Issues

Issue	Cause	Solution
The driver fails to start and a yellow sign appears near the "`Mellanox ConnectX 10Gb Ethernet Adapter`" in the Device Manager display. (Code 10)	A hardware error might have occurred.	Disable and re-enable "`Mellanox ConnectX Adapter`" from the Device Manager display. In case it does not work, refer to support.
The driver fails to start and in the Event log, under the mlx4_bus source, the following error message appears: `"RUN_FW command failed with error -22"`	A wrong firmware image might have been programmed on the adapter card.	See Section 2.7,"Firmware Upgrade," on page 34.
No connectivity to a Fault Tolerance team while using network capture tools (e.g., Wireshark).	The network capture tool might have captured the network traffic of the non-active adapter in the team. 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 team interface instead.
No Ethernet connectivity on 10Gb adapters after activating Performance Tuning (part of the installation).	A TcpWindowSize registry value might have been added.	• Remove the value key under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\TcpWindowSize` Or • Set its value to `0xFFFF`.
Packets are being lost.	The port MTU might have been set to a value higher than the maximum MTU supported by the switch.	Change the MTU according to the maximum MTU supported by the switch.
NVGRE changes done on a running VM, are not propagated to the VM.	The configuration changes might not have taken effect until the OS is restarted.	Stop the VM and afterwards perform any NVGRE configuration changes on the VM connected to the SR-IOV-enabled virtual switch.

5.4 Performance Related Troubleshooting

Table 47 - Performance Related Issues

Issue	Cause	Solution
Low performance issues	The OS profile might not be configured for maximun performace.	1. Go to "Power Options" in the "Control Panel". Make sure "Maximum Performance" is set as the power scheme 2. Reboot the machine.
Flow Control is disabled when kernel debugger is configured in Windows server 2012 and above.	When a kernel debugger is configured (not necessarily physically connected) then the flow control might be disabled.	Set the registry key as following: `HKLM\SYSTEM\CurrentControlSet\Services\NDIS\Parameters` • Type: `REG_DWORD` • Key name: `AllowFlowControlUnderDebugger` • Value: `1`
Package drop or low performance on specific traffic class.	Might be a lack of QoS and Flow Control settings configuration or their misconfiguration.	Check the configured settings for all of the QoS options. Open a PowerShell prompt and use "Get-NetAdapterQos". To achieve maximum performance all of the following must exist: • All of the hosts, switches and routers should use the same matching flow control settings. If Global-pause is used, all devices must be configured for it. If PFC (Priority Flow-control) is used all devices must have matching settings for all priorities. • ETS settings that limit speed of some priorities will greatly affect the output results. • Make sure Flow-Control is enabled on the Mellanox Interfaces (enabled by default). Go to the device manager, right click the Mellanox interface go to "Advanced" and make sure Flow-control is enabled for both TX and RX. • To eliminate QoS and Flow-control as the performance degrading factor, set all devices to run with Global Pause and rerun the tests: • Set Global pause on the switches, routers. • Run "Disable-NetAdapterQos *" on all of the hosts in a PowerShell window.

5.4.1 General Diagnostic

Issue 1. Go to “Device Manager”, locate the Mellanox adapter that you are debugging, right-click and choose “Properties” and go to the “Information” tab:

- PCI Gen 1: should appear as "PCI-E 2.5 GT/s"
- PCI Gen 2: should appear as "PCI-E 5.0 GT/s"
- PCI Gen 3: should appear as "PCI-E 8.0 GT/s"
- Link Speed: 56.0 Gbps / 40.0Gbps / 10.0Gbps

Issue 2. To determine if the Mellanox NIC and PCI bus can achieve their maximum speed, it's best to run nd_send_bw in a loopback. On the same machine:

1. Run "start /b /affinity 0x1 nd_send_bw -S <IP_host>" where <IP_host> is the local IP.
2. Run "start /b /affinity 0x2 nd_send_bw -C <IP_host>"
3. Repeat for port 2 with the appropriate IP.
4. On PCI Gen3 the expected result is around 5700MB/s

On PCI Gen2 the expected result is around 3300MB/s

Any number lower than that points to bad configuration or installation on the wrong PCI slot. Malfunctioning QoS settings and Flow Control can be the cause as well.

Issue 3. To determine the maximum speed between the two sides with the most basic test:

1. Run "nd_send_bw -S <IP_host1>" on machine 1 where <IP_host1> is the local IP.
2. Run "nd_send_bw -C <IP_host1>" on machine 2.
3. Results appear in Gb/s (Gigabits 2^30), and reflect the actual data that was transferred, excluding headers.
4. If these results are not as expected, the problem is most probably with one or more of the following:

- Old Firmware version.Misconfigured Flow-control: Global pause or PFC is configured wrong on the hosts, routers andswitches. See Section 3.1.4,“RDMA over Converged Ethernet (RoCE),” on page 47
- CPU/power options are not set to "Maximum Performance".

5.5 Virtualization Related Troubleshooting

Table 48 - Virtualization Related Issues

Issue	Cause	Solution
Mellanox driver fails to load a host machine in SR-IOV environment and appears with yellow bang in Device Manager.	The device may not have been able to find enough free resources that it can use. (Code 12).	1. Boot to BIOS and disable SR-IOV. 2. Burn Firmware with lower number of VFs. 3. Re-enable SR-IOV in BIOS. For more information, please contact Mellanox support.
Running Windows server 2012, 2012 R2 and 2016 as VM over ESX with Mellanox adpter cards connected as Direct pass-through fails to power on.	ConnectX adapter network cards might be trying to use too many MSI-X vectors.	1. Go to the vSphere Web Client. 2. Right-click the virtual machine and select Edit Settings. 3. Click the Options tab and expand Advanced. 4. Click Edit Configuration. 5. Click Add Row. 6. Add the parameter to the new row: • In the Name column, add pciPassthru0.maxMSIXvectors. • In the Value column, add 31. 7. Click OK and click OK again. For further details, please refer to: http://kb.vmware.com/selfservice/microsites/search.do?cmd=displayKC&docType=kc&externalId=2032981&sliceId=1&docTypeID=DT_KB_1_1&dialogID=408420191&stateId=1_0_388456420
When enabling the VMQ, in case NVGRE offload is enabled, and a teaming of two virtual ports is performed, no ping is detected between the VMs and/or ping is detected but no establishing of TCP connection is possible.	Might be missing critical Microsoft updates.	Please refer to: http://support.microsoft.com/kb/2975719 “August 2014 update rollup for Windows server RT 8.1, Windows server 8.1, and Windows server 2012 R2” – specifically, fixes.

Table 48 - Virtualization Related Issues

Issue	Cause	Solution
In Hyper-V environment, `Enable-NetAdapterVmq` powershell command can enable VMQ on a network adapter only if the virtual switch which does not have SR-IOV enabled is defined over corresponding network adapter.	The powershell command might depend on two registry fields: `*VMQ` and `*RssOrVmqPreference`, when the former is controlled by powershell and the latter is controlled by the virtual switch.	For further information on these registry keys, please refer to: http://msdn.microsoft.com/en-us/library/windows/hardware/hh451362(v=vs.85).aspx
Mellanox driver fails to load in a guest machine in SR-IOV environment and appears with yellow bang in the Device Manager.	The host machine cannot reserve enough QPs for the specific VF.	Increase the LogNumQP in the mlx4_bus registry.
Mellanox driver fails to load in a guest machine with Windows 10 Inbox driver version 4.91 in SR-IOV environment and appears with yellow bang in the Device Manager.	The host machine cannot reserve enough QPs for the specific VF.	Update to the latest version of Windows 10 or install driver version 5.22 and above.

5.6 Reported Driver Events

The driver records events in the system log of the Windows server event system which can be used to identify, diagnose, and predict sources of system problems.

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

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

  OR

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>

5.7 Extracting WPP Traces

WinOF Mellanox driver automatically dumps trace messages that can be used by the driver developers for debugging issues that have recently occurred on the machine.

The default location for the trace files is:

```
%SystemRoot%\system32\LogFiles\Mlnx\
```

There are one or more trace files, those whose name begins with:

```
Mellanox-System.etl
```

The automatic trace session is called Mellanox-Kernel.

In order to view the session, run the following command:

```
logman query Melloanox-Kernel -ets
```

In order to stop the session, run the following command:

```
logman stop Mellanox-Kernel -ets
```

When opening a support ticket, it is advised to attach the file to the ticket.

5.8 State Dumping

Upon several types of events, the drivers can produce a set of files reflecting the current state of the adapter.

Automatic state dumps are done upon the following events:

Table 49 - Events Causing Automatic State Dumps

Event Type	Description	Provider	Default	Tag
FATAL_ERR	The driver detects an error that does not allow the device to function normally and requires a reset	Mlx4_bus	On	f
CMD_TIMEOUT	Timeout on a command, sent to HCA	Mlx4_bus	On	c
EQ_STUCK	The driver decides that an Event Queue is stuck	Mlx4eth63, IPoIB6x	On	e
TXCQ_STUCK	The driver decides that the transmit completion queue is stuck	Mlx4eth63, IPoIB6x	On	t
PORT_STATE	The port state changes	Mlx4eth63, IPoIB6x	On	p
ON_IOCTL	User application asks to generate dump files	Mlx4_bus	N/A	u

where:

Provider	The driver creating the set of files.
Default	Whether or not the state dumps are created by default upon this event.
Tag	Part of the file name, used to identify the event that has triggered the state dump.

PORT_STATE events can be disabled by adding DumpModeFlags DWORD32 parameter into HKLM\System\CurrentControlSet\Services\xxx\Parameters (where xxx is either mlx4eth63 or ipoib6). It is a bit-field with the following bit meanings::

```
DUMP_MODE_FLAGS_DISABLE_DUMP_ON_EQ_STUCK        (1 << 0)// i.e. 0x01
DUMP_MODE_FLAGS_DISABLE_DUMP_ON_TXCQ_STUCK    (1 << 1)// i.e. 0x02
DUMP_MODE_FLAGS_DISABLE_DUMP_ON_PORT_DN          (1 << 2)// i.e. 0x04
```

Events EQ_STUCK and TXCQ_STUCK can be disabled by setting the following bits:

```
DUMP_MODE_FLAGS_DISABLE_DUMP_ON_PORT_NONE     (1 << 3)// i.e. 0x08
DUMP_MODE_FLAGS_DISABLE_DUMP_ON_PORT_UP         (1 << 4)// i.e. 0x10
```

The set consists of the following files:

- 3 consecutive mstdump files
- 2 EQ dump files
- 1 FW trace file

These files are created in the %SystemRoot%\temp directory, and should be sent to Mellanox Support for analysis when debugging WinOF driver problems. Their names have the following format:<Driver_mode_of_work>_<card_location>_<event_tag_name>_<event_number>_<event_name>_<file_type>_<file_index>.log

where:

Driver_mode_of_work	The mode of driver work. For example: ‘SingleFunc’
card_location	In form bus_device_function, For example: 4_0_0
event_tag_name	One-symbol tag. See in Table 49 - “Events Causing Automatic State Dumps,” on page 207
event_number	The index of dump files set and created for this event. This number is restricted by the hidden Registry parameter DumpEventsNum
event_name	A short string naming the event. For example: ‘eth-down-1’ = “Ethernet port1 passed to DOWN state”
file_type	Type of file in the set. For example: “crspace”, “fwtrace”, “eq_dump” and “eq_print”
file_index	The file number of this type in the set

Example:

Name: SingleFunc_4_0_0_p000_eth-down-1_eq_dump_0.log

The default number of sets of files for each event is 20. It can be changed by adding DumpEventsNum DWORD32 parameter under HKLM\System\CurrnetControlSet\Services\mlx4_bus\Parameters and setting it to another value.

Appendix A: NVGRE Configuration Scripts Examples

The setup is as follow for both examples below:

```
 Hypervisor mtlae14 = "Port1", 192.168.20.114/24
 VM on mtlae14 = mtlae14-005, 172.16.14.5/16, Mac 00155D720100
 VM on mtlae14 = mtlae14-006, 172.16.14.6/16, Mac 00155D720101
 Hypervisor mtlae15 = "Port1", 192.168.20.115/24
 VM on mtlae15 = mtlae15-005, 172.16.15.5/16, Mac 00155D730100
 VM on mtlae15 = mtlae15-006, 172.16.15.6/16, Mac 00155D730101
```

A.1 Adding NVGRE Configuration to Host 14 Example

The following is an example of adding NVGRE to Host 14.

```
 # On both sides
 # vSwitch create command

 # Note, that vSwitch configuration is persistent, no need to configure it after each reboot

 New-VMSwitch "VSwMLNX" -NetAdapterName "Port1" -AllowManagementOS $true

 # Shut down VMs
 Stop-VM -Name "mtlae14-005" -Force -Confirm
 Stop-VM -Name "mtlae14-006" -Force -Confirm
 # Connect VM to vSwitch (maybe you have to switch off VM before), doing manual does also work
 # Connect-VMNetworkAdapter -VMName " mtlae14-005" -SwitchName "VSwMLNX"
 Add-VMNetworkAdapter -VMName "mtlae14-005" -SwitchName "VSwMLNX" -StaticMacAddress
"00155D720100"
 Add-VMNetworkAdapter -VMName "mtlae14-006" -SwitchName "VSwMLNX" -StaticMacAddress
"00155D720101"
 # -------  The commands from Step 2 - 4 are not persistent, Its suggested to create script is run-
ning after each OS reboot

 # Step 2. Configure a Subnet Locator and Route records on each Hyper-V Host (Host 1 and Host 2)
mtlae14 & mtlae15
 New-NetVirtualizationLookupRecord -CustomerAddress 172.16.14.5 -ProviderAddress 192.168.20.114 -
VirtualSubnetID 5001 -MACAddress "00155D720100" -Rule "TranslationMethodEncap"
 New-NetVirtualizationLookupRecord -CustomerAddress 172.16.14.6 -ProviderAddress 192.168.20.114 -
VirtualSubnetID 5001 -MACAddress "00155D720101" -Rule "TranslationMethodEncap"
 New-NetVirtualizationLookupRecord -CustomerAddress 172.16.15.5 -ProviderAddress 192.168.20.115 -
VirtualSubnetID 5001 -MACAddress "00155D730100" -Rule "TranslationMethodEncap"
 New-NetVirtualizationLookupRecord -CustomerAddress 172.16.15.6 -ProviderAddress 192.168.20.115 -
VirtualSubnetID 5001 -MACAddress "00155D730101" -Rule "TranslationMethodEncap"
 # Add customer route
 New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}" -
VirtualSubnetID "5001" -DestinationPrefix "172.16.0.0/16" -NextHop "0.0.0.0" -Metric 255
```

```
 # Step 3. Configure the Provider Address and Route records on Hyper-V Host 1 (Host 1 Only)
mtlae14
 $NIC = Get-NetAdapter "Port1"
 New-NetVirtualizationProviderAddress -InterfaceIndex $NIC.InterfaceIndex -ProviderAddress
192.168.20.114 -PrefixLength 24
 New-NetVirtualizationProviderRoute -InterfaceIndex $NIC.InterfaceIndex -DestinationPrefix
"0.0.0.0/0" -NextHop 192.168.20.1
 # Step 5. Configure the Virtual Subnet ID on the Hyper-V Network Switch Ports for each Virtual
Machine on each Hyper-V Host (Host 1 and Host 2)
 # Run the command below for each VM on the host the VM is running on it, i.e. the for  mtlae14-
005, mtlae14-006 on
 # host 192.168.20.114 and for VMs mtlae15-005, mtlae15-006 on host 192.168.20.115
 # mtlae14 only
 Get-VMNetworkAdapter -VMName mtlae14-005 | where {$_.MacAddress -eq "00155D720100"} | Set-VMNet-
workAdapter -VirtualSubnetID 5001
 Get-VMNetworkAdapter -VMName mtlae14-006 | where {$_.MacAddress -eq "00155D720101"} | Set-VMNet-
workAdapter -VirtualSubnetID 5001
```

A.2 Adding NVGRE Configuration to Host 15 Example

The following is an example of adding NVGRE to Host 15.

```
 # On both sides
 # vSwitch create command

 # Note, that vSwitch configuration is persistent, no need to configure it after each reboot

 New-VMSwitch "VSwMLNX" -NetAdapterName "Port1" -AllowManagementOS $true

 # Shut down VMs
 Stop-VM -Name "mtlae15-005" -Force -Confirm
 Stop-VM -Name "mtlae15-006" -Force -Confirm
 # Connect VM to vSwitch (maybe you have to switch off VM before), doing manual does also work
 # Connect-VMNetworkAdapter -VMName " mtlae14-005" -SwitchName "VSwMLNX"
 Add-VMNetworkAdapter -VMName "mtlae15-005" -SwitchName "VSwMLNX" -StaticMacAddress
"00155D730100"
 Add-VMNetworkAdapter -VMName "mtlae15-006" -SwitchName "VSwMLNX" -StaticMacAddress
"00155D730101"
```

```
 # -------  The commands from Step 2 - 4 are not persistent, Its suggested to create script is run-
ning after each OS reboot

 # Step 2. Configure a Subnet Locator and Route records on each Hyper-V Host (Host 1 and Host 2)
mtlae14 & mtlae15
 New-NetVirtualizationLookupRecord -CustomerAddress 172.16.14.5 -ProviderAddress 192.168.20.114 -
VirtualSubnetID 5001 -MACAddress "00155D720100" -Rule "TranslationMethodEncap"
 New-NetVirtualizationLookupRecord -CustomerAddress 172.16.14.6 -ProviderAddress 192.168.20.114 -
VirtualSubnetID 5001 -MACAddress "00155D720101" -Rule "TranslationMethodEncap"
 New-NetVirtualizationLookupRecord -CustomerAddress 172.16.15.5 -ProviderAddress 192.168.20.115 -
VirtualSubnetID 5001 -MACAddress "00155D730100" -Rule "TranslationMethodEncap"
 New-NetVirtualizationLookupRecord -CustomerAddress 172.16.15.6 -ProviderAddress 192.168.20.115 -
VirtualSubnetID 5001 -MACAddress "00155D730101" -Rule "TranslationMethodEncap"
 # Add customer route
 New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}" -
VirtualSubnetID "5001" -DestinationPrefix "172.16.0.0/16" -NextHop "0.0.0.0" -Metric 255
 # Step 4. Configure the Provider Address and Route records on Hyper-V Host 2 (Host 2 Only)
mtlae15
 $NIC = Get-NetAdapter "Port1"
 New-NetVirtualizationProviderAddress -InterfaceIndex $NIC.InterfaceIndex -ProviderAddress
192.168.20.115 -PrefixLength 24
 New-NetVirtualizationProviderRoute -InterfaceIndex $NIC.InterfaceIndex -DestinationPrefix
"0.0.0.0/0" -NextHop 192.168.20.1
 # Step 5. Configure the Virtual Subnet ID on the Hyper-V Network Switch Ports for each Virtual
Machine on each Hyper-V Host (Host 1 and Host 2)
 # Run the command below for each VM on the host the VM is running on it, i.e. the for  mtlae14-
005, mtlae14-006 on
 # host 192.168.20.114 and for VMs mtlae15-005, mtlae15-006 on host 192.168.20.115
 # mtlae15 only
 Get-VMNetworkAdapter -VMName mtlae15-005 | where {$_.MacAddress -eq "00155D730100"} | Set-VMNet-
workAdapter -VirtualSubnetID 5001
 Get-VMNetworkAdapter -VMName mtlae15-006 | where {$_.MacAddress -eq "00155D730101"} | Set-VMNet-
workAdapter -VirtualSubnetID 5001
```

Appendix B: Windows MPI (MS-MPI)

B.1 Overview

Message Passing Interface (MPI) is meant to provide virtual topology, synchronization, and communication functionality between a set of processes.

With MPI you can run one process on several hosts.

- Windows MPI run over the following protocols:
  - Sockets (Ethernet)
  - Network Direct (ND)

B.1.1 System Requirements

- Install HPC (Build: 4.0.3906.0).
- Validate traffic (ping) between the whole MPI Hosts.
- Every MPI client need to run smpd process which open the mpi channel.
- MPI Initiator Server need to run: mpiexec. If the initiator is also client it should also run smpd.

B.2 Running MPI

Step 1. Run the following command on each mpi client.

```
start smpd -d -p <port>
```

Step 2. Install ND provider on each MPI client in MPI ND.

Step 3. Run the following command on MPI server.

```
mpiexec.exe -p <smpd_port> -hosts <num_of_hosts>
<hosts_ip_list> -env MPICH_NETMASK <network_ip/subnet> -
env MPICH_ND_ZCOPY_THRESHOLD -1 -env MPICH_DISABLE_ND <0/
1> -env MPICH_DISABLE_SOCK <0/1> -affinity <process>
```

B.3 Directing MSMPI Traffic

Directing MPI traffic to a specific QoS priority may delayed due to:

- Except for NetDirectPortMatchCondition, the QoS powershell CmdLet for NetworkDirect traffic does not support port range. Therefore, NetwrokDirect traffic cannot be directed to ports 1-65536.
- The MSMPI directive to control the port range (namely: MPICH_PORT_RANGE 3000,3030) is not working for ND, and MSMPI chose a random port.

B.4 Running MSMPI on the Desired Priority

Step 1. Set the default QoS policy to be the desired priority (Note: this prio should be lossless all the way in the switches*)

Step 2. Set SMB policy to a desired priority only if SMD Traffic running.

Step 3. **[Recommended]** Direct ALL TCP/UDP traffic to a lossy priority by using the “IPProtocolMatchCondition”.

> TCP is being used for MPI control channel (smpd), while UDP is being used for other services such as remote-desktop.

Arista switches forwards the pcp bits (e.g. 802.1p priority within the vlan tag) from ingress to egress to enable any two End-Nodes in the fabric as to maintain the priority along the route.

In this case the packet from the sender goes out with priority X and reaches the far end-node with the same priority X.

> The priority should be losslessin the switches

➢ ***To force MSMPI to work over ND and not over sockets, add the following in mpiexec command:***

```
-env MPICH_DISABLE_ND 0 -env MPICH_DISABLE_SOCK 1
```

B.5 Configuring MPI

Step 1. Configure all the hosts in the cluster with identical PFC (see the PFC example below).

Step 2. Run the WHCK ND based traffic tests to Check PFC (ndrping, ndping, ndrpingpong, ndpingpong).

Step 3. Validate PFC counters, during the run-time of ND tests, with “Mellanox Adapter QoS Counters” in the perfmon.

Step 4. Install the same version of HPC Pack in the entire cluster.
NOTE: Version mismatch in HPC Pack 2012 can cause MPI to hung.

Step 5. Validate the MPI base infrastructure with simple commands, such as `“hostname”`.

B.5.1 PFC Example

In the example below, ND and NDK go to priority 3 that configures no-drop in the switches. The TCP/UDP traffic directs ALL traffic to priority 1.

- Install dcbx.

```
Install-WindowsFeature Data-Center-Bridging
```

- Remove the entire previous settings.

```
Remove-NetQosTrafficClass
Remove-NetQosPolicy -Confirm:$False
```

- Set the DCBX Willing parameter to false as Mellanox drivers do not support this feature

```
Set-NetQosDcbxSetting -Willing 0
```

- Create a Quality of Service (QoS) policy and tag each type of traffic with the relevant priority.
  In this example we used TCP/UDP priority 1, ND/NDK priority 3.

```
New-NetQosPolicy “SMB" -NetDirectPortMatchCondition 445 -PriorityValue8021Action 3
New-NetQosPolicy “DEFAULT" -Default -PriorityValue8021Action 3
New-NetQosPolicy “TCP" -IPProtocolMatchCondition TCP -PriorityValue8021Action1
New-NetQosPolicy “UDP" -IPProtocolMatchCondition UDP -PriorityValue8021Action 1
```

- Enable PFC on priority 3.

```
Enable-NetQosFlowControl 3
```

- Disable Priority Flow Control (PFC) for all other priorities except for 3.

```
Disable-NetQosFlowControl 0,1,2,4,5,6,7
```

- Enable QoS on the relevant interface.

```
Enable-netadapterqos -Name
```

B.5.2 Running MPI Command Examples

- Running MPI pallas test over ND.

```
> mpiexec.exe -p 19020 -hosts 4 11.11.146.101 11.21.147.101
11.21.147.51
11.11.145.101 -env MPICH_NETMASK 11.0.0.0/
255.0.0.0 -env MPICH_ND_ZCOPY_THRESHOLD -1 -env MPICH_DISABLE_ND 0
-env
MPICH_DISABLE_SOCK 1 -affinity c:\\test1.exe
```

- Running MPI pallas test over ETH.

```
> exempiexec.exe -p 19020 -hosts 4 11.11.146.101 11.21.147.101
11.21.147.51
11.11.145.101 -env MPICH_NETMASK 11.0.0.0/
255.0.0.0 -env MPICH_ND_ZCOPY_THRESHOLD -1 -env MPICH_DISABLE_ND 1
-env
MPICH_DISABLE_SOCK 0 -affinity c:\\test1.exe
```